Advanced Survey Statistics: Disclosure Control

Part 6: Utility

Matthias Templ

Institut für Datenanalyse und Prozessdesign
School of Engineering
Zürcher Hochschule für Angewandte Wissenschaften

FU-Berlin, 2019

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw
School of
Engineering
IDP Institut für Datenanalyse
und Prozessdesign

Measuring the data utility

complementary approaches to assess information loss:

- distances between the original data and perturbed data
- comparing statistics computed on the original and perturbed data.
    - gerneral purpose statistics
    - data-specific measures

Comparing missing values

Let $\mathbf{R}^{(X)}$ and $\mathbf{R}^{(Y)}$ be indicator matrices of the same size as $\mathbf{X}$ (original data) and $\mathbf{Y}$ (anonymized data of the same size) with n observations and p variables. A cell/element of $\mathbf{R}^{(X)}$ is 1 when $\mathbf{X}$ has a missing value on that position, otherwise 0 (same for $\mathbf{R}^{(Y)}$).

$$\tilde{r}_{ij} = \begin{cases} 0 & \text{if } r_{ij}^{(X)} = r_{ik}^{(Y)} = 0 \quad , \\ 0 & \text{if } r_{ij}^{(X)} = 1 \ \wedge \ r_{ij}^{(Y)} = 1 \quad , \\ 1 & \text{if } r_{ij}^{(X)} = 0 \ \wedge \ r_{ij}^{(Y)} = 1 \quad , \\ 0 & \text{if } r_{ij}^{(X)} = 1 \ \wedge \ r_{ij}^{(Y)} = 0 \quad . \end{cases}$$

Comparing missing values

Number of additional missings per variable caused by anonymizing the data using the indicator matrix **R** with *n* observations and *p* variables,

$$m_j = \sum_i^n \tilde{r}_{ij} \quad , \; j \in \{1, \ldots, p\} \quad .$$

Relative measure:

$$mp_j = 100 \cdot \frac{m_j}{n} \quad .$$

The higher m_j (or mp_j) the higher the information loss.

Comparing missing values in R

```r
library("laeken"); library("sdcMicro")
data("eusilc")
sdc <- createSdcObj(eusilc,
          keyVars = c("db040", "hsize", "pb220a",
                      "rb090"),
          weightVar = "rb050", hhId = "db030")
sdc <- kAnon(sdc) # local suppression produces additional
print(sdc, "ls")
```

```
## Local suppression:
##  KeyVar | Suppressions (#) | Suppressions (%)
##   db040 |                0 |            0.000
##   hsize |                9 |            0.061
##  pb220a |                0 |            0.000
##   rb090 |                0 |            0.000
## ---------------------------------------------------------
```

Comparing contigency tables

Contingency table $\mathbf{T}^{(\mathbf{X})}$ calculated from categorical variables of the original data $\mathbf{X}$ and the contingency table $\mathbf{T}^{(\mathbf{Y})}$ from the anonymized data $\mathbf{Y}$.

$$UT = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left| T_{ij}^{(\mathbf{X})} - T_{ij}^{(\mathbf{Y})} \right| \quad . \tag{1}$$

The higher UT the lower the data quality.

Comparing contigency tables

Relative change in each cell (in percentages).

$$UT2 = 100 \cdot \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left| \frac{T_{ij}^{(\mathbf{X})} - T_{ij}^{(\mathbf{Y})}}{T_{ij}^{(\mathbf{X})}} \right| \quad . \qquad (2)$$

In the following code, a contingency table of $rb090 \times db040$ (gender × federal state) is computed for the anonymized data and original data. First, an object of class *sdcMicroObj* is created, and then we apply PRAM on federal state.

Let's start from the beginning. First we create the `sdcMicro` object, then we apply PRAM and compare the original and table considering the prammed variable `db040` (region).

Comparing contigency tables in R

```r
X <- Y <- eusilc
sdc <- createSdcObj(X,
          keyVars = c("db040", "hsize", "pb220a",
                      "rb090", "pl030", "age"),
          numVars = "eqIncome",
          pramVars = "db040",
          weightVar = "rb050", hhId = "db030")
sdc <- pram(sdc)
```

```
## Warning in pramX(obj = obj, variables = variables, strat
```

```r
Y <- extractManipData(sdc)
```

Comparing contigency tables in R

We now compare the tables according to Equation~(2).

```
ct <- c("rb090", "db040")
Tx <- table(X[, ct])
Ty <- table(Y[, ct])
Tx
```

```
##         db040
## rb090    Burgenland Carinthia Lower Austria Salzburg Sty
##   male          261       517          1417      440   1
##   female        288       561          1387      484   1
##         db040
## rb090    Upper Austria Vienna Vorarlberg
##   male            1363   1132        359
##   female          1442   1190        374
```

Comparing contigency tables in R

```r
n1 <- nrow(Ty); n2 <- ncol(Ty)
## UT
sum(abs(Tx - Ty)) / (n1 * n2)
```

```
## [1] 9.222222
```

```r
## UT2
sum(abs(Tx - Ty)/Tx) / (n1 * n2) * 100
```

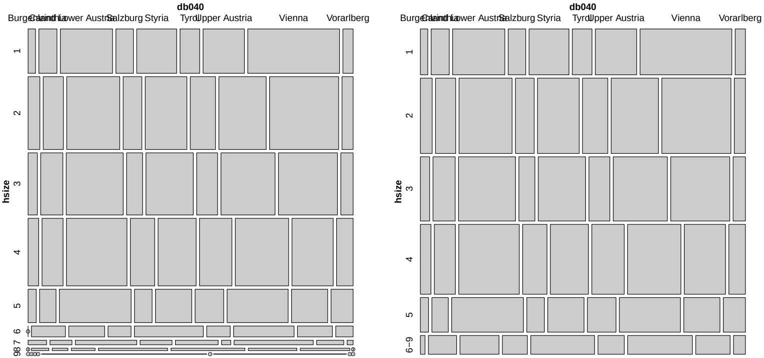
```
## [1] 1.516536
```

→ mean difference in the cell values of the tables is approximately 1.127%.

Comparing contigency tables, visually

```r
require(vcd)
ct <- c("rb090", "pb220a", "hsize")
library(simPop)
Tx <- tableWt(X[, ct], X$rb050)
Ty <- tableWt(Y[, ct], X$rb050)
par(mfrow=c(1,2))
mosaic(Tx); mosaic(Ty)
```

Comparing contigency tables, visually



Mosaic plot of gender (`rb090`) × citizenship (`pb220a`) × household size (`hsize`) showing the original sample frequencies (left plot) and the sample frequencies from the perturbed data (right plot).

Comparing continuous key variables

- IL1s can be interpreted as the scaled distances between original and perturbed values. Again let $\mathbf{X} = \{x_{ij}\}$ be the original data set, $\mathbf{Y} = \{y_{ij}\}$ is a perturbed version of $\mathbf{X}$. Both data sets consist of n observations and p variables each. The measure of information loss is defined by

$$IL1 = \frac{1}{pn} \sum_{j=1}^{p} \sum_{i=1}^{n} \frac{|x_{ij} - y_{ij}|}{\sqrt{2} S_j} \quad ,$$

where S_j is the standard deviation of the j-th variable in the original data set.

Comparing continuous key variables

- **prediction quality** measures the differences between estimates obtained from fitting a pre-specified regression model on the original data and the perturbed data:

  $$|(\bar{\hat{y}}_w^o - \bar{\hat{y}}_w^m)/\bar{\hat{y}}_w^o| \quad ,$$

  with $\bar{\hat{y}}_w$ being fitted values from a pre-specified model obtained from the original (index o) and the modified data (index m). Index w indicates that the survey weights should be considered when fitting the model.

```r
sdc <- microaggregation(sdc)
get.sdcMicroObj(sdc, "utility")
```

```
## $il1
## [1] 638.4001
##
## $il1s
## [1] 10.69445
```

Mixed-scaled

→ book

Entropy

The alternative option is to use an entropy function. Given $c_1, c_2, \ldots, c_k$ categories of a variable $\mathbf{X_j}$, the entropy E_{c_j} is defined as

$$E_{c_j} = -\frac{1}{n} \sum_{c_j \in \mathbf{X_j}} f_{c_j} log \left(\frac{f_{c_j}}{n} \right) \quad , \tag{3}$$

where f_{c_j} is the frequency of category c_j of variable $\mathbf{X}_j$ and n the total number of observations.

- Useful when recoding is done in an (semi-)automatized manner
- Useful when the choice of suppression variables is done in an automized manner

```
## entropy of key variables on original data X
entropy <- function(fk, n){
  (-1) * 1 / n  * sum(fk * log(fk / n))
}
```

Entropy

```r
## for hsize
n <- nrow(eusilc)
fk <- as.numeric(table(eusilc$hsize))
entropy(fk, n)
```

```
## [1] 1.765339
```

```r
## for age
entropy(as.numeric(table(eusilc$age)), n)
```

```
## [1] 4.440551
```

```r
## for pb220a
entropy(as.numeric(table(eusilc$pb220a)), n)
```

```
## [1] 0.4446661
```

Propensity Scores

- Rowbind **X** (*n* observations) and **Y** (*m* observations)
- Create indicator response variable that expresses memberships of observations to **X** and **Y**
- logistic regression using the indicator variable as response
- predict prob. $p_i \quad, i = 1, .., n + m$ of the indicator variable
- Look at the differences

$$UP = \frac{1}{n+m} \sum_{i=1}^{n+m} (p_i - c)^2 \quad ,$$

where p_i is the estimated probabilities being in group 1 (original data) or group 2 (perturbed data). c is usually determined as 0.5.

  - If UP is close to zero, the data utility is high.
  - worst case: $UP \sim 1/4$, the two data sets are completely distinguishable

Propensity Scores in R

```r
Z <- rbind(eusilc, extractManipData(sdc))
Z$index <- rep(0:1, each=nrow(eusilc))
form <- as.formula("index ~ db040 + hsize + pb220a +
                   rb090 + pl030 + eqIncome")
res <- glm(form, data=Z, family = binomial())
1 / nrow(Z) * sum((predict(res, type="response") - 0.5)^2)
```

```
## [1] 5.815102e-06
```

→ data utility is high.

Data-specific utility measures

If you release the data, what users of the data will analyse? Determine the most important variables of the micro data set and take the most important indicators into account. Steps:

1. selection of a set of (benchmarking) indicators;
2. estimation of all benchmarking indicators based on the original micro data;
3. estimation of the benchmarking indicators based on the protected micro data set;
4. comparison of statistical properties such as point estimates, variances or overlaps in confidence intervals for each benchmarking indicator, regression coefficients, . . . ;
5. assessment of the data utility of the protected micro data set

Example EU-SILC

Important indicators such as the Gini coefficient, the at-risk-at-poverty rate, ...

Given a vector $x_1, \ldots, x_n$ with sample weights $w_1, \ldots, w_n$ the Gini coefficient can be estimated by

$$\widehat{Gini} = \frac{2\sum_{i=1}^{n}\left(w_i x_i \sum_{j=1}^{i} w_i\right) - \sum_{i=1}^{n} w_i^2 x_i}{\left(\sum_{i=1}^{n} w_i\right)\sum_{i=1}^{n} w_i x_i} - 1 \quad .$$

The Gini coefficient takes on values between 0 and 1. A value of 0 stand for perfect equality

Example EU-SILC

Point estimates, relative difference in percent:

```
sdc@additionalResults$gini <- gini(inc = "eqIncome",
       weigths = "rb050",
       breakdown = "db040",
       data = extractManipData(sdc))$valueByStratum$value
```

```
res <- gini(inc = "eqIncome",
            weigths = "rb050",
            breakdown = "db040",
            data = eusilc)$valueByStratum$value
100*abs((res - sdc@additionalResults$gini)/res)
```

```
## [1] 4.1542026 0.3620482 0.1297038 0.8987387 1.9085991 0.
## [8] 0.3149498 0.4979061
```

Example EU-SILC

variance estimates

```
res <- gini(inc = "eqIncome",
            weigths = "rb050",
            breakdown = "db040", # region
            data = eusilc)
resVar <- variance("eqIncome", weights = "rb050",
            design = "db040", breakdown = "db040",
            data = eusilc, indicator = res, R = 50,
            X = calibVars(eusilc$db040), seed = 123)
res <- resVar$valueByStratum$value
resVar <- resVar$varByStratum$var
```

Example EU-SILC

variance estimates

```
eusilcA <- extractManipData(sdc)
resA <- gini(inc = "eqIncome",
           weigths = "rb050",
           breakdown = "db040",
           data = eusilcA)
resVarA <- variance("eqIncome", weights = "rb050",
            design = "db040", breakdown = "db040",
            data = eusilcA, indicator = resA, R = 50,
            X = calibVars(eusilc$db040), seed = 123)
resA <- resVarA$valueByStratum$value
resVarA <- resVarA$varByStratum$var
```

Example EU-SILC

variance estimates: relative differences in percent

```
100*abs((res - resA) / res)
```

```
## [1] 4.1542026 0.3620482 0.1297038 0.8987387 1.9085991 0.
## [8] 0.3149498 0.4979061
```

```
100*abs((resVar - resVarA) / resVar)
```

```
## [1] 24.465329 16.643231 48.145491 19.610998 31.471014  4
## [8] 25.425715  9.981877
```

Some kind of MSE:s

```
(res - resA)^2 + abs(resVar - resVarA) # MSE
```

```
## [1] 2.01481543 0.08652782 0.07022015 0.18781366 0.283108
## [7] 0.16740108 0.07589927 0.11497837
```

Alternative: Overlap of confidence intervals

Conclusions

- **general purpose measures** such as IL1s, differences in means, outcome of multivariate statistical methods, propensity scores, distances, etc. are useful for giving a quick answer about the utility of the anonymized data set.
- **data- and context-specific utility measures** such as useful regression models, the most interesting indicators, . . .
  - gives more trustful indication on utility as general purpose measures
  - needs a lot of time to get sure about the user needs