Lab session practice

Slawski and Ben-David

8/21/2019

Ground Truth: all variables with correct correspondence

- Import the original data and check perform a LS regression
- This regression is the oracle model

```
wages_f <- read.csv("../data/fakedata/wages_df.csv",
                    header = TRUE)
lm0 <- lm(log(WAGE) ~ SEX
          + EXPERIENCE + I(EXPERIENCE^2)
          + EDUCATION + as.factor(OCCUPATION)
          + UNION, data = wages_f)
```

R summary

```
coef(lm0)
```

```
##             (Intercept)                     SEX
##            1.0094294300           -0.2135490993
##       I(EXPERIENCE^2)                EDUCATION as.factor(
##            -0.0004441696            0.0730130844          -
## as.factor(OCCUPATION)3 as.factor(OCCUPATION)4 as.factor(
##            -0.2262765855           -0.3658979172          -
## as.factor(OCCUPATION)6                   UNION
##            -0.1964030978            0.1998655883
```

Importing wages data files

- First import the data files.

```
wages_dA <- read.csv("../data/fakedata/wages_dA.csv", heade
wages_dB <- read.csv("../data/fakedata/wages_dB.csv", heade
```

- Check both data files
- What variables are approriate for linkage?

```
## [1] "ID"      "FNAME"   "LNAME"   "SEX"      "AGE"      "A
## [8] "WAGE"
```

```
##  [1] "ID"         "OCCUPATION" "SECTOR"     "UNION"
##  [6] "EXPERIENCE" "AGE"        "SEX"        "MARR"
## [11] "SOUTH"      "FNAME"      "LNAME"      "ADDRESS"
```

Linking the data files

- Upload the R package **fastLink** for linkg files
- In this experiment we link two files based on **ZIPCODE**.
- Check *fastlink* function first.

```
help(fastLink)
```

```
## starting httpd help server ... done
```

- To specify matching variables we choose varnames = "ZIPCODE".

```
set.seed(1427)
wages_link<-fastLink(wages_dA, wages_dB, varnames = "ZIPCOI
```

```
##
## ====================
## fastLink(): Fast Probabilistic Record Linkage
## ====================
##
```

The linked data

- We can get the linked data file form *getMatch" function.

```
matched_wages <- getMatches(wages_dA,
                    wages_dB, wages_link, combine.dfs = FALSE

dA<-matched_wages$dfA.match; dB<-matched_wages$dfB.match
```

Merging these two files

- First ensure unique column names in the merged file

```
commonvars <- intersect(colnames(dA), colnames(dB))
colnames(dA)[colnames(dA) %in% commonvars] <-
  paste("A.", colnames(dA)[colnames(dA) %in%
        commonvars], sep="")
colnames(dB)[colnames(dB) %in% commonvars] <-
        paste("B.", colnames(dB)[colnames(dB)
              %in% commonvars], sep="")
```

Merge the data linked by fastLink

```
merged_wages <-cbind.data.frame(dA, dB)
```

- Compute the fraction of mismatches (about 13%)

```
mean(merged_wages[,"A.ID"] != merged_wages[,"B.ID"])
```

```
## [1] 0.1292134831460674
```

Check the linear regression with linked file

```
lm_merged <- lm(log(WAGE) ~ B.SEX
                + EXPERIENCE + I(EXPERIENCE^2)
                +  EDUCATION + as.factor(OCCUPATION)
                + UNION, data = merged_wages)
```

Ceofficients of Naive LS regression

```
coef(lm_merged)
```

```
##                (Intercept)                      B.SEX
##  1.196847505769103481 3 -0.188788227038139994 1  0.028082
##         I(EXPERIENCE^2)                  EDUCATION as.factor
## -0.000428005694242329 4  0.059158561556855011 9 -0.257591
## as.factor(OCCUPATION)3 as.factor(OCCUPATION)4 as.factor
## -0.225527739791407472 2 -0.351471860861947205 1 -0.075200
## as.factor(OCCUPATION)6                    UNION
## -0.180964283934906572 4  0.180226534672758842 4
```

- compare the result with the original regression with original data

Robust regression

- Try robust regression with Huber loss

```
library(MASS)
rlm_merged <- rlm(log(WAGE) ~ B.SEX + EXPERIENCE + I(EXPERI
                EDUCATION + as.factor(OCCUPATION) + UNION,
              data = merged_wages)
```

- Comparing estimation error of naive and robust estimation

```
sqrt(sum((coef(lm0) - coef(lm_merged))^2))
```

```
## [1] 0.2027454431130401
```

```
sqrt(sum((coef(lm0) - coef(rlm_merged))^2))
```

```
## [1] 0.1701889571280111
```

Mixture model

- Now we try the mixure modeling approach with composite likelihood:

```
source("../code/mixture_model.R")

X <- model.matrix(lm_merged)
X <- X[,!(colnames(X) %in% "(Intercept)")]
y <- model.extract(lm_merged$model, "response")
Xc <- apply(X, 2, function(z) z - mean(z))
yc <- y - mean(y)
tausq <- mean(yc^2)
f0 <- function(z) dnorm(z, mean = 0, sd = sqrt(tausq))

res <- fit_mixture(Xc, yc, f0, control = list(init = "robus
interc <- mean(y - X %*% res$betahat)
```

Comparing the results

- Compute $\|\widehat{\beta} - \widehat{\beta}_{LS}\|$, where $\widehat{\beta}_{LS}$ denotes the estimates from the original LS regression model with original data and $\widehat{\beta}$ is the estimate from, Nïve, Huber, or mixture models.

```
coef_mixture <- c(interc, res$betahat)
```

Comparing the results (continued)

```
sqrt(sum((coef(lm0) - coef_mixture)^2))
```

```
## [1] 0.0272892695833842
```

```
sqrt(sum((coef(lm0) - coef(rlm_merged))^2))
```

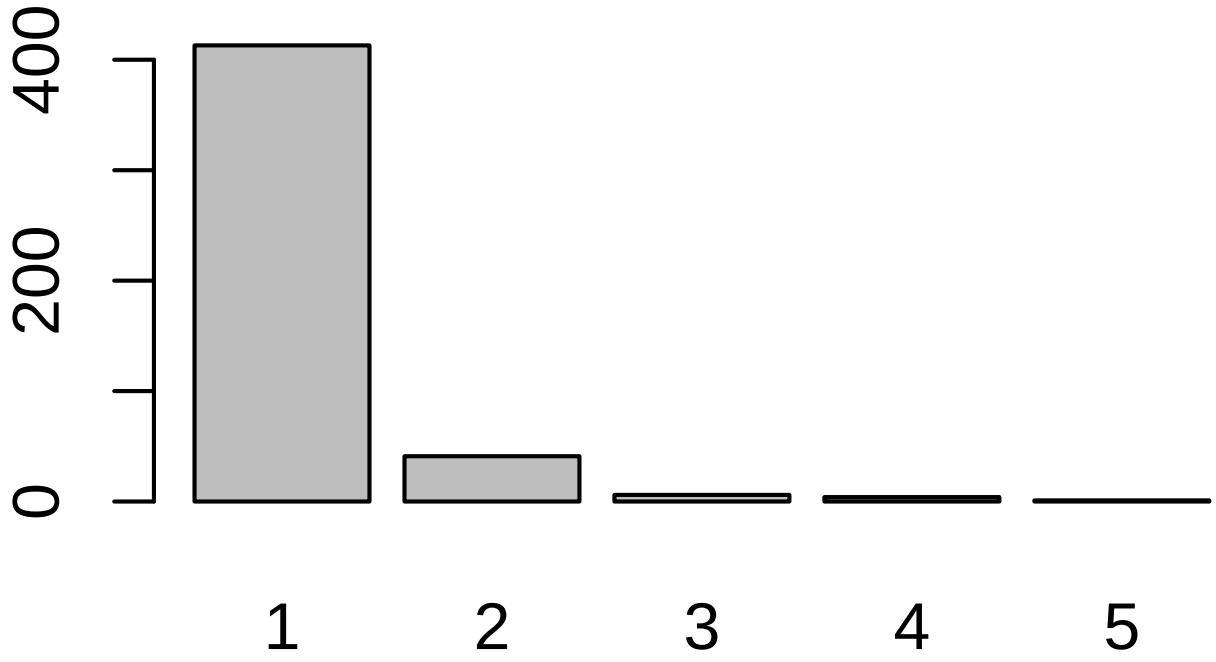
```
## [1] 0.1701889571280111
```

```
sqrt(sum((coef(lm0) - coef(lm_merged))^2))
```

```
## [1] 0.2027454431130401
```

Blocking method using information information about the linkage process

Histohram of duplicates



Compute the estimates

```
beta_Q <- coef(lahiri_larsen_block(Xc, yc, blockix))
Q <- generate_Q_block(blockix)
beta_Q_check <- lahiri_larsen(Xc, yc, Q)

interc <- mean(y - X %*% beta_Q)
coef_Q <- c(interc, beta_Q)
```

- Compute $\|\widehat{\beta} - \widehat{\beta}_{LS}\|$:

```
sqrt(sum((coef(lm0) - coef_Q)^2))
```

```
## [1] 0.07430098233549876
```

Re-matching based on sorting

```
source("../code/optimal_matching.R")

pihat <- optimal_matching(drop(Xc %*% res$betahat)
         , yc, blockix)
pihatinv <- order(pihat)
pistar <- match(merged_wages[,"B.ID"],
        merged_wages[,"A.ID"])
```

Plot of mismatches