Package 'strategize'

April 11, 2023

Title Optimal Stochastic Interventions with High-dimensional Data

Version 0.0

Authors Connor Jerzak <connor.jerzak@austin.utexas.edu> [aut, cre]', 'Kosuke Imai <imai@harvard.edu> [aut]

Description
Software for performing optimal stochastic intervention analysis with high-dimensional data.

Depends R (>= 3.3.3)

License
Creative Commons Attribution-Noncommercial-No Derivative Works 4.0, for academic use only.

Encoding UTF-8

LazyData true

Maintainer 'Connor Jerzak' <connor.jerzak@gmail.com>

RoxygenNote 7.2.1

R topics documented:

computeQ_lda *computeQ_lda*

Description

Implements ...

Usage

```
computeQ_lda(
  hypotheticalTopicProportion = NULL,
  n_fold = 3,
  Yobs,
  topicProportions,
  documents_list,
  wordTopicDistributions,
  se_ub = sd(Yobs)/10,
  split1_indices = NULL,
  split2_indices = NULL,
  computePiSEs = T,
  findMax = T,
  nboot = 10,
  trim_q = 1,
  maxWt = 1e+10,
  maxWt_hajek = NULL,
  alphaLevel = 0.05,
  openBrowser = F
)
```

Arguments

dfm	'document-feature matrix'. A list ...

Value

A list consiting of

- Items.

References

- Kosuke Imai, Rohit, Connor

Examples

```
#set seed
set.seed(1)

#Geneate data
x <- rnorm(100)
```

generate_ExactSol *generate_ExactSol*

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
generate_ExactSol(x, y, by ...)
```


```
generate_ExactSol(x, y, by ...)
```


Arguments

x, y	data frames to be merged

Details

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)
```

```
print( linkedOrgs )


#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

`generate_GD_WithExactGradients` *generate_ExactSol*

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
generate_ExactSol(x, y, by ...)
```

Arguments

`x, y`	data frames to be merged

Details

`LinkOrgs` automatically processes the name text for each dataset (specified by `by`, `by.x`, and/or `by.y`. Users may specify the following options:

- Set `DistanceMeasure` to control algorithm for computing pairwise string distances. Options include "`osa`", "`jaccard`", "`jw`". See `?stringdist::stringdist` for all options. (Default is "`jaccard`")

Value

`z` The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

`generate_ManualDoUpdates`
generate_ManualDoUpdates

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
generate_ManualDoUpdates(x, y, by ...)
```

Arguments

`x, y`	data frames to be merged

Details

`LinkOrgs` automatically processes the name text for each dataset (specified by `by`, `by.x`, and/or `by.y`. Users may specify the following options:

- Set `DistanceMeasure` to control algorithm for computing pairwise string distances. Options include "`osa`", "`jaccard`", "`jw`". See `?stringdist::stringdist` for all options. (Default is "`jaccard`")

Value

`z` The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)
```

```
# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

`generate_ModelOutcome` *generate_ModelOutcome*

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
generate_ModelOutcome(x, Y, by ...)
```

Arguments

`x, Y`	data frames to be merged

Details

`LinkOrgs` automatically processes the name text for each dataset (specified by `by`, `by.x`, and/or `by.Y`. Users may specify the following options:

- Set `DistanceMeasure` to control algorithm for computing pairwise string distances. Options include "`osa`", "`jaccard`", "`jw`". See `?stringdist::stringdist` for all options. (Default is "`jaccard`")

Value

`z` The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
Y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       Y = Y,
                       by.x = "orgnames_x",
                       by.Y = "orgnames_y",
                       MaxDist = 0.6)
```

```
print( linkedOrgs )
```

`generate_ModelOutcome_FindIt`
generate_ModelOutcome_FindIt

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
generate_ModelOutcome_FindIt(x, y, by ...)
```

Arguments

`x, y`	data frames to be merged

Details

`LinkOrgs` automatically processes the name text for each dataset (specified by `by`, `by.x`, and/or `by.y`. Users may specify the following options:

- Set `DistanceMeasure` to control algorithm for computing pairwise string distances. Options include "`osa`", "`jaccard`", "`jw`". See `?stringdist::stringdist` for all options. (Default is "`jaccard`")

Value

`z` The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

get_se *get_se*

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
get_se(x, y, by ...)
```

Arguments

x, y	data frames to be merged

Details

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

initialize_m *initialize_m*

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
initialize_m(x, y, by ...)
```

Arguments

x, y	data frames to be merged

Details

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

ml_build *ml_build*

Description

Implements...

Usage

```
ml_build(x, y, by ...)
```

Arguments

x, y	data frames to be merged

Details

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

ml_train *ml_train*

Description

Implements the organizational record linkage algorithms of Jerzak and Libgober (2021).

Usage

```
ml_train(x, y, by ...)
```

Arguments

x, y	data frames to be merged

Details

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

strategize_OneStep *computeQ_OneStep*

Description

Implements...

Usage

```
computeQ_OneStep(x, y, by ...)
```

Arguments

x, y	data frames to be merged

Details

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

strategize_TwoStep *computeQ_TwoStep*

Description

Implements...

Usage

```
computeQ_TwoStep(x, Y, by ...)
```

Arguments

x, Y	data frames to be merged

Details

LinkOrgs automatically processes the name text for each dataset (specified by by, by.x, and/or by.Y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
Y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       Y = Y,
                       by.x = "orgnames_x",
                       by.Y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

Index