Package ‘VBmix’

August 24, 2015

Version 0.3.1
Date 2015-08-12
Author Pierrick Bruneau
Maintainer Pierrick Bruneau <pbruneau@gmail.com>
Title Variational Bayesian Mixture Models
Description Variational algorithms and methods for fitting mixture models. Main functions are varbayes, vbcomp, vbconstr, mppca, mmppca and gmmkmsock.
SystemRequirements GSL
Depends R (>= 2.10.0)
Imports lattice, grid, pixmap, mnormt
Suggests e1071, nnet
LazyLoad yes
LazyData yes
License GPL-2 | GPL-3
NeedsCompilation yes
OS_type unix
Repository CRAN
Date/Publication 2015-08-24 16:21:35

R topics documented:

appendToGmm ......................................................... 4
appendToList ....................................................... 4
appendToMppca ..................................................... 5
binnedEntropy ...................................................... 6
buildFrame ......................................................... 6
circlegen .......................................................... 7
constrClassif ...................................................... 8
covgen ............................................................. 9
R topics documented:

dat1sample ......................................................... 10
dat2sample ........................................................ 11
dat3sample ........................................................ 12
datagen ............................................................. 13
dDirichlet ......................................................... 14
displayGraph ..................................................... 14
displayNnet ....................................................... 15
displayScatter .................................................... 16
displaySVM ........................................................ 17
eigenMppca ....................................................... 18
EM ................................................................. 19
extractSimpleModel ............................................. 20
gaussianKL ....................................................... 21
gdist .............................................................. 22
generate2Dtransform ........................................... 23
generateSparsePoints ......................................... 23
getBic .............................................................. 24
getColor .......................................................... 25
getCouple ......................................................... 26
getDataLikelihood .............................................. 27
getLabels ........................................................ 27
getQforComp ..................................................... 28
getResp ........................................................... 29
getVarbayesResp ............................................... 30
gmmdensity ....................................................... 31
gmmgen ........................................................... 31
gmmkmsock ....................................................... 32
gmmpen ........................................................... 33
gmmToMppca ..................................................... 33
gramschmidt ..................................................... 34
gridGen ........................................................... 35
handdat ........................................................... 35
handdomains ..................................................... 36
handlab ........................................................... 36
handnonvoid ..................................................... 37
handvoid ........................................................ 37
imglabels ........................................................ 38
imgmods .......................................................... 38
imgnames ......................................................... 39
incremMerge ..................................................... 39
irisdata ......................................................... 40
irislabels ....................................................... 40
isNonVoid ....................................................... 41
jsmc ............................................................... 41
jsut ............................................................... 42
klmc .............................................................. 43
klut ............................................................... 44
l2norm ........................................................... 45
R topics documented:

mergeClassif .......................................................... 45
mixKnn ................................................................. 46
mkmeans ............................................................... 47
mppca ................................................................. 48
mppca ................................................................. 49
mppcaToGmm .......................................................... 50
multinomial ............................................................ 51
mvndensity ............................................................. 52
mvngen ................................................................. 53
mvnradiusdensity ...................................................... 53
mymvn2plot ............................................................ 54
mySmoothScatter ........................................................ 55
newGmm ................................................................. 55
newMppca .............................................................. 56
normalizeVariable ..................................................... 57
normMppca ............................................................. 58
pca ................................................................. 58
pcapen ................................................................. 59
pendat ................................................................. 60
penlab ................................................................. 60
pixmapToVector ........................................................ 61
plotGmm ................................................................. 61
randomGmm ........................................................... 62
rDirichlet .............................................................. 63
reBuild ................................................................. 64
RGBtoLab ............................................................... 65
sampleClassif .......................................................... 66
semispheregen ........................................................ 67
setDomain .............................................................. 67
sort_index ............................................................. 68
spiralgen ............................................................... 69
subGmm ................................................................. 70
subMppca .............................................................. 70
subVarbayes ........................................................... 71
varbayes ............................................................... 72
vbcomp ................................................................. 73
vbconstr .............................................................. 74
vbpden ................................................................. 75
ZtoLabels ............................................................... 76

Index .................................................................. 77
**appendToGmm**

**Description**
concatenates mod2 to mod1.

**Usage**
appendToGmm(mod1, mod2)

**Arguments**
- **mod1**
  GMM to which mod2 is appended.
- **mod2**
  GMM appended to mod1.

**Value**
GMM with concatenated models, with a set accordingly.

**Author(s)**
Pierrick Bruneau

**Examples**
```
temp <- appendToGmm(gmmpen[[1]], gmmpen[[2]])
```

---

**appendToList**

**Description**
appends 1 list object to another.

**Usage**
appendToList(lst, obj, appendList = FALSE)

**Arguments**
- **lst**
  list object to which we append an object.
- **obj**
  object to append.
- **appendList**
  if TRUE, obj should be a list object, which elements are appended. if FALSE, obj is simply added to lst.
**Value**

list object with obj appended to lst.

**Author(s)**

Pierrick Bruneau

**See Also**

appendToGmm appendToMppca

**Examples**

```r
temp <- list()
temp <- appendToList(temp, pcapen[[1]]$mumean, appendList=TRUE)
temp <- appendToList(temp, pcapen[[2]]$mumean, appendList=TRUE)
```

---

**Description**

appends mppca2 to mppca1.

**Usage**

```r
appendToMppca(mppca1, mppca2)
```

**Arguments**

- `mppca1`: MPPCA model to be appended to.
- `mppca2`: MPPCA to append to mpp1.

**Value**

appended models.

**Author(s)**

Pierrick Bruneau

**See Also**

appendToGmm appendToList

**Examples**

```r
temp <- appendToMppca(pcapen[[1]], pcapen[[2]])
```
**binnedEntropy**

**Description**

uses bins to approximate the empirical entropy of a variable.

**Usage**

\[
\text{binnedEntropy}(v, \text{nbins} = 100)
\]

**Arguments**

- \(v\) a numeric vector.
- \(\text{nbins}\) number of bins used to estimate the entropy.

**Value**

entropy value.

**Author(s)**

Pierrick Bruneau

**Examples**

```r
temp <- binnedEntropy(irisdata[, 1])
```

---

**buildFrame**

**Description**

builds a data frame from a matrix of elements and a vector of numeric labels.

**Usage**

\[
\text{buildFrame}(\text{datamatrix}, \text{labels}, \text{dims} = 1:2)
\]

**Arguments**

- \(\text{datamatrix}\) matrix of row-elements.
- \(\text{labels}\) vector of numeric labels.
- \(\text{dims}\) subset of variables extracted from \(\text{datamatrix}\).
Value

built data frame.

Author(s)

Pierrick Bruneau

Examples

circlegen(npts = 200, radius = 10, noise = 1)

Arguments

npts number of elements to generate.
radius radius of the circle.
noise determines the width of the circle stroke.

Value

matrix of sampled row-elements.

Author(s)

Pierrick Bruneau

Examples

temp <- circlegen()
Description

performs task analogous to mixKnn (i.e. leave-one-out classification), but uses synthetic representatives to infer labels, instead of k-NN. Each representative is obtained by concatenating all GMM (i.e. elements) of a specific label value, and applying vbconstr on this redundant mixture.

Usage

cstrClassif(data, labels, KLparam = 500, rho = new.env())

Arguments

data list of GMM.
labels vector of numeric labels associated to data.
KLparam number of samples for jsmc.
rho R environment object. Used to issue R commands within the C routine.

Value

classification error ratio in [0,1].

Author(s)

Pierrick Bruneau

See Also

mixKnn vbconstr

Examples

temp1 <- sample(1:200, 150)
temp2 <- list()
for(i in temp1) temp2 <- appendToList(temp2, imgmods[[i]])
temp3 <- imglabels[temp1]
# de-activated because this process is very long...
#temp4 <- cstrClassif(temp2, temp3)
Description

generates random definite positive matrices (i.e. valid covariance matrices).

Usage

covgen(d = 2, bounds = c(1, 5))

Arguments

d rank of the square matrix to be returned.

bounds minima and maximal values for diagonal values.

Value

random definite positive matrix

Note

Matrix cells are sampled with an heuristic not guaranteed to lead to definite positiveness: this characteristic is only controlled before function return. If positive definite after control, the matrix is returned. If not, an error message is issued.

Author(s)

Pierrick Bruneau

See Also

randomGmm

Examples

temp <- covgen()
dat1sample

Description

generates data elements according to SYN1 process (sample from a 2D GMM, linearly transformed with additive noise, see reference).

Usage

dat1sample(nelts, gmm, noise, transform=generate2Dtransform(2),
oldbounds = NULL, newbounds = NULL)

Arguments

- **nelts**: number of elements to generate.
- **gmm**: 2D GMM to be sampled from.
- **noise**: additive noise magnitude.
- **transform**: matrix defining linear transform. Defaults to I.
- **oldbounds**: optional argument for sample rescaling. If not NULL, transmitted to setDomain as oldspan.
- **newbounds**: optional argument for sample rescaling. If not NULL, transmitted to setDomain as newspan.

Value

matrix of sampled row-elements

Author(s)

Pierrick Bruneau

References


See Also

dat2sample dat3sample

Examples

temp <- dat1sample(500, randomGmm(), I, generate2Dtransform())
**Description**

generates data elements according to SYN2 process (sample along a semi-sphere with additive noise, see reference).

**Usage**

dat2sample(nelts, radius, noise, oldbounds = NULL, newbounds = NULL)

**Arguments**

- `nelts` number of elements to generate.
- `radius` radius of the sphere to sample from.
- `noise` additive noise magnitude.
- `oldbounds` optional argument for sample rescaling. If not NULL, transmitted to setDomain as oldspan.
- `newbounds` optional argument for sample rescaling. If not NULL, transmitted to setDomain as newspan.

**Value**

matrix of sampled row-elements.

**Author(s)**

Pierrick Bruneau

**References**


**See Also**

dat1sample dat3sample

**Examples**

temp <- dat2sample(500, 10, 1)
dat3sample
dat3sample
dat3sample

Description
generates data elements according to SYN3 process (sample along a 2D circle with additive noise, and linearly transform to higher dimensional space with further noise addition, see reference).

Usage
dat3sample(nelts, radius, noise, transform=generate2Dtransform(2),
oldbounds = NULL, newbounds = NULL)

Arguments
nelts number of elements to generate.
radius radius of the sphere to sample from.
noise additive noise magnitude.
transform matrix defining linear transform. Defaults to I.
oldbounds optional argument for sample rescaling. If not NULL, transmitted to setDomain as oldspan.
newbounds optional argument for sample rescaling. If not NULL, transmitted to setDomain as newspan.

Value
matrix of sampled row-elements.

Author(s)
Pierrick Bruneau

References

See Also
dat1sample dat2sample

Examples
temp <- dat3sample(500, 10, 1, generate2Dtransform())
datagen

Description

generates data from a random multivariate Gaussian, and adds redundant dimensions by random linear combinations with noise.

Usage

datagen(dreal = 2, deff = 6, npts = 200, noise = 0.1, genmean = rep(0, dreal), genspan = 6, iso = FALSE)

Arguments

dreal dimensionality of the multivariate Gaussian.
deff dimensionality of the returned sample.
npts number of elements to be sampled.
noise noise magnitude for the linear combination.
genmean mean of the multivariate Gaussian.
genspan maximal magnitude of the diagonal elements in the covariance matrix. Non-diagonal elements are sampled under constraints of positive-definiteness.
iso sample from an isotropic multivariate Gaussian (i.e. diagonal covariance matrix).

Value

matrix of sampled row-elements.

Author(s)

Pierrick Bruneau

Examples

temp <- datagen()
**ddirichlet**

**Description**

get density of a sample w.r.t Dirichlet distribution (3D only).

**Usage**

```r
dDirichlet(alpha = 0.1, x1, x2)
```

**Arguments**

- `alpha`: alpha parameter of the distribution (i.e. alpha repeated 3 times).
- `x1`: 1st dimension of the sample.
- `x2`: 2nd dimension of the sample.

**Value**

density value.

**Author(s)**

Pierrick Bruneau

**See Also**

`rDirichlet`

**Examples**

```r
temp <- dDirichlet(x1=0.4, x2=0.2)
# 3rd dimension is 1-x1-x2 = 0.2
```

**displayGraph**

**Description**

displays a curve (vect, measure), and associated deviations. Typically used to present experimental results.

**Usage**

```r
displayGraph(measure, dev, vect, xlab = "K", ylab = "measure", main = "")
```
displayNnet

Arguments

measure  y-axis for the curve.
deve   deviations for the y-axis measures.
vect  x-axis for the curve.
xlab  label for x-axis.
ylab  label for y-axis.
main main label for the plotting window.

Value

a new plotting window displaying the curve.

Author(s)

Pierrick Bruneau

Examples

displayGraph(rnorm(10, mean=4, sd=3), rnorm(10, mean=0, sd=0.5), 1:10)

displayNnet  displayNnet

Description

displays the colored decision regions of a neural network model. Data symbols are also optionally displayed. Data and model should be 2D.

Usage

displayNnet(nnet.model, datamatrix, datalabels, subset = NULL,
displayPoints = TRUE, steps = 100, alpha = 0.4, lwd = 1)

Arguments

nnet.model a neural network model, as returned by nnet (nnet library)
datamatrix a matrix of row-elements.
datalabels matrix of binary indicator variables for labels (as used by nnet).
subset vector of indexes of a data subset to be displayed. If NULL, all points are displayed.
displayPoints if FALSE, only decision regions are displayed.
steps influences the resolution of the decision regions. Low values will provoke aliasing, high values are slower to be displayed.
alpa alpha blending parameter between decision regions and data symbols.
lwd magnification factor for the stroke width used to plot symbols.
**Value**

a new plotting window displaying decision regions associated to the parametrized neural network.

**Author(s)**

Pierrick Bruneau

**See Also**

nnet

**Examples**

```r
temp <- nnet::class.ind(irislabels)
temp2 <- setDomain(irisdata[,1:2], 10)
temp3 <- nnet::nnet(temp2, temp, size=10)
displayNnet(temp3, temp2, temp)
```

**Description**

general plotting function for data sets (matrix of row-elements), optionally associated to labels and a GMM. Labels influence the color and symbols of plotted data points. Gaussian envelopes of the components in the GMM are drawn. NB: data set and GMM arguments cannot be both NULL.

**Usage**

displayScatter(data = NULL, model = NULL, labels = NULL, datasizes = NULL, compcolors = NULL, complabels = NULL, compstrokes = "solid", space = 1:2, xlim = NULL, ylim = NULL, main = "", xlab = "", ylab = "", smooth = FALSE, alphacol = 0.8, alphanocol = 0.5, cex.lab = 1, lwd = 1)

**Arguments**

data matrix of row-elements. If NULL, the GMM is plotted alone.
model GMM object.
labels vector of numeric labels. May alternatively be present as a member of model, labels.
datasizes vector of integer magnification factors for data symbols. If length=1, same co-efficient applies to all points.
compcolors vector of integer color indexes. These indexes are internally associated to one color among a set of appropriately chosen ones. If length=1, all GMM components are colored the same way. If length=k, each component is associated to its own color index. This k-length vector may contain NA values: associated components will be white-colored.
displaySVM

complabels character vector containing text strings to be printed over Gaussian envelopes.

compstrokes this character vector may be used to specify non default strokes for envelopes.

space this function prints a 2D scatterplot. If data and model have higher dimension-

xlim bounds for the first variable. If NULL, will be inferred from available data.

ylim bounds for the second variable. If NULL, will be inferred from available data.

main main label for the plotting window.

xlab label for the x-axis.

ylab label for the y-axis.

smooth if TRUE, display the response to a kernel density function, instead of symbols for data elements.

alphacol alpha blending parameter when a component is non-white colored.
alphanocol alpha blending parameter when a component is white colored.
cex.lab magnification factor for all text in the plotting window.
lwd width of the stroke used for data symbols.

Value

a new plotting window displaying the data set and associated model.

Author(s)

Pierrick Bruneau

See Also

plotGmm

displayScatter(irisdata, NULL, irislables)

displaySVM

displaySVM

Description

displays the colored decision regions of a SVM model. Data symbols are also optionally displayed. Data and model should be 2D.

Usage

displaySVM(svm.model, dataframe, displayPoints = TRUE,

subset = NULL, steps = 100, alpha = 0.4, lwd = 1)
eigenMppca

Arguments

  - **svm.model**: a SVM model, as returned by `svm` (e1071 library)
  - **dataframe**: data.frame object, containing row-elements, and associated labels in the last variable.
  - **displayPoints**: if FALSE, only decision regions are displayed.
  - **subset**: vector of indexes of a data subset to be displayed. If NULL, all points are displayed.
  - **steps**: influences the resolution of the decision regions. Low values will provoke aliasing, high values are slower to be displayed.
  - **alpha**: alpha blending parameter between decision regions and data symbols.
  - **lwd**: magnification factor for the stroke width used to plot symbols.

Value

  a new plotting window displaying SVM decision regions.

Author(s)

  Pierrick Bruneau

See Also

  svm

Examples

```r
# extract 2 first variables and build data.frame
temp <- buildFrame(irisdata, irislabels)
iris.model <- e1071::svm(labels ~ ., data=temp, cost=100, gamma=1)
displaySVM(iris.model, temp)
```

eigenMppca

Description

  uses eigen decompositions to align factor matrices to principal bases (see references). NB: mppca and mmppca already perform this operation during their post-processing.

Usage

  `eigenMppca(mod)`

Arguments

  - **mod**: MPPCA model which components have to be aligned.
Value
adjusted MPPCA.

Author(s)
Pierrick Bruneau

References

See Also
mppca newMppca

Examples
temp <- eigenMppca(pcapen[[2]])

Description
estimates a GMM on data using EM algorithm.

Usage
EM(data, ncomp, model=c("general", "diagonal", "spherical"), class=FALSE, thres = 0.1, maxit = NULL, rbic=FALSE, debug=FALSE)

Arguments
data matrix of row-elements.
ncomp maximal number of components in the GMM. In case of degeneracies, the final model size may be less than ncomp.
model Hypothesis on the model to estimate: "general", "diagonal" or "spherical" covariance matrices.
class If TRUE, hard allocate elements in the E step (see CEM variant in Biernacki et al.). If FALSE, compute soft responsibilities as in usual EM algorithm.
thres threshold for lower bound variations between 2 iterations. Convergence is decided if this variation is below thres.
maxit if NULL, the stopping criterion is related to thres. If not NULL, maxit iterations are performed.
rbic if FALSE, output BIC criterion associated to the obtained GMM. If TRUE, use a variant that accounts for the dimensionality of the model.
debug if TRUE, display debug markers.
extractSimpleModel

Value

estimated GMM with at most ncomp components, with labels containing associated labels for data in addition.

labels  Cluster labels taking values in 1..k
w        Numeric vector of cluster weights
mean     List of mean vectors
cov      List of covariance matrices
likelihood Likelihood value of the model
bic      BIC criterion of the model

Author(s)

Pierrick Bruneau

References

Bishop, C. M. (2006) _Pattern Recognition and Machine Learning_. Chapter 9, Springer. Bier-
nacki, C. et al. _Model-based cluster and discriminant analysis with the MIXMOD software_.,

See Also

newGmm varbayes

Examples

temp <- EM(irisdata, 4)

extractSimpleModel

Description

extracts a GMM from a posterior variational distribution. Only relevant components (i.e. associated to a significant population) are extracted.

Usage

extractSimpleModel(model = model, labels = FALSE)

Arguments

model      variational posterior.
labels     boolean indicating wether to extract a label vector. If TRUE, model, a list object, should also contain a data attribute, used to build label vector.
\textit{gaussianKL}

\textbf{Value}

GMM object.

\textbf{Author(s)}

Pierrick Bruneau

\textbf{See Also}

\texttt{varbayes \subVarbayes}

\textbf{Examples}

\begin{verbatim}
  temp <- varbayes(irisdata, 20)
  temp2 <- extractSimpleModel(temp)
\end{verbatim}

\textbf{Description}

computes $\text{KL}( N(0, \Sigma_0) \parallel N(0, \Sigma_1) )$.

\textbf{Usage}

\begin{verbatim}
  gaussianKL(N0, N1)
\end{verbatim}

\textbf{Arguments}

\begin{verbatim}
  N0  Sigma_0
  N1  Sigma_1
\end{verbatim}

\textbf{Value}

KL value.

\textbf{Author(s)}

Pierrick Bruneau

\textbf{See Also}

\texttt{kmc}

\textbf{Examples}

\begin{verbatim}
  temp <- gaussianKL(gmmpen[[1]]$cov[[1]], gmmpen[[1]]$cov[[2]])
\end{verbatim}
**gdist**  
*Pairwise distance between groups*

**Description**

dist computes distances between every pair of elements in a group: gdist takes two arguments, and returns the matrix of distances wrt every possible pair with one argument from each group. General Mahalanobis metrics are also allowed.

**Usage**

```
gdist(g1, g2, metric = NULL)
```

**Arguments**

- **g1**: n1 x d matrix with n1 data elements.
- **g2**: n2 x d matrix with n2 data elements.
- **metric**: If NULL, defaults to identity (i.e. Euclidean distance). A d x d matrix is assumed as a sample covariance matrix, and its inverse is used to compute distances (i.e. Mahalanobis distance). Likewise, a list of n2 d x d matrices can be provided, yielding distances specific to each row in g2.

**Details**

This function is especially useful to make algorithms such as k-means (or mkmeans in the package) more efficient - rows in g1 are then generally the data set, and in g2 respectively cluster centres.

**Value**

n1 x n2 matrix of distances

**Author(s)**

P. Bruneau

**See Also**

`dist, mkmeans`

**Examples**

```
dists <- gdist(irisdata, irisdata[c(1,11,21),])
```
generate2Dtransform

Description

generate a random matrix to transform a 2D signal to higher dimensional spaces.

Usage

generate2Dtransform(dims = 4)

Arguments

dims dimensionality of the target space.

Value

a dims x 2 matrix defining the transform.

Author(s)

Pierrick Bruneau

See Also

dat1sample dat3sample

Examples

temp <- generate2Dtransform()

generateSparsePoints

generateSparsePoints

Description

generates a set of points pairwise-separated by a minimal distance. Is not guaranteed to converge: when maxit is reached, current points are returned.

Usage

generateSparsePoints(npoints, dim = 2, span = 10, mindist = 2, maxit = 20)
**Arguments**

- `npoints`: number of points to generate (i.e. in a matrix with elements as rows).
- `dim`: number of variables to generate.
- `span`: `[-span, span]` is used as bounds to uniform sampling for all variables.
- `mindist`: minimal distance that each element should have with all others. The "control" C routine is used to perform this verification. All points that do not respect this constraint are resampled.
- `maxit`: maximal number of iterations before current elements are returned.

**Value**

A matrix with well separated elements as its rows.

**Author(s)**

Pierrick Bruneau

**Examples**

```r
temp <- generateSparsePoints(10)
```

---

**Description**

Computes BIC criterion (see references) for a specific GMM and data set.

**Usage**

```r
getBic(gmm, dat)
```

**Arguments**

- `gmm`: GMM object.
- `dat`: matrix of row-elements.

**Value**

BIC estimate.

**Author(s)**

Pierrick Bruneau
**getColor**  

**References**  

**See Also**  
getDataLikelihood varbayes

**Examples**  

temp <- getBic(gmppen[[1]], pendat)

---

**Description**  
associates a R color name (i.e. in the output of colors()) to each possible integer input index. Colors are chosen in a reduced, well differentiated, subset.

**Usage**  
getColor(index)

**Arguments**  
index integer input index.

**Value**  
color name.

**Author(s)**  
Pierrick Bruneau

**Examples**  
getColor(3)
Description

computes classification error function described in references, a.k.a couple error. In brief, evaluates how elements are gathered similarly, irrespectively of exact label values (adapted to clustering).

Usage

getCouple(vec1, vec2)

Arguments

vec1 vector of numeric labels.
vec2 vector of numeric labels.

Value

classification error in [0,1].

Author(s)

Pierrick Bruneau

References


Examples

temp <- EM(irisdata, 4)
getCouple(temp$labels, irislabels)
Description

gets log-likelihoods associated to a matrix of row-elements.

Usage

getdatalikelihood(gmm, dat)

Arguments

  gmm  GMM object.
  dat  matrix of row-elements.

Value

numeric vector of log-likelihoods.

Author(s)

Pierrick Bruneau

See Also

getBic gmmgen

Examples

temp <- getdatalikelihood(gmmpen[[3]], pendat)

Description

gets numeric labels that associates a data set and a GMM.

Usage

getLabels(model, data)

Arguments

  model  GMM.
  data  matrix of row-elements.
getQforComp

**Value**
vector of numeric labels, that take values of the respective component indexes in the GMM.

**Author(s)**
Pierrick Bruneau

**See Also**
newGmm

**Examples**
```r
temp <- EM(irisdata, 4)
temp2 <- getLabels(temp, irisdata)
```

**Description**
gets the rank associated with a properly aligned factor matrix.

**Usage**
```r
getQforComp(loadings, tau = 1, verbose = FALSE, quick = FALSE)
```

**Arguments**
- `loadings`: aligned factor matrix.
- `tau`: diagonal noise used for KL computations.
- `verbose`: if TRUE maximal info is displayed.
- `quick`: if TRUE, column norm values are used instead of KL computations (less accurate but faster).

**Value**
rank associated with loadings.

**Author(s)**
Pierrick Bruneau

**References**
**getResp**

See Also

newMppca mppca

Examples

```
getResp
```

getResp

**Description**

get posterior responsibilities of elements in a data set, according to a posterior MPPCA distribution.

**Usage**

```
getResp(data, model)
```

**Arguments**

- **data**
  - matrix of row-elements.

- **model**
  - posterior MPPCA.

**Value**

nxk matrix (with n the number of row-elements, and k the number of components in the MPPCA) of membership probabilities. (i.e. Z in references)

**Author(s)**

Pierrick Bruneau

**References**


**See Also**

mppca

**Examples**

```
temp <- getResp(pendat, pcapen[[1]])
```
**Description**

gets posterior responsibilities for a data set, according to the variational posterior of a GMM.

**Usage**

```r
getVarbayesResp(data, model)
```

**Arguments**

- `data` matrix of row-elements.
- `model` variational posterior of a GMM

**Value**

responsibility matrix (Z in references) resulting from the parameters.

**Author(s)**

Pierrick Bruneau

**References**


**See Also**

- `getResp`  
- `ZtoLabels`

**Examples**

```r
# get resp for only a subsample, as this operation is rather long.  
temp <- getVarbayesResp(pendat[1:10,], vbpen[[2]])
```
\textbf{gmmdensity}

\textbf{Description}

get densities of a set of elements w.r.t a GMM.

\textbf{Usage}

\texttt{gmmdensity(mod, data)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{mod} \hspace{1cm} reference GMM.
  \item \texttt{data} \hspace{1cm} matrix of row-elements.
\end{itemize}

\textbf{Value}

numeric vector containing densities.

\textbf{Author(s)}

Pierrick Bruneau

\textbf{See Also}

gmmgen

\textbf{Examples}

\begin{verbatim}
  temp <- gmmgen(gmmpen[[1]], 50)
  temp2 <- gmmdensity(gmmpen[[1]], temp[[1]])
\end{verbatim}

\textbf{gmmgen}

\textbf{Description}

sample elements from a GMM.

\textbf{Usage}

\texttt{gmmgen(mod, nitem)}
Arguments

mod  GMM sampled from.
nitem  number of elements to be sampled.

Value

nitem x d matrix with elements as rows.

Author(s)

Pierrick Bruneau

Examples

temp <- gmmgen(gmmpen[1], 50)

description

perform k-means specifically designed for a set of GMM (see references). At each iteration, sends information about current prototypes to a server via a socket connection (see references) for info about protocol.

Usage

gmmkmsock(models, names, ngroups, rho = new.env(), host = "127.0.0.1")

Arguments

models  list of GMM objects.
names  character vector with respective names of the GMM objects.
ngroups  (maximal) number of clusters.
rho  R environment object, used for calls to R functions within C code.
host  IP address of the server for the socket (port 1979).

Value

a set of GMM prototypes, and inferred labels (i.e. associated to the input objects).

Note

gmmkmsock includes a socket client that sends formatted data to a server. Detailed information about this protocol may be found in the source package (inst/doc/old_manual.pdf). Simple standalone client and server are also provided (socket/socketclient.cpp and socketserver.cpp). These can be build by running make in the source folder.
Author(s)
Pierrick Bruneau

References

Examples
```
 temp1 <- sample(1:200, 150)
 temp2 <- list()
 for(i in 1:length(temp1)) temp2 <- appendToList(temp2, imgmods[[temp1[i]]])
 temp3 <- imgnames[temp1]
 # next command may be executed only if a server is running on 127.0.0.1:1979.
 # temp4 <- gmmkmsock(temp2, temp3, 5)
```

---

gmmpen

Description
list of 10 GMM objects, estimated on subsets of the original 10992-elements pendat data set.

Format
The format is: List of 10 GMM objects

Examples
```
 temp <- gmmgen(gmmpen[[1]], 1000)
```

---

gmmToMppca

Description
uses eigen decompositions to convert a GMM to a MPPCA model.

Usage
```
gmmToMppca(model, alpha = 500)
```
gramschrmidt

Arguments

model  GMM to be converted.
alpha  GMM are associated to weights, and MPPCA models to population sizes. alpha
       is the chosen population size for the output MPPCA.

Value

converted MPPCA model.

Author(s)

Pierrick Bruneau

See Also

mppcaToGmm

Examples

temp <- gmmToMppca(gmmpen[[3]])

Description

performs Gram-Schmidt orthogonalization on mat.

Usage

gramschrmidt(mat)

Arguments

mat  matrix object to orthogonalize.

Value

orthogonalized matrix.

Author(s)

Pierrick Bruneau

See Also

mppca newMppca
gridGen

Examples

temp <- gram schmidt(pcapen[[3]]$wmean[[1]])

description

generates a matrix valued with a regular grid of 2D coordinates.

Usage

gridGen(xlim = c(-10, 10), ylim = c(-10, 10), step = 50)

Arguments

xlim x bounds.
ylim y bounds.
step size of the square matrix.

Value

'grid' matrix

Author(s)

Pierrick Bruneau

Examples

temp <- gridGen()

description

matrix 300 x 717 of real row-elements. See reference. May be loaded into R with readDataFile.
handdat was built using pixmapToVector and filtering variables with zero entropy.

Format

The format is: num [1:300, 1:717] 10 10 10 10 10 10 10 10 10 10 ...

handdat
Source
http://yann.lecun.com/exdb/mnist/

References

Examples
temp <- reBuild(handdat[123,], handvoid, handnonvoid, handdomains)

<table>
<thead>
<tr>
<th>handdomains</th>
</tr>
</thead>
<tbody>
<tr>
<td>handdomains</td>
</tr>
</tbody>
</table>

Description
original domains of non-void pixels in the handwritten digits collection, to be used along with reBuild.

Format
The format is: List of 2 $: num [1:717] 0.816 0.251 0.278 0.161 0.412 ... $ : num [1:717] 1 1 1 1 1 1 1 ...

Examples
temp <- reBuild(handdat[123,], handvoid, handnonvoid, handdomains)

<table>
<thead>
<tr>
<th>handlab</th>
</tr>
</thead>
<tbody>
<tr>
<td>handlab</td>
</tr>
</tbody>
</table>

Description
vector of numeric labels associated to handdat.

Format
The format is: int [1:300] 0 3 2 0 8 1 3 7 3 7 ...

Source
http://yann.lecun.com/exdb/mnist/
handnonvoid

References


Examples

handlab[1:10]

handnonvoid

Description

vector of non-void pixel indices.

Format

The format is: int [1:717] 8 9 10 11 12 13 14 15 16 17 ...

Examples

temp <- reBuild(handdat[,], handvoid, handnonvoid, handdomains)

handvoid

Description

vector of void pixel indices.

Format

The format is: num [1:67] 1 2 3 4 5 6 7 18 21 24 ...

Examples

temp <- reBuild(handdat[,], handvoid, handnonvoid, handdomains)
**imglabels**

**Description**

vector of numeric labels, indicating the sub-directory in the Caltech-256 collection associated to respective elements in `imgmods`.

**Format**

The format is: `num 1:200 1 1 1 1 1 1 1 1 1 ...`

**Examples**

`imglabels[1:10]`

---

**imgmods**

**Description**

list of 200 3D GMM, sampled from the 1243 images in the 10 first categories of the Caltech-256 image collection. Built using RGBtoLab and varbayes. See reference for information about this image collection.

**Format**

The format is: List of 200 GMM

**References**


**Examples**

`temp <- gmmgen(imgmods[[10]], 1000)`
Description

absolute file paths of respective elements in imgmods.

Format

vector of character objects.

Examples

imgnames[1:10]

Description

updates a reference MPPCA model with an input distribution.

Usage

`incremMerge(modref, newmod, k = 200, nit = 100, quick = FALSE)`

Arguments

- `modref` reference MPPCA to update.
- `newmod` new MPPCA to incorporate.
- `k` number of components of the output variational posterior.
- `nit` number of iterations used in the mmppca call that performs the update.
- `quick` boolean parameter transmitted to the subMppca routine that shrinks the output variational posterior.

Value

updated variational posterior.

Author(s)

Pierrick Bruneau

See Also

mppca mmppca
Examples

```r
# commented for packaging needs (requires approx. 5s)
#temp <- incrMmerge(pcapen[[1]], pcapen[[2]], quick=T)
```

---

### Description

matrix 150 x 4 of row-elements, extracted from iris standard data.frame (4 first variables). See reference.

### Format


### References


### Examples

```r
displayScatter(irisdata)
```

---

### Description

vector of numeric labels associated to irisdata.

### Format

The format is: num [1:150] 1 1 1 1 1 1 1 1 1 ...

### Examples

```r
displayScatter(data=irisdata, labels=irislabels)
```
isNonVoid

Description
checks if loadings contains only void columns.

Usage
isNonVoid(loadings)

Arguments
loadings matrix from which we check the columns.

Value
TRUE if at least 1 column is not void.

Author(s)
Pierrick Bruneau

See Also
mppca newMppca

Examples
isNonVoid(pcapen[[1]][wmean[[2]]])
# [1] TRUE

jsmc

Description
computes Monte Carlo estimate of Jensen-Shannon (JS) divergence between GMM.

Usage
jsmc(mod1, mod2, nsamp = 5000)
Arguments

- **mod1**: GMM parameter to JS\((\text{mod1} \parallel \text{mod2})\).
- **mod2**: GMM parameter to JS\((\text{mod1} \parallel \text{mod2})\).
- **nsamp**: number of samples used to build estimate.

Value

- JS divergence value.

Author(s)

- Pierrick Bruneau

See Also

- klmc gaussianKL

Examples

```r
temp <- jsmc(gmmpen[[1]], gmmpen[[2]])
```

Description

compute Unscented Transform approximation to Jensen-Shannon (JS) divergence between GMM.

Usage

```r
jsut(mod1, mod2)
```

Arguments

- **mod1**: GMM parameter to JS\((\text{mod1} \parallel \text{mod2})\).
- **mod2**: GMM parameter to JS\((\text{mod1} \parallel \text{mod2})\).

Value

- JS divergence value.

Author(s)

- Pierrick Bruneau
klmc

References

See Also
klut jsmc

Examples
```r
temp <- jsut(gmmpen[[1]], gmmpen[[2]])
```

Description
computes Monte Carlo estimate of KL divergence between GMM.

Usage
```r
klmc(mod1, mod2, nsamp = 5000)
```

Arguments
- `mod1`: GMM parameter to KL(mod1 \| mod2).
- `mod2`: GMM parameter to KL(mod1 \| mod2).
- `nsamp`: number of samples used to build estimate.

Value
KL value.

Author(s)
Pierrick Bruneau

See Also
jsmc gaussianKL

Examples
```r
temp <- klmc(gmmpen[[1]], gmmpen[[2]])
```
Description

compute Unscented Transform approximation to KL divergence between GMM.

Usage

klut(mod1, mod2)

Arguments

mod1               GMM parameter to KL(mod1 || mod2).
mod2               GMM parameter to KL(mod1 || mod2).

Value

KL value.

Author(s)

Pierrick Bruneau

References


See Also

klmc

Examples

temp <- klut(gmmpen[[1]], gmmpen[[2]])
**12norm**

---

**Description**

computes Euclidian norm of vec.

**Usage**

12norm(vec)

**Arguments**

vec numeric vector.

**Value**

norm value.

**Author(s)**

Pierrick Bruneau

**Examples**

temp <- 12norm(gmmpen[[2]]$mean[[1]])

---

**mergeClassif**

---

**Description**

performs task analogous to mixKnn (i.e. leave-one-out classification), but uses synthetic representatives to infer labels, instead of k-NN. Each representative is obtained by concatenating all GMM (i.e. elements) of a specific label value, and applying vbcomp on this redundant mixture.

**Usage**

mergeClassif(data, labels, KLparam = 500, rho = new.env())

**Arguments**

data list of GMM.

labels vector of numeric labels associated to data.

KLparam number of samples for jsmc.

rho R environment object. Used to issue R commands within the C routine.
Description

performs k-nearest neighbors over a collection of GMM. It uses jsme to compute distances. Each elements in data is classified against all the others, and inferred class is compared to the true one (leave-one-out).

Usage

mixKnn(data, labels, n = 2, KParam = 500)

Arguments

data list of GMM.
labels vector of numeric labels associated to data.
n k of the algorithm.
KParam number of samples for jsme.

Value

classification error ratio in [0,1].

Author(s)

Pierrick Bruneau
See Also
mergeClassif constrClassif sampleClassif

Examples

```r
temp1 <- sample(1:200, 150)
temp2 <- list()
for(i in temp1) temp2 <- appendToList(temp2, imgmods[[i]])
temp3 <- imglabs[temp1]
# de-activated because this process is very long...
#temp4 <- mixKnn(temp2, temp3)
```

---

**mkmeans**  
*Mahalanobis K-means*

Description

K-means variant that uses a class-wise Mahalanobis metric. The implementation follows somewhat Lloyd’s, with class-wise covariance computation step following that of centres.

Usage

```r
mkmeans(dat, k, maxiter = 100, seeds = NULL)
```

Arguments

- **dat**: Matrix with n rows and d columns of n d-dimensional data elements to cluster.
- **k**: Number of clusters in the output.
- **maxiter**: Maximum number of iterations.
- **seeds**: Optional indexes of initial centres taken in the input data. If NULL, uniform sampling is used.

Details

K-means is characterized by the use of identity as the metric. To remain close to this in spirit, each class-wise covariance matrix is normalized after computation so that its trace equals d. This avoids excessively unbalanced classes, while facilitating the case where the support of a given cluster is less than 2 - covariance cannot be computed in this case. Covariance then defaults to identity. Also to prevent degeneracies when 2 < cluster size < d, a regularization term proportional to sample data features is added to the covariance diagonal.

The returned value follows the GMM data structure (i.e., as returned by e.g. varbayes() and newGmm())
Value

labels Cluster labels taking values in 1..k
w Numeric vector of cluster weights
mean List of mean vectors
cov List of covariance matrices

Author(s)
P. Bruneau

See Also
newGmm, varbayes

Examples

mod <- mkmeans(irisdata, 3)

Description

estimates the variational posterior distribution of a MPPCA that aggregates a collection of input MPPCA models. A lower bound is calculated and monitored at each iteration. This posterior can be used for various purposes (e.g. MC proposal distribution). It can be transformed using mppcaToGmm and subMppca, outputing a GMM. The maximal rank of output factor matrices is determined by the inputs.

Usage

mmppca(mods, ncomp, thres = 0.1, maxit = NULL)

Arguments

mods input MPPCA that concatenates the set of components to aggregate.
ncomp number of components in the posterior.
thres threshold for lower bound variations between 2 iterations. Convergence is decided if this variation is below thres.
maxit if NULL, the stopping criterion is related to thres. If not NULL, maxit iterations are performed.

Value

estimated posterior MPPCA with ncomp components.
Author(s)

Pierrick Bruneau

References


See Also

newMppca mppca subMppca

Examples

temp <- newMppca()
for(i in 1:3) temp <- appendToMppca(temp, pcapen[[i]])
temp2 <- mppca(temp, 50, maxit=30)

Description

estimates the variational posterior distribution of a MPPCA on a data set. A lower bound is calculated and monitored at each iteration. This posterior can be used for various purposes (e.g. MC proposal distribution). It can be transformed using mppcaToGmm and subMppca, outputing a GMM.

Usage

mppca(data, ncomp, thres = 0.1, maxit = NULL, qmax = NULL)

Arguments

data matrix of row-elements.
ncomp number of components in the posterior.
thres threshold for lower bound variations between 2 iterations. Convergence is decided if this variation is below thres.
maxit if NULL, the stopping criterion is related to thres. If not NULL, maxit iterations are performed.
qmax maximal rank of the posterior factor matrices. If NULL, is set to d-1.

Value

estimated posterior MPPCA with ncomp components.
Author(s)
Pierrick Bruneau

References

See Also
newMppca mppcaToGmm subMppca

Examples
# for packaging needs, a low amount of initial components (ie 10) was used.
# A larger amount may be used for better results.
temp <- mppca(pendat, 10, maxit=20, qmax=8)

mppcaToGmm

Description
converts a MPPCA model to a GMM.

Usage
mppcaToGmm(model, notau = FALSE)

Arguments
model MPPCA model to be converted.
notau if TRUE, covariances are built with Lambda.Lambda^T without adding tau.

Value
GMM after conversion.

Author(s)
Pierrick Bruneau

References
**multinomial**

**See Also**
- mppca
- varbayes

**Examples**
```r
temp <- mppcaToGmm(pcapen[[1]])
```

---

**Description**
samples from a k-multinomial.

**Usage**
```r
multinomial(weights, k)
```

**Arguments**
- `weights` numeric vector with the weights of the multinomial. Sum to 1.
- `k` size of the weight vector.

**Value**
an integer value in [1,k], coded as a 1-of-k variable (see reference).

**Author(s)**
Pierrick Bruneau

**References**

**Examples**
```r
weights <- c(0.3, 0.5, 0.2)
multinomial(weights, 3)
# [1] 0 1 0
```
Description

get densities of a set of elements w.r.t a multivariate normal.

Usage

mvndensity(mean, cov, data, rescaled=FALSE)

Arguments

mean numeric vector, mean of the multivariate normal.
cov covariance matrix of the multivariate normal.
data matrix of row-elements.
rescaled if TRUE, a variant accounting for data dimensionality is computed.

Value

numeric vector containing densities.

Author(s)

Pierrick Bruneau

See Also

mvngen

Examples

temp <- mvngen(c(0, 0), diag(2), 5)
mvndensity(c(0,0), diag(2), temp)
# [1] 0.137188286 0.032318242 0.005181099 0.047312602 0.033178600
mvngen

Description
sample nitem elements from N(mean, cov).

Usage
mvngen(mean, cov, nitem)

Arguments
mean numeric vector.
cov covariance matrix.
nitem number of items to generate.

Value
nitem x d matrix with elements as rows (further denoted as a matrix of row-elements).

Author(s)
Pierrick Bruneau

Examples
mvngen(c(0, 0), diag(2), 5)
# [,1] [,2]
# [1,] -0.09898211 1.4516438
# [2,] 0.20814926 -0.1233861
# [3,] 0.18410071 0.5995621
# [4,] 0.65994562 0.8328315
# [5,] 2.33098055 -0.5943117

mvnradiusdensity

Description
get densities of a set of squared radii, i.e. obtained from a Mahalanobis distance computed externally wrt an inverse covariance matrix.

Usage
mvnradiusdensity(cov, radii)
Arguments

- **cov**: Covariance matrix from which we compute the determinant.
- **radii**: Radii wrt which we directly take the density values.

Value

numeric vector containing densities.

Author(s)

Pierrick Bruneau

See Also

mvngen mvdensity

Examples

```r
temp <- mvngen(c(0, 0), diag(2), 5)
R <- chol(solve(diag(2)))
# trivial here, as Cholesky R of I(-1) is I
temp <- temp
mvradiadensity(diag(2), diag(temp))
```

Description
displays mvn envelopes. For internal usage in graphical functions.

Usage

```r
mymvn2plot(w, mu, sigma, k = 15, alone = FALSE, col = NA,
            alphacol = 0.8, alphanocol = 0.5, lty = "solid")
```

Arguments

- **w**: weight of the component.
- **mu**: mean of the component.
- **sigma**: covariance matrix of the component.
- **k**: resolution used for drawing the elliptic envelope.
- **alone**: if TRUE, the component is to be plotted alone in its own window.
- **col**: optional background color for the component.
- **alphacol**: alpha coefficient for a component with a color.
- **alphanocol**: alpha coefficient for a component with no color.
- **lty**: line type for the ellipsis.
\textbf{Description}

Personalized version of smoothScatter. For internal usage in graphical functions.

\textbf{Usage}

\texttt{mySmoothScatter(data, model, xlim, ylim)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{data} \hspace{1cm} matrix of row-elements to plot.
  \item \texttt{model} \hspace{1cm} Optional Gaussian components to plot.
  \item \texttt{xlim} \hspace{1cm} optional bound for plotting.
  \item \texttt{ylim} \hspace{1cm} optional bound for plotting.
\end{itemize}

\textbf{Description}

creates an empty GMM data structure.

\textbf{Usage}

\texttt{newGmm()}

\textbf{Value}

list object with the following members:

\begin{itemize}
  \item \texttt{w} \hspace{1cm} numeric vector containing the component weights of the mixture model.
  \item \texttt{mean} \hspace{1cm} list with respective means (numeric vectors) as elements.
  \item \texttt{cov} \hspace{1cm} list with respective covariance matrices as elements.
  \item \texttt{a} \hspace{1cm} constraints between components, encoded in a numeric vector. One value per component. 2 components associated to the same value are said to be from the same origin. Used in \texttt{vbconstr}.
\end{itemize}

\textbf{Author(s)}

Pierrick Bruneau
newMppca

See Also

varbayes vbconstr

Examples

temp <- newGmm()

Description

creates an empty posterior MPPCA data structure.

Usage

newMppca()

Value

list object with the following members:

- `alpha` numeric vector for bayesian alpha parameter.
- `numoment` list of numeric vectors, containing E[nu_(kj)] parameters.
- `nub` list of numeric vectors, containing b_(kj) parameters for nu.
- `taumoment` numeric vector for tau parameter. NB: all set identically and statically to 1, as in [Bruneau 2011] a single static tau parameter is used.
- `taua` numeric vector for a_k parameters for tau.
- `taub` numeric vector for b_k parameters for tau.
- `wmean` list of matrices containing E[Lambda_k] parameters.
- `wsigma` list of matrices containing Cov(Lambda_k^(i.)).
- `xsigma` list of matrices containing Cov(x_k).
- `mumean` list of numeric vectors, containing means of the MPPCA model.
- `musigma` list of matrices with covariances for the mean estimates.
- `mustar` list of numeric vectors, containing prior means of the MPPCA model, used for initialisation.

Author(s)

Pierrick Bruneau
normalizeVariable

References


See Also

mppca mmppca

Examples

temp <- newMppca()

temp <- normalizeVariable(irisdata[,1])
**Description**

adjusts a MPPCA model to ensure that all factor matrices have same rank (q).

**Usage**

\[
\text{nrmMppca(mppca1)}
\]

**Arguments**

- \( \text{mppca1} \) MPPCA model to be adjusted.

**Value**

adjusted MPPCA.

**Author(s)**

Pierrick Bruneau

**See Also**

newMppca, mppca

**Examples**

```r
temp <- newMppca()
for(i in 1:5) temp <- appendToMppca(temp, pcapen[i])
temp <- nrmMppca(temp)
```

---

**Description**

transforms a data set, and returns coordinates in the principal basis.

**Usage**

\[
\text{pca(dat, ncomp = NULL)}
\]
Arguments

dat      matrix of row-elements.
ncomp    number of retained variables in the output result. If NULL, all transformed variables are returned.

Value

matrix of transformed row-elements.

Author(s)

Pierrick Bruneau

References


See Also

mppca

Examples

tmp <- pca(irisdata, 3)

Description

list of 10 MPPCA posterior objects, estimated on subsets of the original 10992-elements pendat data set.

Format

The format is: List of 10 posterior MPPCA objects

Examples

tmp <- mppcatoGmm(pcpen[[1]])
Description

matrix 2000 x 16 of real row-elements.

Format

The format is: num [1:2000, 1:16] -4.6 -1.2 -2.4 8.4 0.6 3.8 -10 8.8 -10 4.4 ...

Source

http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits

References


Examples

displayScatter(pendat)

displayscatter(data=pendat, labels=penlab)
bitmapToVector

Description

converts a pixmapGrey object to a numeric vector. The pixel matrix is casted to a vector by appending successive columns.

Usage

bitmapToVector(p)

Arguments

p  pixmapGrey object.

Value

numeric vector containing pixel intensities.

Author(s)

Pierrick Bruneau

See Also

bitmapGrey reBuild readPixmapFile

Examples

# use with path to actual train-... file
#temp <- readPixmapFile("data/train-images.idx3-ubyte")
#temp2 <- bitmapToVector(temp[[3]])

plotGmm

Description

3D density plot of a 2D GMM.

Usage

plotGmm(mod, steps=200)
randomGmm

Arguments

mod
steps

GMM object to plot
specifies the horizontal and vertical amount of vertices used to build the wireframe plot.

Value

a new plotting window with the 3D density plot.

Author(s)

Pierrick Bruneau

See Also

displayScatter

densityScatter

covgen

covgen newGmm

densityScatter

densityScatter

Examples

# a larger number of steps (eg 200) should be used for a visually effective 3D plot.
plotGmm(randomGmm(), steps=20)

randomGmm

randomGmm

Description

sample randomly a GMM. Number of components is sampled from a Poisson law, means uniformly from \([-\text{domain}, \text{domain}]\), and covariance matrices using covgen function.

Usage

randomGmm(domain = 10)

Arguments

domain
determines the domain from which means are sampled.

Value

randomly sampled GMM.

Author(s)

Pierrick Bruneau

See Also

covgen newGmm
**rDirichlet**

**Examples**

```r
temp <- randomGmm()
```

---

**Description**

samples from the Dirichlet distribution.

**Usage**

```r
rDirichlet(K, alpha = 0.1)
```

**Arguments**

- **K**: order of the sample.
- **alpha**: alpha parameter of the distribution (i.e. alpha repeated K times).

**Value**

numeric vector, which values are in [0,1] and sum to 1.

**Author(s)**

Pierrick Bruneau

**See Also**

dDirichlet

**Examples**

```r
temp <- rDirichlet(4)
```
Description

re-build a pixmapGrey object from a vector of pixel intensities. As some pixels may be irrelevant over a collection of images (e.g. pixel always white in handwritten digits), some variables may have been filtered or transformed before performing some machine learning process. These transforms are indicated as parameters, and give clues to recover objects in the original image space. NB: assumes that v is scaled in [-10,10]. Additional transformations may thus be performed as appropriate before using this function.

Usage

reBuild(v, voids, nonvoids, domains, placeholder = 1)

Arguments

v vector to be converted to a pixmapGrey object.
voids vector of position indices in the original signal (i.e. 2D matrix with its columns casted in a vector) that did not carry any information. Replaced by a placeholder in recovered image.
nonvoids vector of positions to which v should be associated in the recovered image.
domains original data domains of pixel intensities prior to being transformed to v's domain. Permit appropriate reconstruction in the domain of pixel intensities used by pixmap (i.e. subset of [0,1]). Formatted similarly to what is required in setDomain.
placeholder placeholder value for pixel positions present in voids.

Value

pixmapGrey reconstructed object.

Author(s)

Pierrick Bruneau

See Also

pixmapGrey pixmapToVector

Examples

temp <- reBuild(handdat[123,], handvoid, handnonvoid, handdomains)
**Description**

transform a .ppm file into a matrix of (L,a,b) pixel intensities (1 row-element per pixel).

**Usage**

```r
RGBtoLab(filename, filterWhite = FALSE, addCoords = TRUE)
```

**Arguments**

- **filename**: path to a .ppm file. Alternatively, if needed, R file path manipulating routines are documented in document r-lang.pdf, section 7.1)
- **filterWhite**: if TRUE, filter white points from result to return.
- **addCoords**: if TRUE, append 2 normalized (x,y) coordinates for each pixel.

**Value**

matrix of pixel row-elements.

**Note**

In order to save space, images associated to names in imgnames were not provided in this bundle. Caltech-256 should be retrieved first, converted to .ppm (e.g. with imageMagick), and then values in imgnames associated to relevants file paths, before using RGBtoLab.

**Author(s)**

Pierrick Bruneau

**Examples**

```r
# image collections are large, thus not provided.
# The following commented example relates to a member of this image collection.
#temp <- RGBtoLab(imgnames[[2]], filterWhite=TRUE)
```
Description

performs task analogous to mixKnn (i.e. leave-one-out classification), but uses synthetic representatives to infer labels, instead of k-NN. Each representative is obtained by concatenating all GMM (i.e. elements) of a specific label value, resampling from this redundant mixture, and applying varbayes on this sample.

Usage

sampleClassif(data, labels, KParam = 500, rho = new.env())

Arguments

data list of GMM.
labels vector of numeric labels associated to data.
KParam number of samples for jsma.
rho R environment object. Used to issue R commands within the C routine.

Value
classification error ratio in [0,1].

Author(s)
Pierrick Bruneau

See Also
mixKnn

Examples

temp1 <- sample(1:200, 150)
temp2 <- list()
for(i in temp1) temp2 <- appendToList(temp2, imgmods[[i]])
temp3 <- imglabels[temp1]
# de-activated because this process is very long...
#temp4 <- sampleClassif(temp2, temp3)
semispheregen

Description
sample data points along a semi-sphere.

Usage
semispheregen(npts = 200, radius = 10, noise = 1)

Arguments
- npts: number of elements to be sampled.
- radius: radius of the sphere.
- noise: additive gaussian white noise to the sampled points.

Value
matrix of row-elements with the sampled elements.

Author(s)
Pierrick Bruneau

Examples
```r
temp <- semispheregen()
```

setDomain

Description
performs linear rescaling of given data.

Usage
setDomain(dat, span = 10, oldspan = NULL)
Arguments

dat  
data to rescale. matrix object, with elements as rows, and variables as columns (i.e. variables are rescaled).

span  
ew domain to which dat is rescaled. If type is numeric and length = 1: [-span, span] is used for all variables. If type is numeric and length = 2: [span[1], span[2]] is used for all variables. If a list object: [span[1]_i, span[2]_i] is used for each variable i.

oldspan  
if NULL, old domains are computed from dat inspection. Otherwise, is structured as span and replaces inspected values for rescaling.

Value

scaled data matrix.

Author(s)

Pierrick Bruneau

Examples

temp <- setDomain(irisdata, span=15)

---

sort_index  
sort_index

Description

returns indexes associated to the sorted values of the parameter vector.

Usage

sort_index(vec, order = 0)

Arguments

vec  
vector to be sorted.

order  
if 0, ascending order, if 1, descending order.

Value

indexes associated to the sorted input vector.

Author(s)

Pierrick Bruneau
spiralgen

Examples

```r
temp <- rnorm(10)
temp2 <- sort_index(temp)
```

---

**Description**

generates data elements along a spiral with additional noise.

**Usage**

`spiralgen(radius = 10, n = 1000, laps = 2, noise = 1)`

**Arguments**

- *radius* determines the radius of a spiral revolution.
- *n* number of elements to generate.
- *laps* number of revolutions of the spiral.
- *noise* determines the width of the spiral stroke.

**Value**

matrix of sampled row-elements.

**Author(s)**

Pierrick Bruneau

**See Also**

datagen circlegen

**Examples**

```r
temp <- spiralgen()
```
**Description**

select a subset of components and dimensions from an input GMM.

**Usage**

```r
subGmm(model, dims = c(1, 2), inds = NULL)
```

**Arguments**

- `model`: GMM from which to extract subsets.
- `dims`: numeric vector of the extracted dimensions.
- `inds`: numeric vector of selected components indices. If NULL, all components are selected.

**Value**

subset of input GMM.

**Author(s)**

Pierrick Bruneau

**See Also**

newGmm

**Examples**

```r
temp <- subGmm(gmmPen[[1]], inds=1:3)
```

**Description**

removes unused components and factor columns from model.

**Usage**

```r
subMppca(model, prune = FALSE, thres = 2.001, quick = FALSE, noxmean = TRUE)
```
subVarbayes

Arguments

model  MPPCA model to be shrunk.
prune  if TRUE, void factor columns are removed.
thres  threshold for component selection. A component is selected if alpha > thres.
quick  influences method for void factor columns detection. If FALSE, a KL-based criterion is employed (more accurate). If TRUE, column norms are used (useful for very high dimensional data sets).
noxmean should always be set to TRUE.

Value

shrinked MPPCA model.

Author(s)

Pierrick Bruneau

See Also

mppca newMppca

Examples

# use a subsample of pendat, for runtime (packaging) needs.
temp <- mppca(pendat[sample(1:2000, 150),], 15, qmax=8, maxit=20)
temp2 <- subMppca(temp, prune=TRUE, quick=TRUE)

Description

filters a variational posterior GMM, keeping only components with sufficient support.

Usage

subVarbayes(model, thres = 2.001)

Arguments

model  variational posterior GMM.
thres  minimal support for component selection.

Value

filtered variational posterior GMM.
Author(s)
Pierrick Bruneau

See Also
varbayes extractSimpleModel

Examples
```r
temp <- varbayes(irisdata, 20)
temp2 <- subVarbayes(temp)
```

Description
estimates the variational posterior distribution of a GMM on data using the variational EM algorithm (see references). A lower bound is calculated and monitored at each iteration. This posterior can be used for various purposes (e.g. MC proposal distribution). It can be transformed using extractSimpleModel, outputing a GMM.

Usage
```r
varbayes(data, ncomp, thres = 0.1, maxit = NULL)
```

Arguments
data matrix of row-elements.
ncomp number of components in the posterior.
thres threshold for lower bound variations between 2 iterations. Convergence is decided if this variation is below thres.
maxit if NULL, the stopping criterion is related to thres. If not NULL, maxit iterations are performed.

Value
A list object, with the following items:
model posterior variational distribution.
data a copy of the data parameter.
nk counts, for each iteration, of the population modeled by each Gaussian component.
agitation agitation measures (see Beal 2003 for explanation) for each iteration and Gaussian component.
vbcomp

bound latest monitored bound value (convergence criterion maximized throughout the process).

The model item is structured in a list as follows:

alpha hyperparameters influencing the active components in the posterior.
beta hyperparameters regarding shaping of the Normal-Wishart posteriors.
nu hyperparameters regarding shaping of the Normal-Wishart posteriors.
mean hyperparameters regarding shaping of the Normal-Wishart posteriors.
wish hyperparameters regarding shaping of the Normal-Wishart posteriors.

Author(s)

Pierrick Bruneau

References


See Also

EM extractSimpleModel

Examples

temp <- varbayes(irisdata, 20)

Description

estimates the variational posterior distribution of a GMM that aggregates a collection of GMM. A lower bound is calculated and monitored at each iteration. This posterior can be used for various purposes (e.g. MC proposal distribution). It can be transformed using extractSimpleModel, outputing a GMM.

Usage

vbcomp(models, ncomp, thres = 0.1, maxit = NULL)
Arguments

models  GMM made with the weighted sum of the collection of GMM to aggregate.
ncomp  number of components in the posterior.
thres  threshold for lower bound variations between 2 iterations. Convergence is decided if this variation is below thres.
maxit  if NULL, the stopping criterion is related to thres. If not NULL, maxit iterations are performed.

Value

estimated posterior with ncomp components.

Author(s)

Pierrick Bruneau

References


See Also

varbayes extractSimpleModel

Examples

temp1 <- newGmm()
for(i in 1:10) temp1 <- appendToGmm(temp1, gmmpen[[i]])
temp2 <- vbcomp(temp1, 50)

---

vbconstr  vbconstr

Description

estimates the variational posterior distribution of a GMM that aggregates a constrained collection of GMM. A lower bound is calculated and monitored at each iteration. This posterior can be used for various purposes (e.g. MC proposal distribution). It can be transformed using extractSimpleModel, outputing a GMM.

Usage

vbconstr(models, ncomp, thres = 0.1, maxit = NULL)
Arguments

- `models`: GMM made with the weighted sum of the collection of GMM to aggregate. `a` is used to model constraints between components in this GMM.
- `ncomp`: number of components in the posterior.
- `thres`: threshold for lower bound variations between 2 iterations. Convergence is decided if this variation is below `thres`.
- `maxit`: if NULL, the stopping criterion is related to `thres`. If not NULL, `maxit` iterations are performed.

Value

- estimated posterior with `ncomp` components.

Author(s)
Pierrick Bruneau

References

See Also
vbcomp extractSimpleModel

Examples
```r
tempQ <- M newgmm(I
for(i in QZQPI tempQ <- appendToGmm(tempQL gmmpen[[i]]

tempR <- vbconstr(tempQL UPI

---

vbpen

Description
list of 10 variational posterior GMM objects, estimated on subsets of the original 10992-elements pendat data set.

Format
The format is: List of 10 variational GMM.

Examples
```r
temp <- extractSimpleModel(vbpen[[2]])```
Description
converts a responsibility matrix (Z in references) to a vector of numeric labels.

Usage
ZtoLabels(resp)

Arguments

resp responsibility matrix to convert.

Value
labels vector.

Author(s)
Pierrick Bruneau

References

See Also
getResp getVarbayesResp

Examples

temp <- getResp(pendat, pcapen[[2]])
temp2 <- ZtoLabels(temp)
Index

**Topic datasets**
- gmmpen, 33
- handdat, 35
- handdomains, 36
- handlab, 36
- handnonvoid, 37
- handvoid, 37
- imglabels, 38
- imgmods, 38
- imgnames, 39
- irisdata, 40
- irislabels, 40
- pcapen, 59
- pendat, 60
- penlab, 60
- vbpen, 75

appendToGmm, 4
appendToList, 4
appendToMppca, 5

binnedEntropy, 6
buildFrame, 6

circlegen, 7
constrClassif, 8
covgen, 9

dat1sample, 10
dat2sample, 11
dat3sample, 12
datagen, 13
dDirichlet, 14
displayGraph, 14
displayNnet, 15
displayScatter, 16
displaySVM, 17
dist, 22

eigenMppca, 18
EM, 19

eXRler, 18
extractSimpleModel, 20
gaussianKL, 21
gdist, 22
generate2Dtransform, 23
generateSparsePoints, 23
getBic, 24
ggetColor, 25
getCouple, 26
ggetDataLikelihood, 27
getLabels, 27
getQforComp, 28
getResp, 29
getVarbayesResp, 30
gmmdensity, 31
gmmgen, 31
gmmkssock, 32
gmmpen, 33
gmmToMppca, 33
gramschmidt, 34
gridGen, 35

handdat, 35
handdomains, 36
handlab, 36
handnonvoid, 37
handvoid, 37

imglabels, 38
imgmods, 38
imgnames, 39
incredMerge, 39
irisdata, 40
irislabels, 40
isNonVoid, 41

jsmc, 41
jsut, 42

klmc, 43
klut, 44
l2norm, 45
mergeClassif, 45
mixKnn, 46
mkmeans, 22, 47
mpppc, 48
mppca, 49
mppcaToGmm, 50
multinomial, 51
mvndensity, 52
mvngen, 53
mvnradiusdensity, 53
mymvn2plot, 54
mySmoothScatter, 55
newGmm, 48, 55
newMppca, 56
normalizeVariable, 57
normMppca, 58
pca, 58
pcapen, 59
pendat, 60
penlab, 60
 pixmapToVector, 61
plotGmm, 61
randomGmm, 62
rDirichlet, 63
reBuild, 64
RGBtoLab, 65
sampleClassif, 66
semispheregen, 67
setDomain, 67
sort_index, 68
spiralgen, 69
subGmm, 70
subMppca, 70
subVarbayes, 71
varbayes, 48, 72
vbcomp, 73
vbconstr, 74
vbpen, 75
ZtoLabels, 76