

Introduction to Blocking and Classical Record Linkage

Brenda Betancourt and Rebecca C. Steorts

Department of Statistical Science, affiliated faculty in Computer Science,
Biostatistics and Bioinformatics, the information initiative at Duke (i2D) and
the Social Science Research Institute (SSRI)

Duke University and U.S. Census Bureau

`beka@stat.duke.edu`

Population Dynamics and Health Program Workshop, University of Michigan

July 10, 2019

Blocking and Classical Record Linkage

① Blocking

- Focus will be on deterministic blocking

② Classical Record Linkage Methods

- Exact Matching
- String Matching
- Fellegi and Sunter (1969); Newcombe (1959).

③ Demos

- There are demos illustrating each proposed method

Computational challenge of entity resolution

- Assume M total records in two databases.
- Naive record linkage requires M^2 record comparisons.

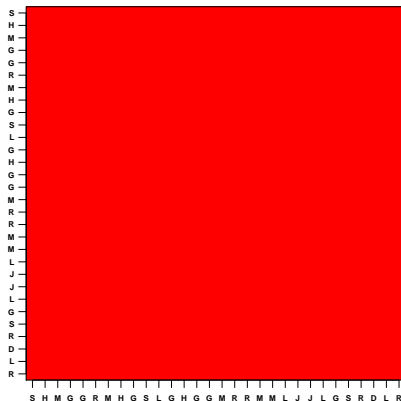


Figure: Must perform all-to-all record comparisons.

Blocking

- Blocking partitions similar records into bins or blocks.
- Record linkage is only performed within the blocks.

Blocking

① Traditional blocking

- A deterministic partition is formed based upon the data.
- A partition is created by treating certain fields that are thought to be nearly error-free as fixed.

Example: Partition of date of birth year.

② Probabilistic blocking

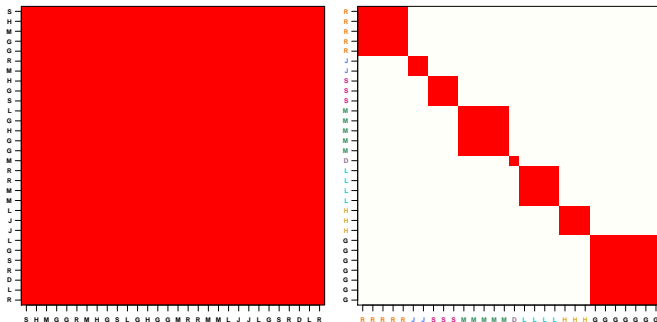
- A probability model is used to cluster the data into blocks/partitions.

Example: Locality Sensitive Hashing.

Under both blocking approaches, 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
```

Continuation: RLdata500

What is the reduction from the overall space to the reduced space?

Hint: The original space of comparisons was

```
choose(500,2)
```

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

and we have reduced the number of comparisons to

```
sum(sapply(tbl, choose, k=2))
```

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

Blocking caveats

- Features often contain errors, noise, etc. and may not be suitable for deterministic blocking.
- A noisy feature used for deterministic blocking can miss a large proportion of matches (i.e. increased false negative rates).
- The frequency distribution of the values of the blocking features will affect the block sizes.
- There is a trade off between the size of the blocks and computational efficiency.
 - If the blocks are too big, then the computational speed increases.
 - If the blocks are too small, then true matches may be missed.

How to choose the blocking features (variables or keys)

- Features containing the fewest errors or missing values should be chosen as blocking variables.
- Understand the kind of errors that are unlikely for a certain feature.
- More complex blocking schemes can be constructed using conjunctions.
 - Retain only pairs which agree on last name initial and zip code.

Classical Record Linkage: Exact matching

- Exact matching is a method that says two records are a match if they agree on every feature.
- Performing exact matching is very common in the social and health sciences in practice, however, this is not common in statistics, computer science, or machine learning.
- Other types of matching or merging are used, where records are called to be a match if they agree based upon a similarity comparison or a probabilistic model.
- Examples include: string matching, Fellegi-Sunter method, semi-supervised methods, and hashing techniques.

Classical Record Linkage: Similarity metrics

- **Levenshtein (edit) (1966)**: minimum number of substitutions required to transform one string into another e.g. **A****d****a****m** vs **A****a****n** has a distance $L = 2$, normalized as $1 - \frac{L}{\text{maxLength}} = 0.5$ for similarity.
- **Jaro-Winkler (1990)**: The Jaro distance (1989) considers common characters and character transpositions. The JW similarity measure is:

$$JW(A, B) = J(A, B) + 0.1p(1 - J(A, B))$$

where p is the # of the first four characters that agree exactly e.g. Adam vs Alan: $p=1$, $J= 0.67$ and $JW=0.7$.

These work well on English names that are less than 7 characters.

Example: RLdata500

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

##		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

Example: RLdata500

```
# Levenshtein similarity  
levenshteinSim("SCHUTE", "SCHULTE")
```

```
## [1] 0.8571429
```

```
levenshteinSim("CHRISTA", "CHRISTAH")
```

```
## [1] 0.875
```

```
# Jaro-Winkler similarity  
jarowinkler(c("SCHUTE", "CHRISTA"),  
             c("SCHULTE", "CHRISTAH"))
```

```
## [1] 0.9714286 0.9750000
```

Similarity metrics (continued)

- The Soundex algorithm generates a code representing the phonetic pronunciation of a word.
- This is typically more useful on non-English names or longer 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
 - The consonants B, F, P and V are each encoded by a 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

```
tail(soundex(dup_set$fname_c1))
```

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

```
tail(soundex(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

```
head(soundex(dup_set$lname_c1))
```

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

Blocking by disjunctions

- Produces overlapping blocks of the data.
 - **Disjunction**: records match on field A or field B
- Using multiple keys to consider typographical or measurement errors that would exclude true matches.
 - Blocking by last name initial or zip code

1.	Mary Clain	123 Oak St	90210
2.	Mary Klein	123 Oak Street	90210
3.	Mary Klain	123 Oak St	50210

- Reduction in false negative rates.

Example: Blocking by disjunctions

```
# Two records must agree in either first name initial  
# or birth year to be compared.  
# Only 2709 pairs instead of 124750!
```

```
rpairs <- compare.dedup(RLdata500c,  
blockfld = list(1, 3), #list with blocking fields  
identity = identity.RLdata500)
```

```
tail(rpairs$pairs)
```

##		id1	id2	fname_c1	lname_c1	by	bm	bd	is_match
##	2704	477	497	1	0	0	0	0	0
##	2705	479	483	0	0	1	1	0	0
##	2706	480	481	1	0	0	0	0	0
##	2707	480	490	1	0	0	0	0	0
##	2708	481	490	1	1	0	1	1	1
##	2709	494	497	0	0	1	1	0	0

Example: String comparison and blocking

```
rpairsfuzzy <- compare.dedup(RLdata500c,  
                             phonetic = FALSE, blockfld = 3,  
                             strcmp = TRUE, strcmpfun = jarowinkler)  
  
tail(rpairsfuzzy$pairs)
```

	##	id1	id2	fname_c1	lname_c1	by	bm	bd	is_match
##	1540	460	485	0.0000000	0.5396825	1	0.7	0.0	NA
##	1541	464	466	0.4555556	0.5396825	1	0.7	0.0	NA
##	1542	467	472	1.0000000	0.9333333	1	1.0	1.0	NA
##	1543	468	469	0.5777778	0.4666667	1	0.7	0.7	NA
##	1544	479	483	0.4370370	0.5619048	1	1.0	0.0	NA
##	1545	494	497	0.6111111	0.5026455	1	1.0	0.0	NA

Fellegi-Sunter (1969); Newcombe (1959)

- Fellegi-Sunter (1969); Newcombe (1959) proposed the first method for record linkage.
- Records are determined to be a match/non-match using a likelihood ratio test, which can be written as a mixture model.
- String metrics were used for names/numerical data (Levenshtein or Jaro-Winkler).

Fellegi-Sunter (1969); Newcombe (1959)

- There are feature vector for record pairs, which are classified into matches (M), nonmatches (U), and possible matches.
- Let $P(\gamma|M)$ and $P(\gamma|U)$ be probabilities of observing a feature vector γ for a matched and nonmatched pair, respectively.

Fellegi-Sunter (1969); Newcombe (1959)

- Perform record-pair classification by calculating the ratio

$$w = (P(\gamma|M)/P(\gamma|U))$$

for each candidate record pair.

- Find two thresholds based on desired error levels to optimally separate the weight values for matches, possible matches, and nonmatches.¹
- The quality of classification of the Fellegi-Sunter method relies strongly on reasonable estimations of M and U probabilities.

¹This method is very sensitive to the values of the thresholds.

Example: Blocking and Fellegi-Sunter

```
# tail(rpairs$pairs)  
# Using comparison data blocking by first name initial  
# and birth year  
rpairs1 <- epiWeights(rpairs)  
  
# Weights to compute thresholds for classification  
rpairs1$Wdata[1:5]
```

```
## [1] 0.2223402 0.2223402 0.2488181 0.2488181 0.3936336
```

Example: Fellegi-Sunter

```
summary(rpairs1)
```

Weight distribution:

(0.35,0.4]	(0.4,0.45]	(0.45,0.5]	(0.55,0.6]	(0.6,0.65]
2	10	30	50	8
(0.65,0.7]	(0.7,0.75]	(0.75,0.8]	(0.8,0.85]	(0.85,0.9]
0	0	35	8	3

Example: Fellegi-Sunter

```
result <- epiClassify(rpairs1, 0.7)
summary(result)
```

```
alpha error: 0.080000 # False negative rate
beta error: 0.000000 # False positive rate
accuracy: 0.998523
```

Classification table:

		classification		
true status		N	P	L
FALSE	2659	0	0	
TRUE	4	0	46	

Summary

- **Blocking**: reduce comparison space by choosing relatively noise free fields to match records
 - Use conjunctions (and) to create a partition of the data or disjunctions (or) to create overlapping blocks.
- **Strings**: choose a string similarity metric to compare record pairs within blocks.
 - Levenshtein or Jaro-Winkler.
- **Record linkage**: use Fellegi-Sunter method to classify records as matches, possible matches or nonmatches.

References

- Sariyar M. and Borg A. (2010), The RecordLinkage Package: Detecting Errors in Data, The R Journal Vol. 2/2. [Check big data functions of the package \(\$\geq 1'000,000\$ record pairs\)](#).
- Steorts et al. (2014) A Comparison of Blocking Methods for Record Linkage. In: Domingo-Ferrer J. (eds) Privacy in Statistical Databases. PSD 2014. Lecture Notes in Computer Science, vol 8744. Springer, Cham.
- Fellegi I.P., Sunter A.B. (1969), A Theory for Record Linkage, Journal of the American Statistical Association 64(328), pp. 1183–1210.
- Dusetzina et al. Linking Data for Health Services Research: A Framework and Instructional Guide. Rockville (MD): Agency for Healthcare Research and Quality (US) (2014). An Overview of Record Linkage Methods.
<https://www.ncbi.nlm.nih.gov/books/NBK253312/>
- Entity Resolution and Information Quality, John R. Talburt, Elsevier (2011)