Package ‘TopicScore’

October 12, 2022

Title The Topic SCORE Algorithm to Fit Topic Models

Version 0.0.1

Description Provides implementation of the ``Topic SCORE'' algorithm that is proposed by Tracy Ke and Minzhe Wang. The singular value decomposition step is optimized through the usage of svds() function in 'RSpectra' package, on a 'dgRMatrix' sparse matrix. Also provides a column-wise error measure in the word-topic matrix A, and an algorithm for recovering the topic-document matrix W given A and D based on quadratic programming.

The details about the techniques are explained in the paper ``A new SVD approach to optimal topic estimation'' by Tracy Ke and Minzhe Wang (2017) <arXiv:1704.07016>.

Depends R (>= 3.5.0)

License MIT + file LICENSE

LazyData true

RoxygenNote 6.1.1

Imports utils, stats, graphics, RSpectra, combinat, quadprog, methods, Matrix, slam

Author Minzhe Wang [aut, cre], Tracy Ke [aut]

Maintainer Minzhe Wang <minzhew@uchicago.edu>

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-06 11:12:14 UTC

R topics documented:

AP *Associated Press data*

Description

Associated Press data from the First Text Retrieval Conference (TREC-1) 1992, which has being processed by stop-words removal, low-frequency words removal and short documents removal.

Usage

```
data("AP")
```

Format

The data set is an object of class `"simple_triplet_matrix"` provided by package **slam**. It is a word-document matrix which contains the term frequency of 7000 words in 2134 documents.

Source

Harman, D. (1992, November). Overview of the First Text REtrieval Conference (TREC-1). In TREC (Vol. 1992, pp. 1-20).

error_A *The l_1 distance between two thin matrices up to a column permuation*

Description

This function computes l_1 distance between two thin matrices up to a column permuation, that is to find the smallest sum of absolute value entry-wise difference between two matrices over all possible permutations over the columns of the first matrix. This can be done either universially or greedily.

Usage

```
error_A(A, A_hat, type = "u")
```

Arguments

`A` The first p-by-K matrix.

`A_hat` The second p-by-K matrix.

`type` The search type for the best permutation. If it's 'u', the search is done universially, that is over all possible permuations of the columns of A. If it's 'g', the search is done greedily, that is at kth step find the closest column in the remaining columns of A to the kth column of A_hat in terms of l_1 distance. Greedy search may result in sub-optimal solutions, but it can be computed much faster than the universal way when K is large. The default value is 'u'.

Value

The l_1 distance.

Author(s)

Minzhe Wang

Examples

```
# The example uses the runif() function in the 'stats' package
A <- matrix(runif(10*3),10,3)
A_hat <- A + 0.1*matrix(runif(10*3),10,3)
error_A(A, A_hat)
error_A(A, A_hat, type='g')
```

simplex_dist *The l_2 distance between a point and a simplex*

Description

This function computes the l_2 distance between a point and a simplex, that is the shortest l_2 distance between the given point and any point in the simplex.

Usage

```
simplex_dist(theta, V)
```

Arguments

`theta`	A (K-1) dimensional vector, representing a point.
`V`	The K-by-(K-1) vertices matrix, with each row being a vertex.

Value

The l_2 distance.

Author(s)

Minzhe Wang

References

Ke, Z. T., & Wang, M. (2017). A new SVD approach to optimal topic estimation. arXiv preprint arXiv:1704.07016.

Examples

```
# Generate 3 vertices
V <- rbind(c(-1/2,-1/2), c(1,0), c(0,1))

theta <- c(3,1)
simplex_dist(theta, V)
```

`topic_score` *The Topic SCORE algorithm*

Description

This function obtains the word-topic matrix A from the word-document matrix X through the Topic SCORE algorithm.

Usage

```
topic_score(K, X, K0, m, Mquantile = 0, scatterplot = FALSE,
  num_restart = 1, seed = NULL)
```

Arguments

`K` The number of topics.

`X` The p-by-n word-document matrix, with each column being a distribution over a fixed set of vocabulary. This matrix can be of class `simple_triplet_matrix` defined in **slam** package, or any other class that can be transformed to class `dgRMatrix` defined in **Matrix** package through `as` function in **methods** package.

`K0` The number of greedy search steps in vertex hunting. If the value is missing it will be set to ceiling(1.5*K).

`m` The number of centers in the kmeans step in vertex hunting. If the value is missing it will be set to 10*K.

`Mquantile` The percentage of the quantile of the diagonal entries of matrix M, which is used to upper truncate the diagonal entries of matirx M. When it's 0, it will degenerate the case when there is no normalization. When it's 1, it means there is no truncation. Default is 0.

`scatterplot` Whether a scatterplot of rows of R will be generated.

`num_restart` The number of random restart in the kmeans step in vertex hunting. Default is 1.

`seed` The random seed. Default value is NULL.

Value

A list containing

A_hat The estimated p-by-K word-topic matrix.

R The p-by-(K-1) left singular vector ratios matrix.

V The K-by-(K-1) vertices matrix, with each row being a vertex found through the vertex hunting algorithm in the simplex formed by the rows of R.

Pi The p-by-K convex combinations matrix, with each row being the convex combination coefficients of a row of R using V as vertices.

theta The K0-by-(K-1) matrix of K0 potential vertices found in the greedy step of the vertex hunting algorithm.

Author(s)

Minzhe Wang

References

Ke, Z. T., & Wang, M. (2017). A new SVD approach to optimal topic estimation. arXiv preprint arXiv:1704.07016.

Examples

```
data("AP")
K <- 3
tscore_obj <- topic_score(K, AP)

# Visualize the result
plot(tscore_obj$R[,1], tscore_obj$R[,2])
```

`vertices_est` *The vertex hunting in the Topic SCORE algorithm*

Description

This function conducts the vertex hunting in the Topic SCORE algorithm. More generally this function finds a simplex with K vertices that best approximates the given p data points in a (K-1) dimensional space.

Usage

```
vertices_est(R, K0, m, num_restart)
```

Arguments

R The p-by-(K-1) data matrix, with each row being a data point.

K0 The number of greedy search steps.

m The number of centers in the kmeans step.

num_restart The number of random start in the kmeans step.

Value

A list containing

V The K-by-(K-1) vertices matrix, with each row being a vertex in the found simplex.

theta The K0-by-(K-1) matrix of potential K0 vertices found in the greedy step.

Author(s)

Minzhe Wang

References

Ke, Z. T., & Wang, M. (2017). A new SVD approach to optimal topic estimation. arXiv preprint arXiv:1704.07016.

Examples

```
# Generate 3 vertices
V <- rbind(c(-1/2,-1/2), c(1,0), c(0,1))

# Randomly generate the convex combination weights of 1000 points
Pi <- matrix(runif(3*1000),3,1000)
Pi <- apply(Pi, 2, function(x){x/sum(x)})

R <- t(Pi)%*%V
v_est_obj <- vertices_est(R, 1.5*3, 10*3, 1)

# Visualize the result
plot(R[,1], R[,2])
points(v_est_obj$V[,1], v_est_obj$V[,2], col=2, lwd=5)
```

W_from_AD *Estimation of W from A and X*

Description

This function estimates the topic-document matrix W from the word-topic matrix A and the word-document X through quadratic programming.

Usage

```
W_from_AD(A, X)
```

Arguments

A	The p-by-K word-topic matrix.
X	The p-by-n word-document matrix.

Value

The estimated K-by-n topic-document matrix W_hat.

Author(s)

Minzhe Wang

Examples

```
data("AP")
K <- 3
tscore_obj <- topic_score(K, AP)
W_hat <- W_from_AD(tscore_obj$A_hat, AP)
```

Index