Package: klsh (via r-universe)

November 3, 2026

Type Package

Title Blocking for Record Linkage

Version 0.1.0

Depends R (>= 3.0.2), blink, stats, utils, plyr

Imports Rcpp, stringi, SnowballC

Suggests knitr, ggplot2, rmarkdown

VignetteBuilder knitr

Description An implementation of the blocking algorithm KLSH in Steorts, Ventura, Sadinle, Fienberg (2014) <DOI:10.1007/978-3-319-11257-2_20>, which is a k-means variant of locality sensitive hashing. The method is illustrated with examples and a vignette.

Encoding UTF-8

LazyData true

License GPL-3

RoxygenNote 7.1.1.9000

NeedsCompilation no

Author Rebecca Steorts [aut, cre]

Maintainer Rebecca Steorts <beka@stat.duke.edu>

Config/pak/sysreqs libicu-dev

Repository https://cranhaven.r-universe.dev

Date/Publication 2026-11-03 06:02:00 UTC

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/klsh

RemoteSha 1599ccdc715e433673b30daa846157a4d139f793

RemoteSubdir klsh

Contents

bag_of_word_ify . . . 2
bag_signatures . . . 3
block.ids.from.blocking . . . 3
calc_idf . . . 4
confusion.from.blocking . . . 5
klsh . . . 5
reduction.ratio . . . 6
reduction.ratio.from.blocking . . . 7
rproject_bags . . . 7
sacks_of_bags_of_words . . . 8
tokenify . . . 9

Index **10**

`bag_of_word_ify` *Function to convert a record into a bag of tokens with a fieldwise flag*

Description

Function to convert a record into a bag of tokens with a fieldwise flag

Usage

```
bag_of_word_ify(record, k, fieldwise = FALSE)
```

Arguments

`record`	String or record
`k`	Parameter k, which is the number of shingle, tokens, or grams to break the string into
`fieldwise`	Flag where the defalt setting to include the record as the entire string

Value

Computes the bag of tokens for a string

Examples

```
data(RLdata500)
data.500 <- RLdata500[-c(2,4)]
bag_of_word_ify(data.500[1,c(-2)],k=2)
bag_of_word_ify(data.500[300,c(-2)],k=2)
names(bag_of_word_ify(data.500[300,c(-2)],k=2))
```

bag_signatures *Function that reduces a bag of words into a signature matrix using multiple random projections*

Description

Function that reduces a bag of words into a signature matrix using multiple random projections

Usage

```
bag_signatures(sack_of_bags, p, weighting_table)
```

Arguments

sack_of_bags	Sack of bag of words
p	Number of random projections p
weighting_table	Weighting table (inverse document frequency)

Value

Computes a signature matrix using multiple random projections and the inverse document frequency weights

Examples

```
data(RLdata500)
data.500 <- RLdata500[-c(2,4)]
sack <- sacks_of_bags_of_words(data.500[1:3,c(-2)],k=2)
idf <- calc_idf(sack)
bag_signatures(sack, p=5, idf)
```

block.ids.from.blocking *Returns the block ids associated with a blocking method.*

Description

Returns the block ids associated with a blocking method.

Usage

```
block.ids.from.blocking(blocking)
```

Arguments

blocking	A list of the blocks.

Value

A list of the blocks ids that corresponds to each block

Examples

```
data("RLdata500")
klsh.blocks <- klsh(RLdata500, p=20, num.blocks=5, k=2)
block.ids.from.blocking(klsh.blocks)
```

---

calc_idf *Function to calculate the inverse document frequency given a shingled bag of words*

Description

Function to calculate the inverse document frequency given a shingled bag of words

Usage

```
calc_idf(sack_of_bags)
```

Arguments

sack_of_bags	Sack of bag of words

Value

Computes the inverse document frequency for a bag of words

Examples

```
data(RLdata500)
data.500 <- RLdata500[-c(2,4)]
sack <- sacks_of_bags_of_words(data.500[1:3,c(-2)],k=2)
(idf <- calc_idf(sack))
match(names(sack[[1]]), names(idf))
```

confusion.from.blocking

Perform evaluations (recall) for blocking.

Description

Perform evaluations (recall) for blocking.

Usage

```
confusion.from.blocking(blocking, true_ids, recall.only = FALSE)
```

Arguments

blocking	A list of the blocks
true_ids	The true identifiers for comparisons
recall.only	Flag that when true only prints the recall, otherwise prints many evaluation metrics in a list

Value

A vector of that returns the recall and the precision

Examples

```
data("RLdata500")
klsh.blocks <- klsh(RLdata500, p=20, num.blocks=5, k=2)
confusion.from.blocking(klsh.blocks, identity.RLdata500)
confusion.from.blocking(klsh.blocks, identity.RLdata500, recall.only=TRUE)
```

klsh

Function that reduces a bag of words into a signature matrix using multiple random projections

Description

Function that reduces a bag of words into a signature matrix using multiple random projections

Usage

```
klsh(r.set, p, num.blocks, k, fieldwise = FALSE, quiet = TRUE)
```

Arguments

r.set	Set of records
p	Number of random projections p
num.blocks	The total number of desired blocks
k	The total number of tokens
fieldwise	Flag with default FALSE
quiet	Flag to turn on printed progress, default to TRUE

Value

The blocks from performing KLSH

Examples

```
data(RLdata500)
data.500 <- RLdata500[-c(2,4)]
klsh.blocks <- klsh(data.500, p=20, num.blocks=5, k=2)
```

---

reduction.ratio *Returns the reduction ratio associated with a blocking method*

Description

Returns the reduction ratio associated with a blocking method

Usage

```
reduction.ratio(block.labels)
```

Arguments

block.labels	A list of the blocks labels.

Value

The reduction ratio

Examples

```
data("RLdata500")
klsh.blocks <- klsh(RLdata500, p=20, num.blocks=5, k=2)
block.ids <- block.ids.from.blocking(klsh.blocks)
reduction.ratio(block.ids)
```

reduction.ratio.from.blocking

Returns the reduction ratio associated with a blocking method

Description

Returns the reduction ratio associated with a blocking method

Usage

```
reduction.ratio.from.blocking(blocking)
```

Arguments

blocking	The actual blocks

Value

The reduction ratio

Examples

```
data("RLdata500")
klsh.blocks <- klsh(RLdata500, p=20, num.blocks=5, k=2)
reduction.ratio.from.blocking(klsh.blocks)
```

rproject_bags

Function that generates unit random vectors and takes (weighted) projections onto the random unit vectors given a bag of words

Description

Function that generates unit random vectors and takes (weighted) projections onto the random unit vectors given a bag of words

Usage

```
rproject_bags(sack_of_bags, weighting_table)
```

Arguments

sack_of_bags	Sack of bag of words
weighting_table	Weighting table (inverse document frequency)

Value

Computes the inverse document frequency for a bag of words

Examples

```
data(RLdata500)
data.500 <- RLdata500[-c(2,4)]
sack <- sacks_of_bags_of_words(data.500[1:3,c(-2)],k=2)
idf <- calc_idf(sack)
match(names(sack[[1]]), names(idf))
rproject_bags(sack, idf)
```

---

sacks_of_bags_of_words

Function to convert all records into a bag of tokens

Description

Function to convert all records into a bag of tokens

Usage

```
sacks_of_bags_of_words(r.set, k, fieldwise = FALSE)
```

Arguments

r.set	Record set
k	Parameter k, which is the number of shingle, tokens, or grams to break the string into
fieldwise	Flag where the defalt setting to include the record as the entire string

Value

Computes the bag of tokens for a record set

Examples

```
data(RLdata500)
data.500 <- RLdata500[-c(2,4)]
sacks_of_bags_of_words(data.500[1:3,c(-2)],k=2)
```

tokenify *Function to token a string into its k components*

Description

Function to token a string into its k components

Usage

```
tokenify(string, k)
```

Arguments

string	A string or record
k	A parameter k, which is the number of shingle, tokens, or grams to break the string into

Value

Computes the tokenized or grammed version of a string

Examples

```
tokenify("Alexander",2)
tokenify("Alexander Smith", 2)
```

Index

bag_of_word_ify, 2
bag_signatures, 3
block.ids.from.blocking, 3

calc_idf, 4
confusion.from.blocking, 5

klsh, 5

reduction.ratio, 6
reduction.ratio.from.blocking, 7
rproject_bags, 7

sacks_of_bags_of_words, 8

tokenify, 9