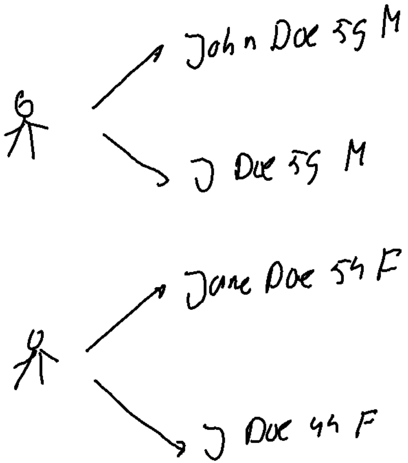
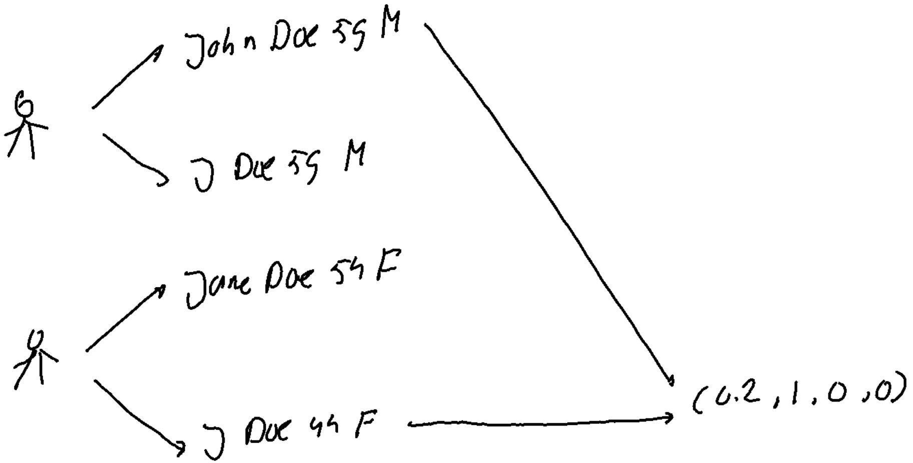
reclin: a package for probabilistic record linkage and deduplication

Jan van der Laan <dj.vanderlaan@cbs.nl>

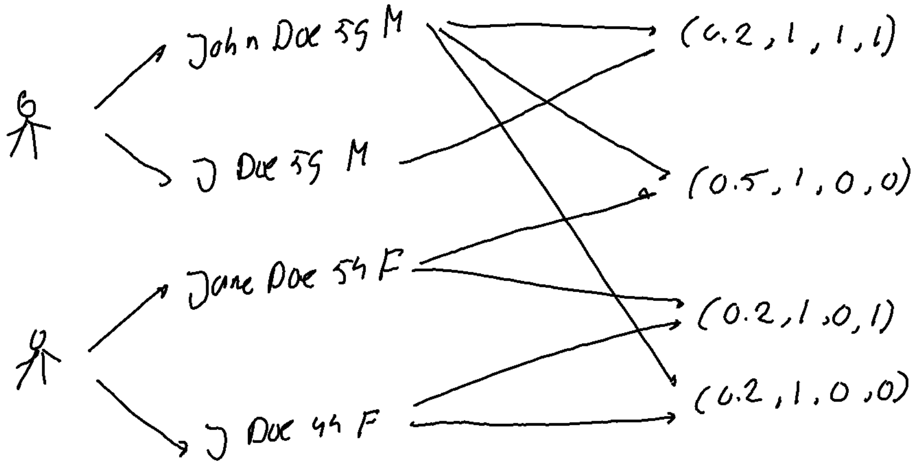
Overview record linkage



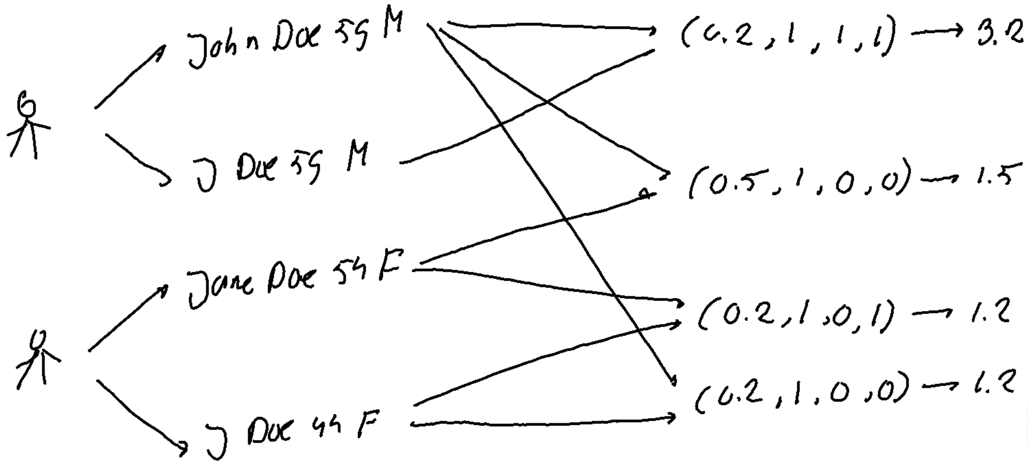
Overview record linkage



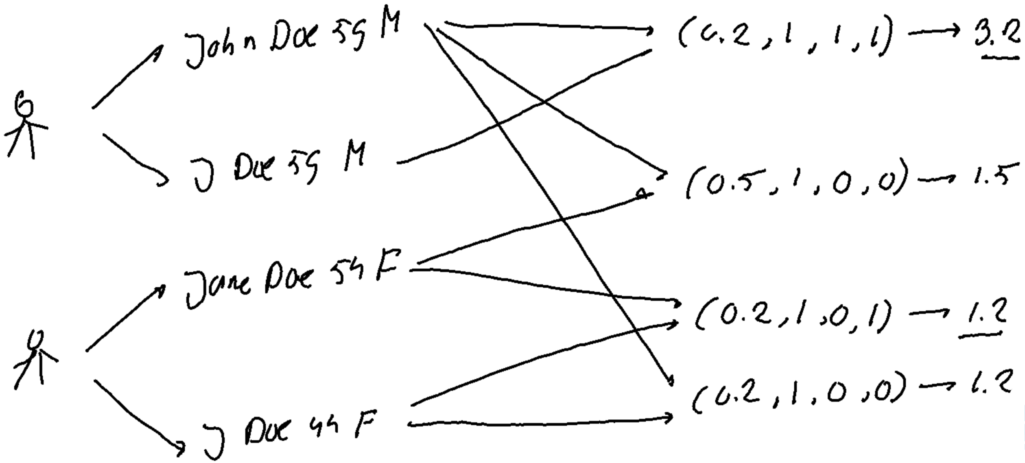
Overview record linkage



Overview record linkage



Overview record linkage



The record linkage process

1. Generate record pairs
   - Blocking
2. Generate comparison vectors
3. Translate comparison vector in a score measuring likelihood of both records in a pairs belonging to the same object.
   - *Classical* probabilistic linkage using EM
   - Machine learning
   - Simple scoring functions
4. Select pairs with a high enough score
5. Generate linked dataset
   - Force one-to-one linkage

Design goals

- Flexibility
  - toolbox: mix-and-match
  - should be simple to add new methods
- Speed
- Memory use

```
library(reclin)
data("linkexample1", "linkexample2")

#   id  lastname  firstname        address  sex  postcode
#1   1     Smith       Anna  12  Mainstr      F   1234  AB
#2   2     Smith     George  12  Mainstr      M   1234  AB
#3   3   Johnson       Anna  61  Mainstr      F   1234  AB
#  ...

#   id  lastname  firstname           address   sex  postcode
#1   2     Smith     Gearge  12  Mainstreet  <NA>   1234  AB
#2   3    Jonson         A.  61  Mainstreet     F   1234  AB
#3   4   Johnson    Charles      61  Mainstr     F   1234  AB
#  ...
```

```r
p <- pair_blocking(linkexample1, linkexample2, "postcode",
  large = FALSE)

p <- compare_pairs(p, by = c("lastname", "firstname",
    "address", "sex"),
  default_comparator = jaro_winkler(0.9))

# ...
# Showing all pairs:
#     x y lastname firstname   address sex
# 1   1 1 1.000000 0.4722222 0.9230769  NA
# 2   1 2 0.000000 0.5833333 0.8641026   1
# 3   1 3 0.447619 0.4642857 0.9333333   1
# 4   2 1 1.000000 0.8888889 0.9230769  NA
# 5   2 2 0.000000 0.0000000 0.8641026   0
# ...
```

```
m <- problink_em(p)

p <- score_problink(p, model = m, var = "weight")

# ...
# Showing all pairs:
#     x y lastname firstname   address sex     weight
# 1   1 1 1.000000 0.4722222 0.9230769  NA  7.7138545
# 2   1 2 0.000000 0.5833333 0.8641026   1 -6.8623638
# 3   1 3 0.447619 0.4642857 0.9333333   1  0.8024181
# 4   2 1 1.000000 0.8888889 0.9230769  NA  8.6108449
# 5   2 2 0.000000 0.0000000 0.8641026   0 -7.2330326
# ...
```

```
p <- select_n_to_m(p, "weight", var = "ntom", threshold = 0)

# ...
# Showing all pairs:
#     x y lastname firstname   address sex     weight  ntom
# 1   1 1 1.000000 0.4722222 0.9230769  NA  7.7138545 FALSE
# 2   1 2 0.000000 0.5833333 0.8641026   1 -6.8623638 FALSE
# 3   1 3 0.447619 0.4642857 0.9333333   1  0.8024181 FALSE
# 4   2 1 1.000000 0.8888889 0.9230769  NA  8.6108449  TRUE
# 5   2 2 0.000000 0.0000000 0.8641026   0 -7.2330326 FALSE
# 6   2 3 0.447619 0.5396825 0.9333333   0  0.7929395 FALSE
# ...
```

```
linked_data_set <- link(p)

#   id.x lastname.x firstname.x id.y lastname.y firstname.y ...
# 1    2      Smith      George    2      Smith      Gearge ...
# 2    3    Johnson        Anna    3     Jonson          A. ...
# 3    4    Johnson     Charles    4    Johnson     Charles ...
# 4    6   Schwartz         Ben    6   Schwartz         Ben ...
# 5    1      Smith        Anna   NA       <NA>        <NA> ...
# 6    5    Johnson      Charly   NA       <NA>        <NA> ...
# 7   NA       <NA>        <NA>    7   Schwartz        Anna ...
```

Future

New package `reclin2` in development

- Switch to `data.table`
- Faster and less memory use
- Cluster implementation: use multiple cores and even multiple machines
- Missing: memory mapped datasets
- Even simpler objects: easier to mess* with default algorithms

* fine tune

Contact and more information

Jan van der Laan <dj.vanderlaan@cbs.nl>

https://cran.r-project.org/package=reclin

https://github.com/djvanderlaan/reclin