Example session for Weight-based deduplication

Andreas Borg, Murat Sariyar

January 9, 2022

This document shows an example session using the package RecordLinkage. A single data set is deduplicated using an EM algorithm for weight calculation. Conducting linkage of two data sets differs only in the step of generating record pairs.

1 Generating record pairs

The data to be deduplicated is expected to reside in a data frame or matrix, each row containing one record. Example data sets of 500 and 10000 records are included in the package as RLDdata500 and RLDdata10000.

```r
data(RLdata500)
RLdata500[1:5,]
## 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
```

For deduplication, `compare.dedup` is to be used. In this example, blocking is set to return only record pairs which agree in at least two components of the subdivided date of birth, resulting in 810 pairs. The argument `identity` preserves the true matching status for later evaluation.

```r
pairs=compare.dedup(RLdata500,identity=identity.RLdata500, blockfld=list(c(5,6),c(6,7),c(5,7)))
summary(pairs)
```

```r
##
## Deduplication Data Set
##
## 500 records
## 571 record pairs
##
## 49 matches
## 522 non-matches
## 0 pairs with unknown status
```
2 Weight calculation

Weights are calculated by means of an EM algorithm. This step is computationally intensive and might take a while. The histogram shows the resulting weight distribution.

pairs=emWeights(pairs)

```r
hist(pairs$Wdata, plot=FALSE)
```

## $breaks
## [1] -15 -10 -5 0 5 10 15 20 25 30 35 40 45
##
## $counts
## [1] 352 13 0 0 5 26 42 123 9 0 0 1
##
## $density
## [1] 0.1232924694 0.0045534151 0.0000000000 0.0000000000
## [5] 0.0017513135 0.0091068301 0.0147110333 0.0430823117
## [9] 0.0031523643 0.0000000000 0.0003502627
##
## $mids
## [1] -12.5 -7.5 -2.5 2.5 7.5 12.5 17.5 22.5 27.5
## [10] 32.5 37.5 42.5
##
## $xname
## [1] "pairs$Wdata"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

3 Classification

For determining thresholds, record pairs within a given range of weights can be printed using `getPairs`. In this case, 24 is set as upper and −7 as lower threshold, dividing links, possible links and non-links. The summary shows the resulting contingency table and error measures.

```r
getPairs(pairs,30,20)
```

<table>
<thead>
<tr>
<th>id</th>
<th>fname_c1</th>
<th>fname_c2</th>
<th>lname_c1</th>
<th>lname_c2</th>
<th>by</th>
<th>bm</th>
<th>bd</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>457</td>
<td>URSULA</td>
<td>BIRGIT</td>
<td>MUELLER</td>
<td>NA</td>
<td>1940</td>
<td>6 15</td>
</tr>
</tbody>
</table>

1 The output of `getPairs` is shortened in this document.
pairs = emClassify(pairs, threshold.upper=24, threshold.lower=-7)
summary(pairs)

## Deduplication Data Set
## 500 records
## 571 record pairs
## 49 matches
## 522 non-matches
## 0 pairs with unknown status
##
## Weight distribution:
## [ -15, -10]  (-10, -5]  (-5, 0]  (0, 5]  (5, 10]  (10, 15]  
##        352      13       0       0       5      26
##          42      123       9       0       0       1
Review of the record pairs denoted as possible links is facilitated by `getPairs`, which can be forced to show only possible links via argument `show`. A list with the ids of linked pairs can be extracted from the output of `getPairs` with argument `single.rows` set to `TRUE`.

```r
possibles <- getPairs(pairs, show="possible")
possibles[1:6,]
```

```
#   id  fname_c1  lname_c1  by  bm  bd
# 1  17   ALEXANDER  MUELLER  1974  9  9
# 2 193  CHRISTIAN   MUELLER  1974  8  9
# 4  61   ANDRE     FISCHER  1943  6 25
# 5 254  STEFANIE   FISCHER  1943 11 25
# 6
```

```r
links=getPairs(pairs,show="links", single.rows=TRUE)
link_ids <- links[, c("id1", "id2")]
link_ids
```

```
#   id1 id2
# 290 290 466
# 50  50 234
# 87  87 117
# 145 145 240
# 286 286 383
```
## 289 289 399
## 297 297 388
## 357 357 414
## 313 313 457
## 467 467 472
## 183 183 444
## 25 25 107
## 106 106 175
## 370 370 478
## 127 127 142