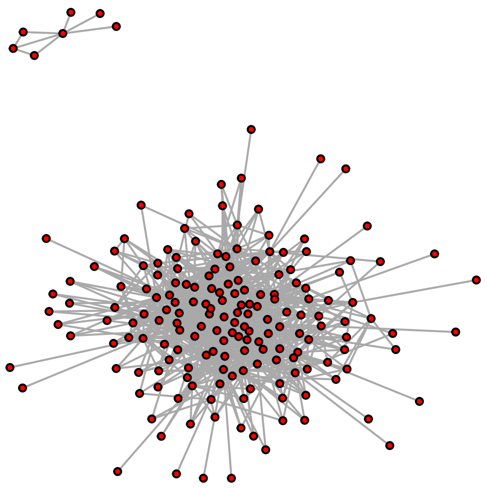
HW4_Mengdi

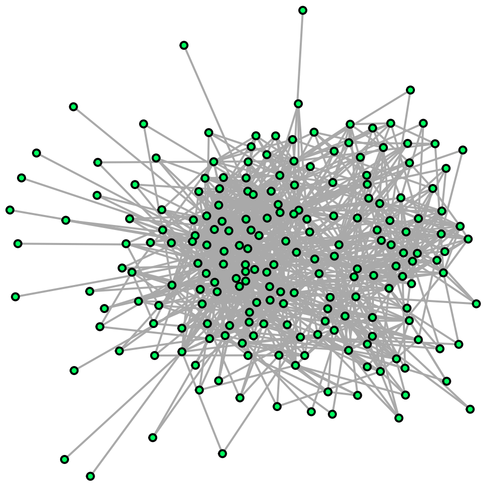
Mengdi Hao

2024-04-15

```
## Warning: package 'igraph' was built under R version 4.3.3

##
## Attaching package: 'igraph'

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

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

## Warning: `graph.edgelist()` was deprecated in igraph 2.0.0.
## i Please use `graph_from_edgelist()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: `induced.subgraph()` was deprecated in igraph 2.0.0.
## i Please use `induced_subgraph()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
## Warning: package 'gamlr' was built under R version 4.3.3
## Loading required package: Matrix
```



Q1. I'd transform degree to create our treatment variable d. What would you do and why?

I would apply a logarithmic transformation to the degree variable for the following reasons:

- Mitigating Skewness: Log transformation reduces right-skewness in count data like network degrees, making the distribution more normal and compatible with linear regression assumptions.
- Linearizing Relationships: It transforms exponential relationships into linear ones, simplifying model interpretation and fitting.
- Interpreting as Ratios: Post-transformation, coefficients represent percentage changes, which is useful in interpreting the result.

```r
# Use log1p to avoid problems when degree = 0
hh$degree_log <- log1p(degree)
```

Q2. Build a model to predict d from x, our controls. Comment on how tight the fit is, and what that implies for estimation of a treatment effect.

The In-Sample R2 is 0.0817 from the first stage of a two-stage LASSO model. This indicates that only about 8.17% of the variance in the treatment variable d is explained by the control variables x. This low R2 suggests a weak predictive power of the controls over the treatment variable. It might indicate that the model has not captured all relevant confounding variables, potentially leading to omitted variable bias in estimating the treatment effect, risking biased results in the second stage.

```r
# Two-stage LASSO

library(gamlr)

# Convert missing values of categorical variables into reference level
hh <- naref(hh)

# Contruct the sparse desgin matrix, excluding "degree_log" and the intercept
x <- model.matrix(~ . - loan - degree_log - 1, data = hh)

# Define the treatment variable
d <- hh$degree_log

# Define the dependent variable
y <- hh$loan

# First stage LASSO: treatment variable d on control variables x
treat <- gamlr(x,d,lambda.min.ratio=1e-4)

# Predict treatment variable using control variables x
dhat <- predict(treat, x, type="response")

# Calculate IS (in-sample) R^2
cat("In-Sample R2 of the first stage model:", cor(drop(dhat),d)^2, "\n")
```

```
## In-Sample R2 of the first stage model: 0.08166857
```

Q3. Use predictions from [2] in an estimator for effect of d on loan.

The coefficient for the treatment variable degree_log is 0.0181, corresponding to an odds multiplier of exp(0.0181) = 1.0183. This suggests a modest positive effect of the degree of connection on the likelihood of making loans. Specifically, a one percent increase in the degree of connection increases the odds of a household making a loan by about 1.83%.

```r
# Second stage LASSO
causal <- gamlr(cbind(d,dhat,x),y,free=2,lmr=1e-4)
```

```
## 'as(<dgeMatrix>, "dgCMatrix")' is deprecated.
## Use 'as(., "CsparseMatrix")' instead.
## See help("Deprecated") and help("Matrix-deprecated").
```

```r
# Extract the treatment effect coefficient
cat("Treatment effect coefficient of degree_log:", coef(causal)["d",], "\n")
```

```
## Treatment effect coefficient of degree_log: 0.01812068
```

```r
# Odds multiplier
cat("Odds Multiplier:", exp(coef(causal)["d",]), "\n")
```

```
## Odds Multiplier: 1.018286
```

Q4. Compare the results from [3] to those from a straight (naive) lasso for loan on d and x. Explain why they are similar or different.

In the naive LASSO model, the coefficient for degree_log is 0.1486, translating to an odds multiplier of exp(0.1486) = 1.1602. This indicates a stronger positive effect on the likelihood of making loans compared to the two-stage LASSO: a one percent increase in the degree of connection raises the odds by about 16.02%. The discrepancy between the two models likely stems from how they handle confounders. The naive LASSO simply puts the treatment variable and the other control variables in the regression. This may lead to some important confounding variables be dropped and leave the coefficient of the treatment variable containing confounding effect. However, in the two-stage LASSO, the confounding impact of control variables are specifically handled by including the "dhat" in the regression and no penalization is imposed on it.

```r
# NAIVE lasso: directly regress y on x and d
naive <- gamlr(x = cbind(x,d), y = y, family = "binomial")

# Extract the treatment effect coefficient from the naive LASSO
cat("Treatment effect coefficient of degree_log:", coef(naive)["d",], "\n")
```

```
## Treatment effect coefficient of degree_log: 0.1485664
```

```r
# Odds multiplier
cat("Odds Multiplier:", exp(coef(naive)["d",]), "\n")
```

```
## Odds Multiplier: 1.16017
```

Q5. Bootstrap your estimator from [3] and describe the uncertainty.

I performed the bootstrap procedure 100 times to estimate the variability of the treatment effect coefficient. The mean coefficient across these samples is 0.0169, with a standard error of 0.0044. The relatively low standard error in comparison to the mean suggests that our estimate is stable and reliable, indicating a robust result with limited variability across the bootstrap samples.

```r
# Bootstrap method to calculate SE
n <- nrow(x)
gamb <- c() # empty gamma

for(b in 1:100){

  ## create a matrix of resampled indices
    ib <- sample(1:n, n, replace=TRUE)

    ## create the resampled data
    xb <- x[ib,]
    db <- d[ib]
    yb <- y[ib]

    ## run the treatment regression
    # first stage
    treatb <- gamlr(xb,db,lambda.min.ratio=1e-3)
    dhatb <- predict(treatb, xb, type="response")
    # second stage
    fitb <- gamlr(cbind(db,dhatb,xb),yb,free=2)
    gamb <- c(gamb,coef(fitb)["db",])
}
```

```r
# Summary statistics of 100 estimators
cat("Summary statistics of 100 estimators:", "\n")
```

```
## Summary statistics of 100 estimators:
```

```r
summary(gamb)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00689 0.01517 0.01780 0.01792 0.01993 0.02734
```

```r
# Standard error of treatment effect using bootstrap
cat("Standard error of treatment effect:", sd(gamb), "\n")
```

```
## Standard error of treatment effect: 0.004208225
```

More: Can you think of how you'd design an experiment to estimate the treatment effect of network degree?