Package ‘chickn’

November 24, 2020

Type Package
Title 'Compressive' Hierarchical Kernel Clustering Toolbox
Version 1.2.3
Date 2020-11-08
Maintainer Olga Permiakova <olga.permiakova@gmail.com>

Description
Routines for efficient cluster analysis of large scale data. This package implements the 'CHICKN' clustering algorithm (see 'Permiakova' et al. (2020) "'CHICKN': Extraction of 'peptide' 'chromatographic' 'elution' profiles from large scale mass 'spectrometry' data by means of 'Wasserstein' 'compressive' hierarchical cluster analysis"). Functions for data compression, hierarchical clustering and post processing are provided.

License GPL (>= 2)
Encoding UTF-8
Depends R (>= 3.5)
Imports Rcpp, bigstatsr (>= 1.2.3), RcppParallel, mvnfast, zipfR, MASS, pracma, nloptr, foreach, doRNG, parallel, doParallel, Rdpack
LinkingTo Rcpp, RcppArmadillo, RcppParallel, bigstatsr (>= 1.2.3), rmio

SystemRequirements C++11
RoxygenNote 7.1.1
RdMacros Rdpack

NeedsCompilation yes
Author Olga Permiakova [aut, cre], Romain Guibert [aut], Thomas Burger [aut]
Repository CRAN
Date/Publication 2020-11-24 14:00:02 UTC
R topics documented:

chickn .................................................. 2
CHICKN_W1 ............................................... 3
COMPR ................................................... 5
cumsum_parallel ................................. 7
DrawFreq ............................................... 7
EstimSigma ........................................... 8
E_parallel ............................................. 10
gamma_estimation ................................... 10
GenerateFrequencies .............................. 11
hcc_parallel ......................................... 12
Nystrom_kernel ..................................... 13
Preimage ............................................... 15
Sketch .................................................. 16
UPS2 .................................................... 17
W1_parallel ......................................... 17

Index 19

chickn  chicken-package

Description

The R package chickn implements the Chromatogram Hierarchical Compressive K-means with Nystrom approximation clustering approach. It is designed to cluster a large collection of high-resolution mass spectrometry signals (chromatographic profiles) relying on a compressed version of the data (a.k.a. data sketch). Data compression is achieved following the guidelines for Nystrom approximation provided by (Wang et al. 2019) and the sketching operator from (Keriven et al. 2018). The Filebacked Big Matrix (FBM) class from the bigstatsr package is used to store and to manipulate matrices, which cannot be load in memory.

Author(s)

Olga Permiakova, Romain Guibert, Thomas Burger

References

CHICKN_W1

Chromatogram Hierarchical Compressive K-means with Nystrom approximation

Description
An implementation of the complete pipeline of the CHICKN algorithm.

Usage

CHICKN_W1(
  Data,
  K = 2,
  k_total,
  K_W1 = NULL,
  kernel_type = "Gaussian",
  distance_type = "W1",
  Freq = NULL,
  ncores = 2,
  max_neighbors = 32,
  nblocks = 64,
  N0 = 10000,
  max_Nsize = 32,
  DoPreimage = FALSE,
  DIR_output = tempfile(),
  DIR_tmp = tempfile(),
  BIG = FALSE,
  verbose = FALSE,
  ...
)

Arguments

Data
A Filebacked Big Matrix n x N.

K
Number of cluster at each call of clustering method. Default is 2.

k_total
An upper bound of the total number of clusters.

K_W1
A Filebacked Big Matrix. Nystrom kernel matrix s x N, where N is the number
of signals in the training collection and s is the Nystrom sample size. By default
is NULL and it is generated using Nystrom_kernel function.

kernel_type
Kernel function type c('Gaussian', 'Laplacian').

distance_type
Distance function type. The available types are Wasserstein-1 ('W1') and Euclidean ('Euclide'). The default value is 'W1'.

Freq
A frequency matrix m x n with frequency vectors in rows. If NULL, the frequency vectors are generated by GenerateFrequencies function.

ncores
Number of cores. Default is 2.
max_neighbors Number of neighbors used to estimate the kernel parameter gamma. Default is 32.

nblocks Number of blocks, on which the regression is performed. Default is 32.

N0 Number of data vectors used for the variance estimation in EstimSigma.

max_Nsize Number of neighbors used to compute consensus chromatograms.

DoPreimage logical that controls whether to compute the consensus chromatograms. Default is TRUE.

DIR_output A directory to save the results.

DIR_tmp A directory for temporal files.

BIG logical parameter that controls whether the resulting consensus chromatograms are stored as a Filebacked Big Matrix (‘Centroid_preimage.bk’). Default is FALSE.

verbose logical that indicates whether display the processing steps.

... Additional arguments passed on to COMPR.

Details

CHICKN_W1 compresses the data by computing a Nystrom kernel approximation and applying the sketching operator from (Keriven et al. 2018). See Nystrom_kernel and Sketch functions. Then clusters are recovered by operating on the compressed data version. It can use the kernel function based on the Wasserstein-1 or the Euclidean distances. It generates in DIR_output directory the following files:

• ‘Cluster_assign_out.bk’ is a Filebacked Big Matrix N x maxLevel+1, which stores the cluster assignment at each hierarchical level.

• ‘Centroids_out.bk’ is a Filebacked Big Matrix with the resulting cluster centroids in columns.

Value

A list with the following attributes:

• gamma is the estimated kernel parameter.

• CompressedData is the Nystrom kernel matrix.

• sigma is the estimated variance.

• Frequency is the frequency matrix m x n.

• Clusters is the cluster assignment.

References


See Also

Nystrom_kernel, GenerateFrequencies, hcc_parallel, Preimage, bigstatsr
Examples

```r
data("UPS2")
N = ncol(UPS2)
n= nrow(UPS2)
X_FBM = bigstatsr::FBM(init = UPS2, ncol=N, nrow = n)$save()
output <- CHICKN_W1(Data = X_FBM, K = 2, k_total =8, max_neighbors = 10, ncores = 2,
                      N0 = N, DoPreimage = FALSE)
```

---

**Description**

An implementation of the Compressive Orthogonal Matching Pursuit with Replacement algorithm

**Usage**

```r
COMPR(
  Data,
  ind.col = 1:ncol(Data),
  K,
  Frequencies,
  lower_b,
  upper_b,
  SK_Data,
  maxIter = 300,
  HardThreshold = TRUE,
  options = list(tol_centroid = 1e-08, nIterCentroid = 1500, min_weight = 0, max_weight = Inf, nIterLast = 1000, tol_global = 1e-12)
)
```

**Arguments**

- **Data**: A Filebacked Big Matrix n x N, data vectors are stored in the matrix columns.
- **ind.col**: Column indeces, which indicate which data vectors are considered for clustering. By default the entire Data matrix.
- **K**: Number of clusters.
- **Frequencies**: A frequency matrix m x n with frequency vectors in rows.
- **lower_b**: A vector of the lower boundary of data.
- **upper_b**: A vector of the upper boundary.
- **SK_Data**: Data sketch vector of the length 2m. It can be computed using `Sketch`.
- **maxIter**: Maximum number of iterations in the global optimization with respect to cluster centroid vectors and their weights. Default is 300.
HardThreshold logical that indicates whether to perform the replacement. Default is TRUE.

options List of optimization parameters:

• **tol_centroid** is a tolerance value for the centroid optimization. Default is 1e-8.
• **nIterCentroid** is a maximum number of iterations in the centroid optimization (default is 1500).
• **min_weight** is a lower bound for centroid weights (default is 0).
• **max_weight** is an upper bound for centroids weights (default is Inf)
• **nIterLast** is a number of iteration in the global optimization at the last algorithm iteration. Default is 1000.
• **tol_global** is a tolerance value for the global optimization. Default is 1e-12.

Details

COMPR is an iterative greedy method, which alternates between expanding the cluster centroid set \( C \) with a new element \( c_i \), whose sketch is the most correlated to the residue and the global minimization with respect to cluster centroids \( c_1, \ldots, c_K \) and their weights \( w_1, \ldots, w_K \). It clusters the data collection into \( K \) groups by minimizing the difference between the compressed data version (data sketch) and a linear combination of cluster centroid sketches, i.e. \( \|Sk(Data) - \sum_{i=1}^{K} w_i \cdot Sk(c_i)\| \).

Value

A matrix \( n \times K \) with cluster centroid vectors in columns.

Note

This method is also referred to as Compressive K-means and it has been published in Keriven N, Tremblay N, Traonmilin Y, Gribonval R (2017). “Compressive K-means.” In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6369–6373. IEEE.

Examples

```r
X = matrix(rnorm(1e5), ncol=1000, nrow = 100)
lb = apply(X, 1, min)
ub = apply(X, 1, max)
X_FBM = bigstatsr::FBM(init = X, ncol=1000, nrow = 100)
out = GenerateFrequencies(Data = X_FBM, m = 20, N0 = ncol(X_FBM))
SK = Sketch(Data = X_FBM, W = out$W)
C <- COMPR(Data = X_FBM, K = 2, Frequencies = out$W, lower_b = lb, upper_b = ub, SK_Data = SK)
```
**cumsum_parallel**  
*Cumulative sum computation*

**Description**

Parallel implementation of the cumulative sum of the matrix columns.

**Usage**

```r
cumsum_parallel(X, A_cumsum)
```

**Arguments**

- **X**: A Filebacked Big Matrix n x N.
- **A_cumsum**: A Filebacked Big Matrix n x N, where cumulative sums are stored.

---

**DrawFreq**  
*Draw frequency vectors*

**Description**

Function samples frequency vectors from the selected frequency distribution law.

**Usage**

```r
DrawFreq(
  m,
  n,
  sigma,
  alpha = rep(1, length(sigma)),
  TypeDist = "AR",
  ncores = 1,
  parallel = FALSE
)
```

**Arguments**

- **m**: Number of frequency vectors.
- **n**: Length of frequency vector.
- **sigma**: Data variance, a scalar or a vector in the case of the Gaussian distribution mixture.
- **alpha**: Variance weights. By default all are equal to 1.
- **TypeDist**: Frequency distribution type. Possible values: "G" (Gaussian), "FG" (Folded Gaussian radial) or "AR" (Adapted radius). Default is "AR".
ncores  Number of cores. Multicore computation should be used only when the data is a mixture of Gaussian distributions.

parallel  logical parameter that defines whether to perform the parallel computations. Default is FALSE.

Details

The frequency vectors $w_1, \ldots, w_m$ are randomly sampled from the predefined frequency distribution. The distribution law can be either $N(0, \Sigma^{-1})$ (typeDist = "G") or $p_R \cdot \varphi \cdot \Sigma^{-\frac{1}{2}}$ (typeDist = c("FG", "AR")), where $\varphi$ is a vector uniformly distributed on the unit sphere, $\Sigma$ is a diagonal matrix with the data variance $\sigma$ on the diagonal and where $p_R$ is the radius density function. For "FG" the radius distribution is $N(0, 1)$ and for "AR" $p_R = C \cdot (R^2 + \frac{R^4}{4})^{0.5} \cdot \exp(-0.5 \cdot R^2)$, where $C$ is a normalization constant.

Value

A matrix $m \times n$, with frequency vectors in rows.

Note


See Also

EstimSigma, GenerateFrequencies, Sketch

Examples

```r
W1 = DrawFreq(m = 20, n = 10, sigma = 1e-3, TypeDist = "AR")
W2 = DrawFreq(m = 20, n = 10, sigma = 1e-3, TypeDist = "FG")
W3 = DrawFreq(m = 20, n = 10, sigma = 1e-3, TypeDist = "G")
```

---

<table>
<thead>
<tr>
<th>EstimSigma</th>
<th>Data variance estimation</th>
</tr>
</thead>
</table>

Description

The mean data variance estimation.
EstimSigma

Usage

EstimSigma(
  Data,
  ind.col,
  m,
  nbblocks = 32,
  niter = 3,
  sigma_start = 0.1,
  nparts = 1,
  ...
)

Arguments

Data      A Filebacked Big Matrix n x N. Data signals are stored in the matrix columns.
ind.col   Column indeces for which the data sketch is computed. By default all matrix columns.
m         Number of frequency vectors.
nblocks   Number of blocks, on which the regression is performed. Default is 32.
niter     Number of iterations. Default is 3.
sigma_start An initial value of the data variance. Default is 0.1.
nparts    Number of parts to split the data for the data sketch computation.
...       Additional arguments passed on to DrawFreq function.

Value

The estimated data variance.

Note


See Also

DrawFreq, Sketch, GenerateFrequencies

Examples

X = matrix(rnorm(1e5), ncol=1000, nrow = 100)
X_FBM = bigstatsr::FBM(init = X, ncol=1000, nrow = 100)
sigma = EstimSigma(Data = X_FBM, ind.col = seq(1,1000, by = 2), m = 20, nbblocks = 4)
**E_parallel**  
*Euclidean distance*

**Description**
Euclidean distance

**Usage**

\[ E_{\text{parallel}}(X, C, \text{set}_c) \]

**Arguments**

- **X**: A Filebacked Big Matrix \( n \times N \).
- **C**: A Filebacked Big Matrix \( N \times l \), which stores the Euclidean distances.
- **set_c**: Column index vector. The data vector indices for which the Euclidean distances are computed.

**Details**
The Euclidean distances are computed between the data vectors from \( \text{set}_c \) and all columns of \( X \).

**gamma_estimation**  
*Kernel parameter estimation*

**Description**
Kernel parameter estimation by averaging the distances to the closest neighbors.

**Usage**

\[ \text{gamma_estimation}(X, \text{size}, \text{kernel_type}) \]

**Arguments**

- **X**: A Filebacked Big Matrix \( n \times N \).
- **size**: Neighborhood size.
- **kernel_type**: Kernel function type. Available types are \( \text{c("Gaussian", "Laplacian")} \).

**Value**
The estimated kernel parameter.
GenerateFrequencies

Frequency vector construction

Description

Function performs the data variance estimation and the frequency matrix construction.

Usage

GenerateFrequencies(Data, m, N0 = 5000, TypeDist = "AR", verbose = FALSE, ...)

Arguments

- **Data**: A Filebacked Big Matrix n x N with data vectors in columns.
- **m**: Number of frequency vectors.
- **N0**: Number of data vectors used for the variance estimation in `EstimSigma`.
- **TypeDist**: Frequency distribution type. Possible values: "G" (Gaussian), "FG" (Folded Gaussian radial) or "AR" (Adapted radius). Default is "AR".
- **verbose**: logical that indicates whether display the process steps.
- **...**: Additional arguments passed on to `EstimSigma` and `DrawFreq` functions.

Details

The data variance is estimated on the N0 data vectors randomly selected from Data using `EstimSigma` function. The frequency vectors are sampled using `DrawFreq` function.

Value

A list with the following attributes:

- **W**: the frequency matrix with m frequency vectors in rows.
- **sigma**: the estimated data variance.

References


See Also

`DrawFreq, EstimSigma, Sketch`

Examples

```r
X = matrix(rnorm(1000), ncol=100, nrow = 10)
X_FBM = bigstatsr::FBM(init = X, ncol=100, nrow = 10)
W = GenerateFrequencies(Data = X_FBM, m = 20, N0 = 100, TypeDist = "AR")$W
```
Description

Compressed Hierarchical Clustering.

Usage

hcc_parallel(
  Data,
  W,
  K,
  maxLevel,
  ncores = 2,
  DIR_output = tempfile(),
  hybrid = FALSE,
  verbose = FALSE,
  ...
)

Arguments

Data  A Filebacked Big Matrix n x N. Data signals are stored in the matrix columns.
W     A frequency matrix m x n with frequency vectors in rows.
K     Number of clusters at each call of the clustering algorithm.
maxLevel Maximum number of hierarchical levels.
ncores Number of cores. By default 4.
DIR_output An output directory.
hybrid logical parameter. If TRUE K decreases progressively over hierarchical levels as \( \lceil \frac{K}{level} \rceil \). Default is FALSE.
verbose logical that indicates whether display the processing steps.
...   Additional arguments passed on to COMPR.

Details

This function provides a divisive hierarchical implementation of COMPR. Parallel computations are performed using 'FORK' clusters (Linux-like platform) or 'PSOCK' clusters (Windows platform) using the parallel package. This function generates in the DIR_output directory the following files:

- 'Cluster_assign_out.bk' is a Filebacked Big Matrix N x maxLevel+1, which stores the cluster assignment at each hierarchical level.
- 'Centroids_out.bk' is a Filebacked Big Matrix with the resulting cluster centroids in columns.
Nystrom_kernel

Value

The cluster assignment as a list of clusters with corresponding data vector indeces.

References


See Also

COMPR

Examples

data("UPS2")
N = ncol(UPS2)
n = nrow(UPS2)
X_FBM = bigstatsr::FBM(init = UPS2, ncol=N, nrow = n)$save()
K_W1 = Nystrom_kernel(Data = X_FBM, c = 14, l = 7, s = 5,
  max_neighbors = 3, ncores = 1, kernel = 'Gaussian')$K_W1
W = GenerateFrequencies(Data = K_W1, m = 20, N0 = ncol(X_FBM))$W
C = hcc_parallel(Data = K_W1, W = W, K = 2, maxLevel = 4,
  DIR_output = tempfile(), ncores = 2)

Nystrom_kernel          Nystrom kernel approximation

Description

An implementation of the Nystrom kernel approximation method.

Usage

Nystrom_kernel(
  Data,
  c,
  l,
  s,
  gamma = NULL,
  max_neighbors = 32,
  DIR_output = tempfile(),
  DIR_save = tempfile(),
  ncores = 2,
  ncores_svd = 1,
  distance_type = "W1",
)
Nystrom method consists in approximating the kernel matrix $K$ by $CW^{-1}C^T$, with $C \in R^{N \times c}$ obtained from $K$ by randomly selecting only $c$ columns and $W \in R^{c \times c}$ obtained from $C$ by selecting as well $c$ corresponding rows. The kernel function, based on the distance metric, is given as follows: $k(x_i, x_j) = e^{-\gamma \cdot d_p(x_i, x_j)}$, where $p$ is equal to 1 for 'Laplacian' kernel and equal to 2 for 'Gaussian' kernel and where $d(x_i, x_j)$ is the distance between data vectors $x_i$ and $x_j$.

Value

A list with the following attributes:

- $K_{W1}$ is the Filebacked Big Matrix of the Nystrom kernel approximation.
- $\gamma$ is the estimated kernel parameter.
- $\text{RandomSample}$ is the data vector indices, selected for the Nystrom approximation.

Note

**Preimage**

**See Also**
- *W1_parallel*, *gamma_estimation*, *big_randomSVD*, *cumsum_parallel*.

**Examples**

```r
X = matrix(rnorm(2000), ncol=100, nrow = 20)
X_FBM = bigstatsr::FBM(init = X, ncol=100, nrow = 20)
output = Nystrom_kernel(Data = X_FBM, c = 10, l = 7, s = 5,
                        max_neighbors = 3, ncores = 2)
```

**Description**

Consensus chromatogram computation.

**Usage**

```r
Preimage(
  Data,
  K_W1,
  C_out,
  Cl_assign,
  max_Nsize = 32,
  ncores = 4,
  DIR_out = getwd(),
  BIG = FALSE
)
```

**Arguments**

- **Data** A Filebacked Big Matrix n x N. Data signals are stored in the matrix columns.
- **K_W1** A Filebacked Big Matrix of the Nystrom kernel matrix s x N, where N is the number of signal in the training collection and s is the Nystrom sample size.
- **C_out** A Filebacked Big Matrix of cluster centroids.
- **Cl_assign** A Filebacked Big matrix of the cluster assignment.
- **max_Nsize** Maximum number of neighbors.
- **ncores** Number of cores.
- **DIR_out** A directory to save the result, by default it is the working directory.
- **BIG** logical parameter that controls whether the resulting consensus chromatograms are stored as a Filebacked Big Matrix (‘Centroid_preimage.bk’). Default is FALSE.
Value

A matrix or a Filebacked Big Matrix if BIG = TRUE.

Description

The data sketch computation.

Usage

\texttt{Sketch(Data, W, ind.col = 1:ncol(Data), ncores = 1, parallel = FALSE)}

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>A Filebacked Big Matrix n x N. Data signals are stored in the matrix columns.</td>
</tr>
<tr>
<td>W</td>
<td>A frequency matrix m x n. The frequency vectors are stored in the matrix rows.</td>
</tr>
<tr>
<td>ind.col</td>
<td>Column indeces for which the data sketch is computed. By default all matrix columns.</td>
</tr>
<tr>
<td>ncores</td>
<td>Number of used cores. By default 1. If parallel = FALSE, ncores defines a number of data splits on which the sketch is computed separatelly.</td>
</tr>
<tr>
<td>parallel</td>
<td>logical parameter that indicates whether computations are performed on several cores in parallel or not.</td>
</tr>
</tbody>
</table>

Details

The sketch of the given data collection \(x_1, \ldots, x_N\) is a vector of the length 2m. First m components of the data sketch vector correspond to its real part, \(i.e. \frac{1}{N} \sum_{i=1}^{N} \cos(Wx_i)\). Last m components are its imaginary part, \(i.e. \frac{1}{N} \sum_{i=1}^{N} \sin(Wx_i)\).

Value

The data sketch vector.

References


Examples

\begin{verbatim}
X = matrix(rnorm(1000), ncol=100, nrow = 10)
X_FBM = bigstatsr::FBM(init = X, ncol=100, nrow = 10)
W = GenerateFrequencies(Data = X_FBM, m = 20, N0 = 100, TypeDist = "AR")$W
SK1 = Sketch(X_FBM, W)
SK2 = Sketch(X_FBM, W, parallel = TRUE, ncores = 2)
all.equal(SK1, SK2)
\end{verbatim}
UPS2 dataset

Description
Proteomics data acquired within the mass spectrometry analysis of UPS2 sample.

Usage
data(UPS2)

Format
A matrix with 1653 rows and 190 columns.

Details
Only a small part of data was taken from the original dataset described in (Henning et al. 2018). The UPS2 dataset contains 190 chromatographic traces (matrix columns) acquired along the retention time (matrix rows) in the liquid chromatography.

Source
https://github.com/optimusmoose/ups2GT

References

W1_parallel Wasserstein-1 distance

Description
Wasserstein-1 distance

Usage
W1_parallel(X, C, set_c)
Arguments

- **X**: A Filebacked Big Matrix \( n \times N \).
- **C**: A Filebacked Big Matrix \( N \times l \), which stores the Wasserstein distances.
- **set_c**: Column index vector. The data vector indices for which the Wasserstein distances are computed.

Details

The Wasserstein-1 distances are computed between the data vectors from set_c and all columns of X.
Index

* datasets
  UPS2, 17

big_randomSVD, 15

chickn, 2
CHICKN_W1, 3
COMPR, 4, 5, 12, 13
cumsum_parallel, 7, 15

DrawFreq, 7, 9, 11

E_parallel, 10
EstimSigma, 4, 8, 8, 11

gamma_estimation, 10, 14, 15
GenerateFrequencies, 3, 4, 8, 9, 11

hcc_parallel, 4, 12

Nystrom_kernel, 3, 4, 13

Preimage, 4, 15

Sketch, 4, 5, 8, 9, 11, 16

UPS2, 17

W1_parallel, 15, 17