PWSCUP2020ソースコード

2020/8/13 NTT長谷川，千田

変更 2020/8/26 NTT長谷川，千田

\* 目次

\*\* ファイルの中身

\*\* 各種作り方

\* ファイルの中身

- gen.py : 擬似データ(B)を作成するためのコード

- synthetic.py : 擬似データ生成コード(gen.pyとほぼ同じ. 匿名化データ(D)の作成に利用)

- randomsampling.py : サンプリングデータ(C)を作成するためのコード

- utilityfunc.py : サンプリングデータ(C)と匿名化データ(D)との比較をし有用性を評価するコード

- attack.py : 擬似データ(B)と匿名化データ(D)を用いてmembership推定攻撃を実施するコード. randomsampling.pyで生成したxxx.indexファイルを用いると攻撃成功率を評価できる.

- rr.py, rrp.py : 匿名化データ(D)を作成するためのコード. Randomized Responseによる加工.

- kanony.py : 匿名化データ(D)を作成するためのコード. レコード削除k-匿名化による加工.

- census\_income.data.csv : オリジナルデータ(A) ※擬似データ生成用の元データ

+ Census Income Data Set (https://archive.ics.uci.edu/ml/datasets/census+income) の adult.data (32,561レコード, 15属性) を加工

+ 欠損値を含むレコードを削除 32,561 → 30,162レコード

+ 以下の9属性のみ使用 age, workclass, education, marital-status, occupation, relationship, sex, hours-per-week, income

- trial\_syntheticdata.csv : お試し用擬似データ(B)

- trial\_samplingdata.csv : お試し用サンプリングデータ(C)

- trial\_answer.index : お試し用サンプリングデータの正解行(C')

- trial\_anonymizeddata.csv : お試し用匿名化データ(D)

- test.csv : 匿名化データの有用性評価用テストデータ

+ Census Income Data Set (https://archive.ics.uci.edu/ml/datasets/census+income) の adult.test (16,281レコード, 15属性) を加工

+ 欠損値を含むレコードを削除 16,281 → 15,060レコード

+ 以下の9属性のみ使用 age, workclass, education, marital-status, occupation, relationship, sex, hours-per-week, income

\* 擬似データ(B)の作り方

$ python gen.py census\_income.data.csv False True trial\_syntheticdata.csv 0

# census\_income.data.csv は入力データ名 (オリジナルデータ(A))

# Falseはヘッダの有無. 今回は無いのでFalse

# Trueはカンマ区切りの後の空白を除去するかどうか. Trueを指定

# trial\_syntheticdata.csv は出力データ名

# 0は乱数のシード.

\* サンプリングデータ(C)の作り方

$ python randomsampling.py trial\_syntheticdata.csv False True 0.1 0 trial\_samplingdata.csv trial\_answer.index

# trial\_syntheticdata.csv は入力データ名 (擬似データ(B))

# Falseはヘッダの有無. 今回は無いのでFalse

# Trueはカンマ区切りの後の空白を除去するかどうか. Trueを指定

# 0.1はサンプリング率. 10%サンプリングなので0.1

# 0は乱数のシード.

# trial\_samplingdata.csv は出力データ名

# trial\_answer.index は trial\_syntheticdata.csvの何行目(0～99999)をサンプリングしたか分かる行番号を記した出力データ名 (サンプリングデータの正解行(C'))

\* 匿名化データ(D)の作り方 ※匿名化サンプルコード1

$ python synthetic.py trial\_samplingdata.csv False True 10000 trial\_anonymizeddata1.csv 0

# trial\_samplingdata.csv　は入力データ名 (サンプリングデータ(C))

# Falseはヘッダの有無. 今回は無いのでFalse

# Trueはカンマ区切りの後の空白を除去するかどうか. Trueを指定

# 10000 は出力レコード数

# trial\_anonymizeddata1.csv は出力データ名

# 0は乱数のシード.

\* 匿名化データ(D)の作り方 ※匿名化サンプルコード2

$ python rr.py trial\_samplingdata.csv 0.1 trial\_anonymizeddata2.csv 0\_1\_2\_3\_4\_5\_6\_7\_8

# trial\_samplingdata.csv　は入力データ名 (サンプリングデータ(C))

# 0.1 はrr手法で用いるパラメータ(Randomized Responseの維持確率に相当)

# trial\_anonymizeddata2.csv は出力データ名

# 0\_1\_2\_3\_4\_5\_6\_7\_8 はrrの対象となる列番号. 0〜8列すべてに対して行う場合は0\_1\_2\_3\_4\_5\_6\_7\_8. 2と3だけであれば, 2\_3

\* 匿名化データ(D)の作り方 ※匿名化サンプルコード3

$ python rrp.py trial\_samplingdata.csv 0.1 trial\_anonymizeddata3.csv 0\_1\_2\_3\_4\_5\_6\_7\_8

# trial\_samplingdata.csv　は入力データ名 (サンプリングデータ(C))

# 0.1 はrrp手法で用いるパラメータ(Randomized Responseの維持確率に相当)

# trial\_anonymizeddata3.csv は出力データ名

# 0\_1\_2\_3\_4\_5\_6\_7\_8 はrrpの対象となる列番号. 0〜8列すべてに対して行う場合は0\_1\_2\_3\_4\_5\_6\_7\_8. 2と3だけであれば, 2\_3

\* 匿名化データ(D)の作り方 ※匿名化サンプルコード4

$ python kanony.py trial\_samplingdata.csv 2 trial\_anonymizeddata4.csv 0\_1\_2\_3\_4\_5\_6

# trial\_samplingdata.csv　は入力データ名 (サンプリングデータ(C))

# 2 はk-匿名化のkの値

# trial\_anonymizeddata4.csv は出力データ名

# 0\_1\_2\_3\_4\_5\_6 は準識別子とする列番号. 0〜8列すべてに対して行う場合は0\_1\_2\_3\_4\_5\_6\_7\_8. 2と3だけであれば, 2\_3

\* 有用性評価の仕方

$ python utilityfunc.py trial\_samplingdata.csv trial\_anonymizeddata1.csv test.csv False True

# trial\_samplingdata.csv は入力データ名(サンプリングデータ(C))

# trial\_anonymizeddata1.csv は入力データ名(匿名化データ(D))

# test.csv は有用性評価用の機械学習で用いるテストデータ名

# Falseはヘッダの有無. 今回は無いのでFalse

# Trueはカンマ区切りの後の空白を除去するかどうか. Trueを指定

--> 4つの実数が標準出力される．左から順にヒストグラム，分散共分散行列，決定木分析(目的変数:income)の有用性評価値, 決定木分析(目的変数:relationship(Husbandか否か))

\* 攻撃(メンバシップ推定)の仕方

$ python attack.py trial\_syntheticdata.csv trial\_anonymizeddata1.csv trial\_inference.index False True

# trial\_syntheticdata.csv は入力データ名(擬似データ(B))

# trial\_anonymizeddata1.csv は入力データ名(匿名化データ(D))

# trial\_inference.index は出力データ名(サンプリングデータの推定行(E))

# Falseはヘッダの有無. 今回は無いのでFalse

# Trueはカンマ区切りの後の空白を除去するかどうか. Trueを指定

\* 安全性評価の仕方

$ python privacymeasure.py trial\_answer.index trial\_inference.index

# trial\_answer.index は入力データ名(サンプリングデータの正解行(C'))

# trial\_inference.index は入力データ名(サンプリングデータの推定行(E))

--> 評価値が標準出力される．