Module X: fastLink with blocking

Rebecca C. Steorts

Reading

- Binette and Steorts (2020)
- Edmorando et al. (2020)
- Fellegi and Sunter (1969)

Agenda

- We continue with our exploration of `fastLink` by adding blocking.
- We investigate this using the `RLdata10000` data set

Load Packages

```r
# load libraries
library(fastLink)
```

Load RLdata10000

```r
# load RLdata10000
records <-
  read.table("data/RLdata10000.csv",
             sep=",", header=TRUE)
head(records, 4)
```

```
##   fname_c1 fname_c2   lname_c1 lname_c2   by bm bd rec_id ent_id
## 1    FRANK     <NA>    MUELLER     <NA> 1967  9 27      1   3606
## 2   MARTIN     <NA>    SCHWARZ     <NA> 1967  2 17      2   2560
## 3  HERBERT     <NA> ZIMMERMANN     <NA> 1961 11  6      3   3892
## 4     HANS     <NA>    SCHMITT     <NA> 1945  8 14      4    329
```

RLdata10000

```
# Number of unique records
length(unique(records$ent_id))
```

```
## [1] 9000
```

Linkage Fields

```
# linkage fields
linkageFields <- c("fname_c1", "lname_c1", "by", "bm", "bd")
```

Add Numberic Fields

```r
# We can add numeric comparisons using dissimilarity
numericMatchFields <- c("by")

#  Make sure these are class numeric
records$by <- as.numeric(records$by)
```

Preparation

```r
# linkage fields
linkageFields <- c("fname_c1",
                   "lname_c1", "by", "bm", "bd")

# string distance fields
stringDistFields <- c("fname_c1", "lname_c1")

# partial distance fields (fields where we allow
# for agree, disagree, and partially agree)
partialMatchFields <- c("fname_c1", "lname_c1")
```

Run fastLink

```
out <- fastLink(dfA = records,
                dfB = records,
                varnames = linkageFields,
                stringdist.match = stringDistFields,
                partial.match = partialMatchFields,
                cut.a.num = 1.5,
                cut.a = 0.94, cut.p = 0.84,
                dedupe = FALSE)
```

```
##
## ====================
## fastLink(): Fast Probabilistic Record Linkage
## ====================
##
## If you set return.all to FALSE, you will not be able to
## dfA and dfB are identical, assuming deduplication of a s
## Setting return.all to FALSE.
##
## Calculating matches for each variable
```

How did we do?

```r
recordsfL2 <- getMatches(dfA = records,
                         dfB = records,
                         fl.out = out)
# number of unique individuals that
# fastLink finds
length(unique(recordsfL2$dedupe.ids))
```

```
## [1] 8983
```

Blocking

1. We can use traditional, deterministic blocking, which is simple and easier but does not allow us to propagate error
2. We can also use probabilistic blocking, which allows us propagate (some) of the record linkage error

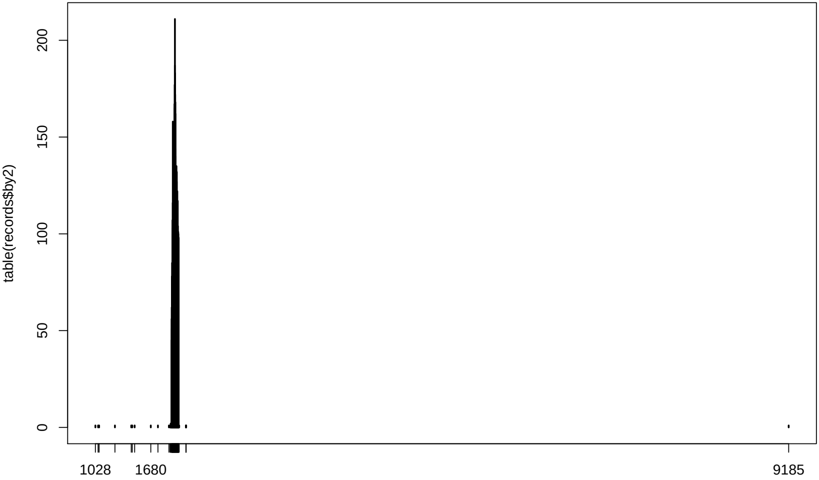
Traditional Blocking

We will create a simplistic blocking scheme based upon date of birth year.

```
records$by2 <- records$by
```

Plot

```
plot(table(records$by2))
```



Traditional Blocking

We will filter out records that are not typical for date of birth year for computational reasons.

```r
head(records$by2[records$by < 1924] <- 1923)
```

```
## [1] 1923
```

```r
head(records$by2[records$by > 2008] <- 2009)
```

```
## [1] 2009
```

Traditional Blocking

```
blockby <- blockData(records, records, varnames = "by2")
```

```
##
## ====================
## blockData(): Blocking Methods for Record Linkage
## ====================
##
## Blocking variables.
##     Blocking variable by2 using exact blocking.
##
## Combining blocked variables for final blocking assignmen
```

Traditional Blocking

```r
# modify the list of linkage fields
# birth year is of no use for
# merging within a traditional block
linkageFields2 <- c("fname_c1", "lname_c1", "bm", "bd")
```

Traditional Blocking

```r
# store the results from each block
results <- list()
for(j in 1:length(blockby)) {
  # subset original data to form block
  records.temp <- records[blockby[[j]]$dfA.inds, ]
  # fastLink applied to a block
  out.temp <- fastLink(dfA = records.temp, dfB =
              records.temp,
              varnames = linkageFields2,
              stringdist.match = stringDistFields,
              partial.match = partialMatchFields,
              cut.a = 0.92, cut.p = 0.84,
              threshold.match = 0.90,
              dedupe = FALSE)
  # get the data
  records.temp <-
    getMatches(dfA = records.temp,
               dfB = records.temp,
               fl.out = out.temp)
    # adjust the unique identifier to be block specific
  records.temp$dedupe.ids <- paste0("B", j, "_", records.temp$dedupe.id
  # Store the deduplicated data in our object for storage
  results[[j]] <- records.temp
```

How many unique entities?

```r
# aggregate all the results from the traditional blocking
recordsfL.blockE <- do.call('rbind', results)

# unique records under fastLink with blocking
length(unique(recordsfL.blockE$dedupe.ids))
```

```
## [1] 9145
```

Evaluation Metrics

```r
matches <- results
library(data.table)
trueMembership <- records$ent_id
recordIds <- records$rec_id
numRecords <- dim(records)[1]
matches <-
  data.table(cbind(out$matches$inds.a,
                   out$matches$inds.b))
head(matches)
```

```
##      V1 V2
## 1:    1  1
## 2:    2  2
## 3:    3  3
## 4:    4  4
## 5: 1957  4
## 6:    5  5
```

```r
dim(matches)[1]
```

True Positives, False Positives, and False Negatives

```
## True Positives, False Positives, and False Negatives:
TP <- sum(records$ent_id[matches$V1]
          == records$ent_id[matches$V2])
FP <- sum(records$ent_id[matches$V1]
          != records$ent_id[matches$V2])
FN <- dim(matches)[1] - TP
```

FDR and FNR

```r
## False Discovery Rate
FDR <- round(FP/(FP + TP), 2)
FDR
```

```
## [1] 0.01
```

```r
## False Negative Rate
FNR <- round(FN/dim(matches)[1], 2)
FNR
```

```
## [1] 0.01
```

Precision and Recall

```r
precision <- 1 - FDR

recall <- 1 - FNR

f1 <- (2.0*TP)/(2.0*TP+FP+FNR)

cbind(precision, recall, f1)
```

```
##      precision recall                 f1
## [1,]      0.99   0.99 0.9959240639091476
```

Summary

- We have just performed traditional blocking
- How would we proceed using probabilistic blocking and applying this to a larger data set?