

# Package ‘optimalcausalities’

February 9, 2023

**Title** Optimal Stochastic Interventions in High-dimensional Data

**Version** 2.0

**Authors**

'Kosuke Imai <imai@harvard.edu> [aut], Rohit <rohitchaudhuri@g.harvard.edu> [aut, cre], Connor Jerzak <cjerzak@g.harvard.edu> [aut, cre]'

**Description** Description here.

**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 . . . . .	2
computeQ_OneStep . . . . .	3
computeQ_TwoStep . . . . .	4
generate_ExactSol . . . . .	5
generate_GD_WithExactGradients . . . . .	6
generate_ManualDoUpdates . . . . .	7
generate_ModelOutcome . . . . .	8
generate_ModelOutcome_FindIt . . . . .	9
get_se . . . . .	10
initialize_m . . . . .	11
ml_build . . . . .	12
ml_train . . . . .	13
<b>Index</b>	<b>14</b>

---

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	<i>generate_ExactSol</i>
-------------------	--------------------------

---

## 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_orenames <- c("apple","oracle","enron inc.,"mcdonalds corporation")
y_orenames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orenames)
y <- data.frame("orgnames_y"=y_orenames)

# 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_orenames <- c("apple","oracle","enron inc.,"mcdonalds corporation")
y_orenames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orenames_x"=x_orenames)
y <- data.frame("orenames_y"=y_orenames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orenames_x",
                       by.y = "orenames_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_orenames <- c("apple","oracle","enron inc.,"mcdonalds corporation")
y_orenames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orenames_x"=x_orenames)
y <- data.frame("orenames_y"=y_orenames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orenames_x",
                       by.y = "orenames_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_orenames <- c("apple","oracle","enron inc.,"mcdonalds corporation")
y_orenames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orenames_x"=x_orenames)
y <- data.frame("orenames_y"=y_orenames)
```

```
# 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_orenames <- c("apple", "oracle", "enron inc.", "mcdonalds corporation")
y_orenames <- c("apple corp", "oracle inc", "enron", "mcdonalds co")
x <- data.frame("orgnames_x"=x_orenames)
y <- data.frame("orgnames_y"=y_orenames)

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









ml_build	<i>ml_build</i>
----------	-----------------



# Index

`computeQ_lda`, [2](#)  
`computeQ_OneStep`, [3](#)  
`computeQ_TwoStep`, [4](#)  
  
`generate_ExactSol`, [5](#)  
`generate_GD_WithExactGradients`, [6](#)  
`generate_ManualDoUpdates`, [7](#)  
`generate_ModelOutcome`, [8](#)  
`generate_ModelOutcome_FindIt`, [9](#)  
`get_se`, [10](#)  
  
`initialize_m`, [11](#)  
  
`ml_build`, [12](#)  
`ml_train`, [13](#)