Methods for Risk Evaluation part 2

Jingchen (Monika) Hu

Vassar College

Statistical Data Privacy

Outline

Outline

Recap

- Lecture 8:
  - Identification disclosure
  - Matching-based method

Plan for this lecture

- Two general types of disclosure: identification and attribute (Lecture 1)

1. Identification disclosure: The intruder correctly identifies records of interest in the released synthetic data
2. Attribute disclosure: The intruder correctly infers the true confidential values of the synthetic records using information from the released synthetic data

- In this lecture, we focus on identification risk evaluation methods, with illustrations to the synthetic CE from Lectures 4 & 5 and the synthetic ACS from Lecture 5

Overview

- Identification disclosure:
  - The intruder correctly identifies records of interest in the released synthetic data
  - Only exist in partially synthetic data
- We will introduce two general approaches
  - Matching-based approaches (last lecture)
  - Record linkage approaches

Outline

Overview

- Record linkage methods are developed mainly for the purpose of linking records from multiple databases
- Based on variables, called **keys**, a link between two records can be established. Therefore, record linkage approaches can be used as metrics of identification risks (Winkler (2004))

Record linkage approaches for synthetic data

- For partially synthetic data, record linkage methods can be applied to linking records in the synthetic dataset to the records in the confidential dataset
- Among these linkages, we can evaluate identification risks in terms of true links (i.e., correct links) and false links (i.e., incorrect links)
  - high percentage of true links / low percentage of false links indicates high identification disclosure risk, and vice versa

Record linkage approaches for synthetic data

- We now present the general procedure of performing probabilistic record linkage (Fellegi and Sunter (1969)) to evaluate identification disclosure risks in partially synthetic data
- As with matching-based approaches
  - $\mathbf{Y} = (\mathbf{Y}^A, \mathbf{Y}^U)$ to represent the confidential data sample containing n observations and r variables
  - $\mathbf{Y}^A$ denotes the variables available to the intruder from external databases and $\mathbf{Y}^U$ denotes the variables unavailable to the intruder
  - similarly, we have $\mathbf{Z} = (\mathbf{Z}^A, \mathbf{Z}^U)$ for a partially synthetic dataset of $\mathbf{Y}$
  - we can further split $\mathbf{Z}^A$ into $\mathbf{Z}^{A_s}$, the synthesized variables and $\mathbf{Z}^{A_{us}}$, the unsynthesized variables

Procedure

1. Given $\mathbf{Y}^A$, the set of variables available to the intruder, we generate pairs between $\mathbf{Y}$ and $\mathbf{Z}$ based on $\mathbf{Y}^A$ and $\mathbf{Z}^A$
   - that is, a pair of record i from $\mathbf{Y}$ and record j from $\mathbf{Z}$ is generated only when $\mathbf{y}_i^A = \mathbf{z}_j^A$ (for the entire vector)
   - note that since $\mathbf{Z}^A = (\mathbf{Z}^{A_s}, \mathbf{Z}^{A_{us}})$, some of these pairs would be incorrect, thanks to the changes brought by the data synthesis process
   - call this collection of pairs as P.
2. For each pair of records in P, we, the data disseminators, compare the values of the unavailable variables
   - for example, if $\mathbf{y}_i$ and $\mathbf{z}_j$ is paired up in step 1, then this step compares the values of $\mathbf{y}_i^U$ and $\mathbf{z}_j^U$
   - we can create a set of similarity score over all the unavailable variables, which will be used for scoring all pairs next.

Procedure

3. With calculated similarity score for each pair, we then score all the pairs
    - think of this as ranking all pairs
    - the higher the ranking one pair is, the more likely a link will be established
    - this is the core of probabilistic record linkage as proposed by Winkler (2000), where we will use an expectation-maximization algorithm (**EM** algorithm)
    - in this approach, a weight value will be estimated for each pair, which will then be used next for determining links, also known as selecting pairs

Procedure

4. Each pair now comes with a weight from step 3. We then select one-to-one linkages between records in $\mathbf{Y}$ and records in $\mathbf{Z}$
   - that is, each record in the synthetic dataset $\mathbf{Z}$ will be linked to at most one record in the confidential $\mathbf{Y}$, and vice versa
5. Among the selected pairs from step 4, we calculate the percentages of true links (i.e., the one-to-one links that are correct links) and of false links (i.e., the one-to-one links that are incorrect links)

Example of the ACS sample

- Now we will illustrate this record linkage approach to evaluating identification disclosure risks in the synthetic ACS sample
- The probabilistic record linkage algorithm is implemented by the `reclin` package (Laan (2018))

```
## make sure to load the reclin package
library(reclin)
```

Example of the ACS sample

We use the record linkage approach to evaluate the identification disclosure risks of a synthetic ACS sample in Lecture 5, where `DIS, HICOV` are synthesized and the other variables remain unsynthesized. The synthesis model is the DPMPM model with the NPBayesImputeCat R package. We assume assume that the intruder knows `SEX, RACE, MAR` of reach record.

Example of the ACS sample

- Load datasets

```
ACSdata <- data.frame(readr::read_csv(file = "ACSdata.csv"))
n <- dim(ACSdata)[1]
ACSdata_syn <- data.frame(readr::read_csv(file = "ACSdata_syn.csv"))
## make sure variables are in the same ordering
ACSdata_syn <- ACSdata_syn[, names(ACSdata)]
## add index for each record
ACSdata$id <- 1:n
ACSdata_syn$id <- 1:n
```

Example of the ACS sample: generate pairs given available variables

- We first generate pairs given the available variables SEX, RACE, MAR
- This is done by using the pair_blocking() function in the reclin package, where the inputs include ACSdata_syn, ACSdata, and the set of available variables
- The collection of pairs is stored in ACS_pairs

```
ACS_pairs <- reclin::pair_blocking(ACSdata_syn, ACSdata,
                                   c("SEX", "RACE", "MAR"))
```

Example of the ACS sample: compare pairs based on unavailable variables

- Next, for each generated pair in `ACS_pairs`, we create a set of similarity score over all the unavailable variables: the synthesized `DIS`, `HICOV` and the unsythesized `LANX`, `WAOB`, `MIG`, `SCH`, `HISP`
- For this step, we use the `compare_pairs()` function in the `reclin` package and we use `jaro_winkler` similarity score (Laan (2018))

```
ACS_pairs_keys <- reclin::compare_pairs(ACS_pairs, by = c("LANX", "WAOB",
                                                          "DIS", "HICOV",
                                                          "MIG", "SCH",
                                                          "HISP"),
                                        default_comparator =
                                          jaro_winkler(0.9))
```

Example of the ACS sample: compare pairs based on unavailable variables

- For illustration purpose, we print out the first few rows of `ACS_pairs_keys`, where `x` and `y` are the record indexes from the two datasets, and each variable column shows a similarity score for that pair on that variable.

```
ACS_pairs_keys[1:3, ]
```

```
## ldat with 3 rows and 9 columns
##   x y LANX WAOB DIS HICOV MIG SCH HISP
## 1 1 1    1    1   1     1   1   1    1
## 2 1 5    1    1   1     1   0   1    1
## 3 1 8    1    1   0     1   0   1    1
```

Example of the ACS sample: score all pairs with EM and produce weights

- We then use the probabilistic record linkage approach with an EM algorithm to produce a weight for each scored pair
- The higher the weight is, the more likely the pair of records belong to the same record
- The EM algorithm is implemented using the `problink_em()` function and the weights are then calculated using the `score_problink()` function in the `reclin` package

```
m <- reclin::problink_em(ACS_pairs_keys)
ACS_pairs_keys_pRL <- reclin::score_problink(ACS_pairs_keys,
                                             model = m,
                                             var = "weight")
```

Example of the ACS sample: score all pairs with EM and produce weights

- The printout of the first few rows of `ACS_pairs_keys_pRL` shows an additional column, `weight`, which corresponds to the calculated weights for all pairs after the EM algorithm for the probabilistic record linkage procedure

```
ACS_pairs_keys_pRL[1:3, ]
```

```
## ldat with 3 rows and 10 columns
##   x y LANX WAOB DIS HICOV MIG SCH HISP   weight
## 1 1 1    1    1   1     1   1   1    1 6.079087
## 2 1 5    1    1   1     1   0   1    1 5.718985
## 3 1 8    1    1   0     1   0   1    1 6.184407
```

Example of the ACS sample: select one-to-one linkag

- Now with calculated weight for each pair, we perform a one-to-one linkage by comparing weights for all pairs while making sure that one record from the synthetic `ACSdata_syn` can be linked to at most one record from the confidential `ACSdata` and vice versa
- There are a few choices provided by the `reclin` package, and for our data size, we choose to use the greedy algorithm with the `select_greedy()` function

```
ACS_pairs_keys_pRL <- reclin::select_greedy(ACS_pairs_keys_pRL, "weight",
                                            var = "greedy", threshold = 0)
```

Example of the ACS sample: select one-to-one linkag

- This process adds one more column `greedy` to `ACS_pairs_keys_pRL` which shows TRUE / FALSE: TRUE indicates a link and FALSE indicates no link

```
ACS_pairs_keys_pRL[1:3, ]
```

```
## ldat with 3 rows and 11 columns
##   x y LANX WAOB DIS HICOV MIG SCH HISP   weight greedy
## 1 1 1    1    1   1     1   1   1    1 6.079087  FALSE
## 2 1 5    1    1   1     1   0   1    1 5.718985  FALSE
## 3 1 8    1    1   0     1   0   1    1 6.184407  FALSE
```

Example of the ACS sample: calculate percentages of true links and false links

- Lastly, we need to evaluate among all the TRUE's in `greedy`, how many of them are true links and how many are false links
- A true link refers to a correct linkage, i.e., record i in `ACSdata_syn` is correctly linked to record i in `ACSdata`
- A false link refers to an incorrect linkage, i.e., record i in `ACSdata_syn` is incorrectly linked to record j in `ACSdata` where $i \neq j$
- To do so, we create a new column `true` by comparing the ID's of each pair
- We add the ID's from `ACSdata_syn` and those from `ACSdata` using the `add_from_x()` and `add_from_y()` functions respectively, and then compare if they are the same

```
ACS_pairs_keys_pRL <- add_from_x(ACS_pairs_keys_pRL, id_x = "id")
ACS_pairs_keys_pRL <- add_from_y(ACS_pairs_keys_pRL, id_y = "id")
ACS_pairs_keys_pRL$true <- ACS_pairs_keys_pRL$id_x ==
  ACS_pairs_keys_pRL$id_y
```

Example of the ACS sample: calculate percentages of true links and false links

- Lastly, we tabulate the `true` and `greedy` columns, as below.

```
table(ACS_pairs_keys_pRL[c("true", "greedy")])
```

```
##         greedy
## true        FALSE     TRUE
##   FALSE 11858692     9266
##   TRUE      9266      734
```

Discussion question: What do the results show us?

Example of the ACS sample: results of the confidential data

- See hidden R scripts to evaluate the results on the confidential data
- The true linkage percentage is $6458/10000 = 64.58\%$, and the false linkage percentage is therefore $3542/10000 = 35.42\%$

Discussion question: How do the synthetic data provide privacy protection compared to the confidential data?

```
##          greedy
## true         FALSE     TRUE
##    FALSE 11864416     3542
##    TRUE       3542     6458
```

Final comments

- Note that in our illustration, all the available variables, `SEX`, `RACE`, `MAR`, are unsynthesized, so our first step of generating pairs would have no errors
- It is possible that the intruder's knowledge of available variables includes some synthesized variables, which means the first step would generate incorrect pairs

Outline

Summary

- Record linkage approaches for identification disclosure risk evaluations
  - the `reclin` R package
  - the true linkage percentage and the false linkage percentage

Summary

- Record linkage approaches for identification disclosure risk evaluations
  - the `reclin` R package
  - the true linkage percentage and the false linkage percentage
- No homework! But you should be working on disclosure risk evaluation for your project
- Lecture 10: Methods for risk evaluation part 3
  - Baillargeon and Charest (2020) (CAP statistic)

References I

Baillargeon, M., and A. Charest. 2020. "A Closer Look at the CAP Risk Measure for Synthetic Datasets." Privacy in Statistical Databases (E-Proceedings).

Fellegi, I. P., and A. B. Sunter. 1969. "A Theory for Record Linkage." Journal of the American Statistical Association 64 (328): 1183–1210.

Laan, J. van der. 2018. Record Linkage Toolkit. R Package Version 0.1.1.

Winkler, W. E. 2000. "Using the Em Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage." U.S. Bureau of the Census.

Winkler, William E. 2004. "Re-Identification Methods for Masked Microdata." In Privacy for Statistical Databases, edited by J. Domingo-Ferrer and V. Torra, 216–30.