Introduction to blocking techniques

Brenda Betancourt

Duke University
Department of Statistical Science
bb222@stat.duke.edu

February 8, 2018

Motivation

- Naively matching two files or finding duplicates within a file requires comparing all pairs of records.
- Infeasible for large files even when the comparisons are computationally inexpensive.
- The number of record pairs grows quadratically with the size of the dataset
  - Two files with 5,000 records $\rightarrow$ 25,000,000 comparisons!

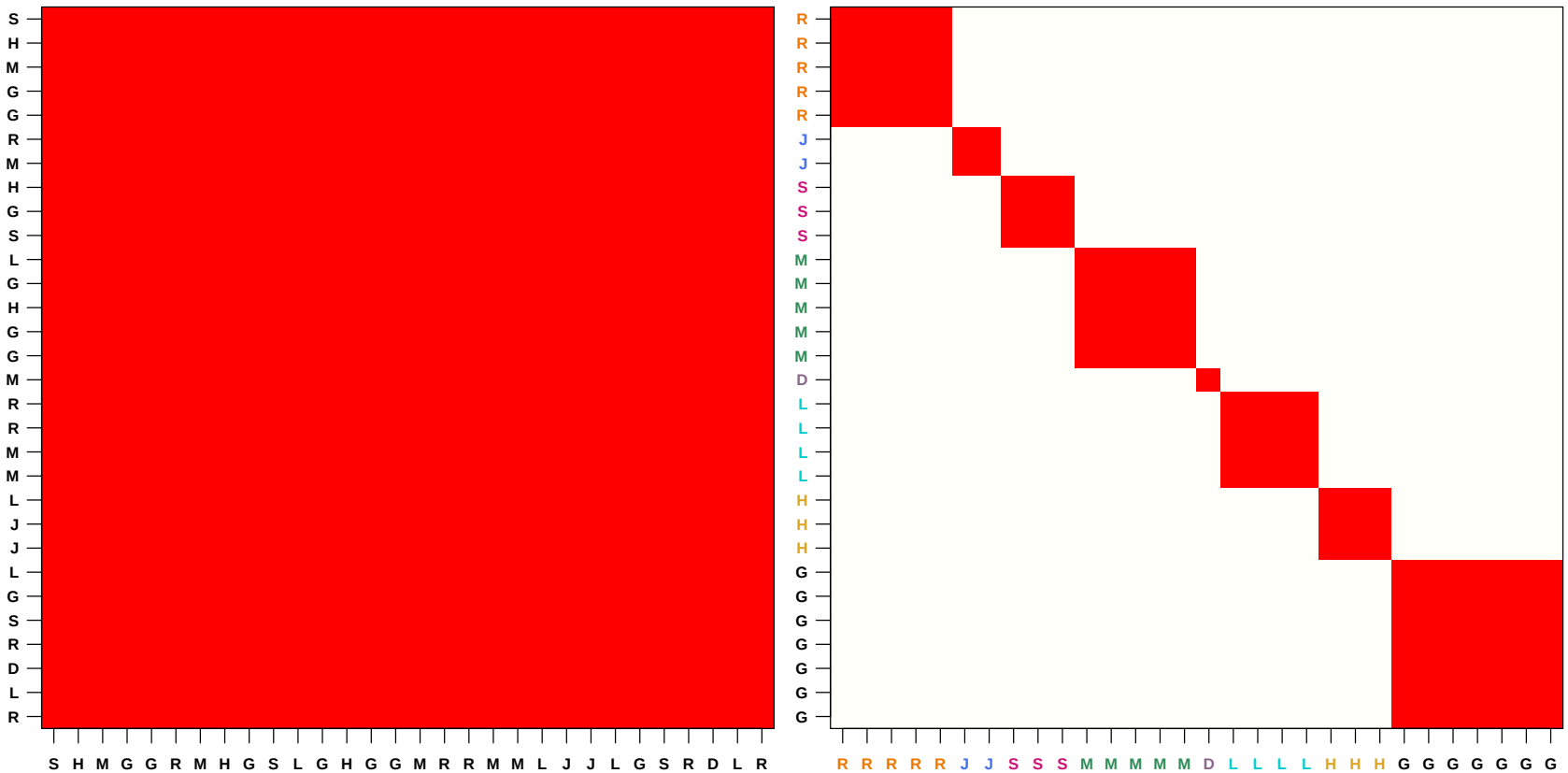
What is blocking?

Technique to reduce the comparison space:

- Filter out dissimilar record pairs that are extremely unlikely to be matches.
  - Perform record linkage only within blocks
- Traditional blocking : compare record pairs that match on one or more keys.
  - Creates a partition of the data
- Record pairs that do not meet the blocking criteria are automatically classified as non-matches.

Example: Traditional blocking

All-to-all record comparisons (left) versus partitioning records into blocks by lastname initial and comparing records only within each partition (right).



Example: RLdata500

```
library(RecordLinkage)
data(RLdata500)
head(RLdata500)
```

```
##   fname_c1 fname_c2 lname_c1 lname_c2   by bm bd
## 1  CARSTEN     <NA>    MEIER     <NA> 1949  7 22
## 2     GERD     <NA>    BAUER     <NA> 1968  7 27
## 3   ROBERT     <NA> HARTMANN     <NA> 1930  4 30
## 4   STEFAN     <NA>    WOLFF     <NA> 1957  9  2
## 5     RALF     <NA>  KRUEGER     <NA> 1966  1 13
## 6  JUERGEN     <NA>   FRANKE     <NA> 1929  7  4
```

Continuation: RLdata500

```
# Record pairs for comparison
choose(500,2)
```

```
## [1] 124750
```

```
# Blocking by last name initial
last_init <- substr(RLdata500[,"lname_c1"], 1, 1)
head(last_init)
```

```
## [1] "M" "B" "H" "W" "K" "F"
```

```
# Number of blocks
length(unique(last_init))
```

```
## [1] 20
```

Continuation: RLdata500

```
# Number of records per block
tbl <- table(last_init)
head(tbl)
```

```
## last_init
##  A  B  D  E  F  G
##  5 56  2  6 38 12
```

```
# Block sizes can vary a lot
summary(as.numeric(tbl))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    5.75    8.00   25.00   40.00  115.00
```

Continuation: RLdata500

```
# Number of records pairs per block
sapply(tbl, choose, k=2)
```

```
##     A     B     D     E     F     G     H     J     K     L     M
##    10 1540     1    15   703    66   496    28 1035    78 2850
##     S     T     V     W     Z
## 6555     1    21 1326    10
```

```
# Reduction on comparison space
sum(sapply(tbl, choose, k=2))
```

```
## [1] 14805
```

How to choose the blocking key or keys

- Fields containing the fewest errors or missing values should be chosen as blocking variables e.g. clinical diagnosis in EHR.
- Understand the kinds of errors that are unlikely for a certain field or a combination of them.
- More complex blocking schemes can be constructed using conjunctions.
  - Retain only pairs which agree on either last name initial and zip code

Example: Voter Survey data

The Views of the Electorate Research (VOTER) Survey was conducted by the survey firm YouGov.

- 8,000 adults (age 18+) with internet access took the survey on-line between November 29 and December 29, 2016.
- These respondents were originally interviewed by YouGov in 2011-2012.
- Barack Obama (Democrat) won in 2012 and Donald Trump (Republican) won in 2016.

Continuation: Voter Survey data

- Demographic variables
  - Year of birth (age)
  - Gender
  - Race
  - State
  - Education level
  - Family income

- Party affiliation: democrat, republican, independent, other

Which fields are reliable for blocking in this example?

Continuation: Is race reliable?

	2012	2016
White	6244	6198
Black	654	645
Hispanic	400	397
Mixed	160	186
Other	137	167
Asian	117	118
Native American	60	59
Middle Eastern	10	12

	White	Black	Mixed	Other
White	6073	5	46	74
Black	4	627	10	10
Mixed	31	6	100	8
Other	50	4	14	62

Continuation: Is party affiliation reliable?

	Democrat	Indepen.	Republican	Not sure	Other
Democrat	2424	192	90	25	23
Indepen.	263	1929	221	16	57
Republican	39	215	1881	11	60
Not sure	48	48	54	41	5
Other	17	46	34	2	41

Blocking caveats

- Fields can be unreliable for many applications and blocking may miss large proportions of matches i.e. increased false negatives rates.
- The frequency distribution of the values in the fields used as blocking keys will affect the size of the blocks.
- Trade-off between block sizes: true matches being missed vs computational efficiency.

Blocking by disjunctions

- Produces overlapping blocks of the data.

- Using multiple keys to consider typographical or measurement errors that would exclude true matches.
  - Blocking by last name initial or zip code

A	Mary Clain	123 Oak St	90210
B	Mary Klein	123 Oak Street	90210
C	Mary Klain	123 Oak St	50210

- Reduction in false negative rates.

Soundex algorithm

- Generates a code that represents the phonetic pronunciation of a word, helps identifying spelling variations of names.

- The Soundex code for a name consists of a letter followed by three numerical digits:
  - the letter is the first letter of the name,
  - the digits encode the remaining consonants.

- Consonants at a similar place of articulation share the same digit
  - e.g. the labial consonants B, F, P and V are each encoded as the number 1.

Example: Soundex algorithm

```
##       fname_c1 lname_c1   by bm bd
## 314     RENATE   SCHUTE 1940 12 29
## 407     RENATE  SCHULTE 1940 12 29
## 289 CHRISTINE   PETERS 1993  2  5
## 399 CHRISTINE   PETERS 1993  2  6
## 402   CHRISTA  SCHWARZ 1965  7 13
## 462  CHRISTAH  SCHWARZ 1965  7 13
```

```
library(SoundexBR)
tail(soundexBR(dup_set$fname_c1))
```

```
## [1] "R530" "R530" "C623" "C623" "C623" "C623"
```

```
tail(soundexBR(dup_set$lname_c1))
```

```
## [1] "S300" "S430" "P362" "P362" "S620" "S620"
```

Example: Soundex algorithm

```
##      fname_c1 lname_c1   by bm bd
## 130  MICHAEL    MEYER 1988  1 31
## 147  MICHAEL     MYER 1988  1 31
## 217     HORST    MEIER 1977  6  6
## 248     HORST    MEIER 1972  6  6
## 34      HEINZ    BOEHM 1938 12 20
## 111     HEINZ   BOEHMR 1938 12 20
```

```
library(SoundexBR)
head(soundexBR(dup_set$lname_c1))
```

```
## [1] "M600" "M600" "M600" "M600" "B500" "B560"
```

Cluster-based Blocking

- Records agree on the blocking variables but are still very different
  - e.g. two different people with the same lastname initial and gender
- The records in a cluster should be similar and therefore good candidate pairs for linkage.
- Methods to find clusters based on strings and cheap distance measures
  - Threshold Nearest Neighbor
  - K-Nearest Neighbor
  - Canopies (fuzzy blocking)

TNN and KNN

- Start with a single record as the base of the first cluster.
- Recursively add the nearest neighbors of records to the cluster until the distance to the nearest neighbor exceeds some threshold t.
- Choose one of the remaining records to start the next cluster.
- For KNN, ensure that each cluster has at least k records.

Canopies

- Canopies is not strictly a blocking method $\rightarrow$ fuzzy blocking
- Records can be assigned to multiple blocks $\rightarrow$ overlapping clusters not a partition of the data.
  - Randomly pick a record to be the base of the first canopy
  - Records within distance t_1 are grouped into that canopy
  - Remove records within distance $t_2 \leq t_1$ of the base
  - Pick another new record to start a second canopy

Drawbacks

- Rough distance measures for complicated high-dimensional records are non-trivial.
- Requires actually performing all or nearly all comparisons to compute
  pairwise distances.
- Note that performing traditional blocking as a first step can reduce the space considerably.

[[I think I'd like to talk about hierarchical clustering and kmeans and show an illustration with the package, don't know if instead of TNN and KNN or added]]