Evaluation Metrics for Blocking/Entity Resolution

STA 325: Homework 2

General instructions for homeworks: Your code must be completely reproducible and must compile. No late homeworks will be accepted.

Reading Read the paper Binette and Steorts (2022) to get an overview of entity resolution. You'll want to refer to this during the course of the semester as it's meant to be a quick reference regarding the concepts that we will be covering. For more details, refer to the book by Christen (2012).

Advice: Start early on the homeworks and it is advised that you not wait until the day of as these homeworks are meant to be longer and treated as case studies.

Commenting code Code should be commented. See the Google style guide for questions regarding commenting or how to write code https://google.github.io/styleguide/Rguide.xml.

R Markdown Test

0. Open a new R Markdown file; set the output to HTML mode and "Knit". This should produce a web page with the knitting procedure executing your code blocks. You can edit this new file to produce your homework submission.

Total points on assignment: 2 (reproducibility) + 23 points for the assignment = 25 total points.

1. (4 points) What are the four main challenges of entity resolution?

1) Costly manual Labeling: Vast amounts of manually-labelled data are typically required for supervised learning and evaluation
2) Scalability/computational efficiency: Approximations are required to avoid quadratic scaling. Need to ensure impact on accuracy is minimal
3) Limited treatment of uncertainty: Given inherent uncertainties, it's important to output predictions with confidence regions
4) Unreliable evaluation: Standard evaluation methods return imprecise estimates of performance

2. (4 points, 1 point each) Suppose there are 10 records in a data set. a.) What are the total number of brute-forte comparison needed to make all-to-all record comparisons? The total number is n choose 2. In this case, n equals 10, so the total number is 45.

b.) Repeat this for 100 records, 1000 records, 10,000 records. The total number is n choose 2.

```
choose(100,2)
```

```
## [1] 4950
```

In the case that n equals 100, so the total number is 4950.

```
choose(1000,2)
```

```
## [1] 499500
```

In the case that n equals 1000, so the total number is 499500.

```
choose(10000,2)
```

```
## [1] 49995000
```

In the case that n equals 10000, so the total number is 49995000.

c.) What do you observe about the number of comparisons that need to be made?

The number of comparisons needed grows quadratically with the number of records n. As nn becomes large, the number of comparisons increases very rapidly. For instance, going from 1000 to 10,000 records increases the number of comparisons from 499,500 to 49,995,000, which is approximately a 100-fold increase. This quadratic growth implies that brute-force all-to-all comparisons become computationally expensive quickly as the size of the dataset increases.

3. (9 points) Consider the following record linkage data set with 1,000,000 total records that are matched between two databases. Assume that 500,000 are true matches. Assume a classified (or method) finds 600,000 record pairs as matches, and of these 400,000 correspond as true matches. The number of TP + FP + TN + FN = 50,000,000.

a. (4 points) Given the information above, find the following information in the confusion matrix: TP, FP, TN, and FN.
TP = Predicted true matches = 400,000
FP = Predicted matches - TP = 200,000
TN = 50,000,000 - TP- FP - FN = 49,300,000
FN = Total true matches - TP = 100,000

b. (1 point) Calculate the accuracy. Comment on the reliability of this metric for this problem.
Accuracy = (TP+TN)/(TP+FP+TN+FN)
= (400,000+49,300,000)/50,000,000
= 0.994

Comment on the reliability: The dataset has a significant imbalance between matches (500,000) and non-matches (49,500,000). The model could achieve high accuracy by mostly predicting the majority class instead of being truely effective.

c. (1 point) Calculate the precision.
Precision = TP/(TP+FP)
= 400,000/(400,000+200,000)
= 0.6666 = 66.67%

d. (1 point) Calculate the recall.
Recall = TP/(TP+FN)
= 400,000/(400,000+100,000)
= 0.8 = 80 %

e. (1 point) Calculate the f-measure.
f-measure = 2 X Precision X Recall / (Precision+Recall)
= 2 X 0.66667 X 0.8 / (0.6667 + 0.8)
= 72.7%

f. (1 point) Comment on the reliability of the precision, recall, and f-measure for this problem.

The precision means that 66.67% of predicted matches are actually true matches. It indicates that classifier is moderately reliable to reliability to identify true matches.

The recall means that 80% actual true matching pairs of records are correctly classified as matches. This indicates that the classifier are relatively strong at identify most of the true matches.

The f-measure of 72.7% indicates that the classifier made reasonable trade-off between precision and recall and overall entity resolution is relatively effective.

They are more reliable than accuracy because they account for the database imbalance between matches and non-matches.

4. (6 points) We will revisit the Italian Survey on Household and Wealth (SHIW) from class, which is a sample survey 383 households conducted by the Bank of Italy every two years (2008 and 2010). The data set is anonymized to remove first and last name (and other sensitive information).

```
library(italy)
library(assert)
data(italy08)
data(italy10)
knitr::opts_chunk$set(echo = TRUE,
                      fig.width=4,
                      fig.height=3,
                      fig.align="center")
head(italy08)
```

```
##        id PARENT SEX ANASC NASCREG CIT ACOM4C STUDIO Q QUAL SETT IREG
## 1 1040021      1   2  1948      16   1      0      5 1    2    3   16
## 2 1040022     10   2  1952      16   1      0      7 1    2    3   16
## 3 1110521      1   1  1972      20   1      2      5 1    1    4   20
## 4 1110522      3   1  1935      20   1      2      2 3    6    5   20
## 5 1110523      3   2  1941      20   1      2      3 3    6    5   20
## 6  119401      1   1  1941       7   1      0      4 3    6    5    7
```

```
head(italy10)
```

```
##        id PARENT SEX ANASC NASCREG CIT ACOM4C STUDIO Q QUAL SETT IREG
## 1 1040021      1   2  1948      16   1      0      5 3    6    5   16
## 2 1040022     11   2  1952      16   1      0      7 1    2    3   16
## 3 1110521      1   2  1941      20   1      2      3 3    6    5   20
## 4 1110522      2   1  1935      20   1      2      2 3    6    5   20
## 5 1110523      6   1  1972      20   1      2      5 1    1    4   20
## 6  119721      1   2  1948      16   1      2      2 2    5    4   17
```

```
id08 <- italy08$id
id10 <- italy10$id
id <- c(italy08$id, italy10$id) # combine the id
italy08 <- italy08[-c(1)] # remove the id
italy10 <- italy10[-c(1)] # remove the id
italy <- rbind(italy08, italy10)
head(italy)
```

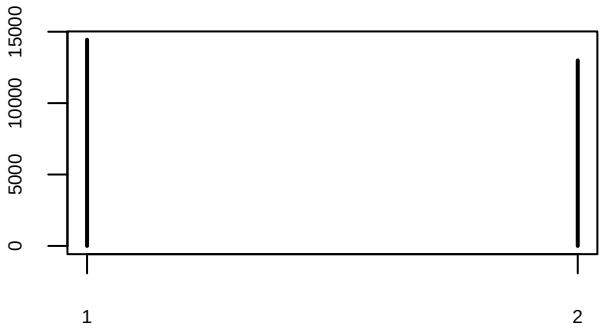
```
##   PARENT SEX ANASC NASCREG CIT ACOM4C STUDIO Q QUAL SETT IREG
## 1      1   2  1948      16   1      0      5 1    2    3   16
## 2     10   2  1952      16   1      0      7 1    2    3   16
## 3      1   1  1972      20   1      2      5 1    1    4   20
## 4      3   1  1935      20   1      2      2 3    6    5   20
## 5      3   2  1941      20   1      2      3 3    6    5   20
## 6      1   1  1941       7   1      0      4 3    6    5    7
```

a. (0 points) Please load the data set in the way that we did in class and block based upon gender.

```
# block by gender
blockByGender <- italy$SEX
```

b. (1 point) Plot the size of the blocks and comment on how many their are and their relative size.

```
# Plot the sizes of each block using a bar plot
recordsPerBlock <- table(blockByGender)
plot(recordsPerBlock, cex.axis=0.6, xlab="", ylab="")
```




```
head(recordsPerBlock)
```

```
## blockByGender
##     1     2
## 14442 12993
```

In gender 1 block, there are 14442. In gender 2 block, there are 12993. They are relatively similar and balanced.

c. (1 point) Calculate the reduction ratio and interpret its meaning.

```
#Function for Reduction Ratio
reduction.ratio <- function(block.labels) {
n_all_comp = choose(length(block.labels), 2)
n_block_comp = sum(choose(table(block.labels), 2))
(n_all_comp - n_block_comp) / n_all_comp
}

#Calculate Reduction Ratio
reduction.ratio(blockByGender)
```

```
## [1] 0.4986234
```

Blocking by gender eliminated 49.86% of comparisons.

d. (2 points) Calculate the precision and recall. Interpret the meaning of each.

```
#Function for Precision
precision <- function(block.labels, IDs) {
  ct = xtabs(~block.labels+IDs)
  # Number of true positives
  TP = sum(choose(ct, 2))
  # Number of positives = TP + FP
  P = sum(choose(rowSums(ct), 2))
  return(TP/P)
}

#Calculate Precision
precision(blockByGender,id)
```

```
## [1] 3.599727e-05
```

```r
#Function for Recall
recall <- function(block.labels, IDs) {
  ct = xtabs(~IDs+block.labels)
  # Number of true positives
  TP = sum(choose(ct, 2))
  # Number of true links = TP + FN
  TL = sum(choose(rowSums(ct), 2))
  return(TP/TL)
}

#Calculate Recall
recall(blockByGender,id)
```

```
## [1] 0.9113109
```

The precision indicates that 0.0036% of record pairs that are classified as matches correspond to true matches, while the rest correspond to false matches.

The recall indicates that 91.11% of the true matching record pairs are correctly classified as matches.

e. (1 point) Would this be a reasonable approach for blocking. Explain. Blocking by gender is reasonable approach for blocking but might not be the best.

Blocking by gender, in this case, is a reasonable approach to reduce the number of comparisons even though it has a relatively high reduction ratio (49.86%). This significant reduction can make the process more computationally feasible.

f. (1 point) Would blocking on gender be recommended for entity resolution. Explain.

Blocking by gender would not be recommended even though it can significantly reduce the number of record comparisons. As indicated by the low precision (0.0036%), most of the record pairs identified as matches are not true matches, suggesting that the method generates many false positives and can lead to inefficiencies in the matching process.