Package ‘VBmix’

February 19, 2015

Version 0.2.17
Date 2014-05-06
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, Qt, fftw3
Depends R (>= 2.10.0)
Imports lattice, grid, pixmap, mnormt
Suggests e1071, nnet
LazyLoad yes
LazyData yes
OS_type unix
License GPL-2 | GPL-3
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-05-06 15:10:50

R topics documented:

appendToGmm .................................. 4
appendToList .................................. 4
appendToMppca ................................ 5
binnedEntropy .................................. 6
buildFrame .................................... 6
circlegen ...................................... 7
classicEM ..................................... 8
constrClassif .................................. 9
topics documented:
covgen ................................................................. 10
dat1sample .......................................................... 11
dat2sample ............................................................ 12
dat3sample ............................................................ 13
datagen ................................................................. 14
dDirichlet ............................................................. 15
displayGraph ........................................................ 15
displayNnet ........................................................... 16
displayScatter ....................................................... 17
displaySVM ........................................................... 18
eigenMppca ........................................................... 19
extractSimpleModel .............................................. 20
gaussianKL ........................................................... 21
generate2Dtransform ............................................. 22
generateSparsePoints ............................................ 22
getBic ................................................................. 23
getColor .............................................................. 24
getCouple ............................................................ 25
getDataLikelihood ............................................... 26
getLabels ............................................................. 26
getQforComp ....................................................... 27
getResp .............................................................. 28
getVarbayesResp .................................................. 29
gmmdensity .......................................................... 30
gmmgen .............................................................. 30
gmmkmsock .......................................................... 31
gmmpen .............................................................. 32
gmmToMppca ........................................................ 32
gramschmidt ........................................................ 33
gridGen .............................................................. 34
handdat .............................................................. 34
handdomains ....................................................... 35
handlab ............................................................. 35
handnonvoid ....................................................... 36
handvoid ........................................................... 36
imglabels ............................................................ 37
imgmods ............................................................. 37
imgnames ........................................................... 38
incremMerge ........................................................ 38
irisdata ............................................................. 39
irislabels ........................................................... 39
isNonVoid .......................................................... 40
jsmc ................................................................. 40
jsut ................................................................. 41
klmc ................................................................. 42
klut ................................................................. 43
l2norm .............................................................. 44
mergeClassif ....................................................... 44
topics documented:

mixKnn ................................................................. 45
mmpca ................................................................. 46
mppca ................................................................. 47
mppcaToGmm ......................................................... 48
multinomial ......................................................... 49
mvndensity ......................................................... 49
mvngen .............................................................. 50
mynmvn2plot ...................................................... 51
mySmoothScatter .................................................. 52
newGmm .............................................................. 52
newMppca ........................................................... 53
normalizeVariable ................................................. 54
normMppca .......................................................... 55
pca ................................................................. 55
pcapen .............................................................. 56
pendat .............................................................. 57
penlab .............................................................. 57
pixmapToVector ................................................... 58
plotGmm ............................................................ 58
randomGmm ....................................................... 59
Rdct ................................................................. 60
Rdct2D .............................................................. 61
rDirichlet ......................................................... 61
readLabelFile .................................................... 62
readPixmapFile .................................................. 63
reBuild ............................................................ 64
RGBtoLab ......................................................... 65
RinvDct2D .......................................................... 66
sampleClassif .................................................... 66
semispheregen ..................................................... 67
setDomain .......................................................... 68
sort_index ......................................................... 69
spiralgen .......................................................... 69
subGmm .............................................................. 70
subMppca .......................................................... 71
subVarbayes ....................................................... 72
varbayes ........................................................... 72
vbcomp ............................................................ 74
vbconstr .......................................................... 75
vben ............................................................... 76
ZtoLabels ........................................................... 76

Index ............................................................... 78
**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.
**appendToMppca**

**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]]$ummean, appendList=TRUE)
temp <- appendToList(temp, pcapen[[2]]$ummean, 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**

binnedEntropy(v, nbins = 100)

**Arguments**

- **v**: a numeric vector.
- **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**

buildFrame(datamatrix, labels, dims = 1:2)

**Arguments**

- **datamatrix**: matrix of row-elements.
- **labels**: vector of numeric labels.
- **dims**: subset of variables extracted from datamatrix.
Value

built data frame.

Author(s)

Pierrick Bruneau

Examples

irisdata[c(1,7,35,56,131),]
# returns:
# Sepal.Length Sepal.Width Petal.Length Petal.Width
#[1,] 5.1 3.5 1.4 0.2
#[2,] 4.6 3.4 1.4 0.3
#[3,] 4.9 3.1 1.5 0.2
#[4,] 5.7 2.8 4.5 1.3
#[5,] 7.4 2.8 6.1 1.9
irislabels[c(1,7,35,56,131)]
# returns:
#[1] 1 1 1 2 3
temp <- buildFrame(irisdata, irislabels, dims=1:4)

circlegen circlegen

description
generate data elements along a 2D circle with additional noise.

Usage

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

estimates a GMM on data using EM algorithm. A lower bound is calculated and monitored at each iteration.

Usage

classicEM(dataL kL thres \] PNQL maxit \] null)

Arguments

data matrix of row-elements.
k maximal number of components in the GMM. In case of degeneracies, the final model size may be less than 0.
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 GMM with at most k components, with labels containing associated labels for data in addition.

Author(s)

Pierrick Bruneau

References


See Also

newGmm varbayes

Examples

temp <- classicEM(irisdata, 4)
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

constrClassif(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 <- constrClassif(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()
**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(), 1, 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nelts</td>
<td>number of elements to generate.</td>
</tr>
<tr>
<td>radius</td>
<td>radius of the sphere to sample from.</td>
</tr>
<tr>
<td>noise</td>
<td>additive noise magnitude.</td>
</tr>
<tr>
<td>oldbounds</td>
<td>optional argument for sample rescaling. If not NULL, transmitted to setDomain as oldspan.</td>
</tr>
<tr>
<td>newbounds</td>
<td>optional argument for sample rescaling. If not NULL, transmitted to setDomain as newspan.</td>
</tr>
</tbody>
</table>

Value

matrix of sampled row-elements.

Author(s)
Pierreck Bruneau

References


See Also

dat1sample dat3sample

Examples

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

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**

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**

displayGraph(measure, dev, vect, xlab = "K", ylab = "measure", main = "")
**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>y-axis for the curve.</td>
</tr>
<tr>
<td>dev</td>
<td>deviations for the y-axis measures.</td>
</tr>
<tr>
<td>vect</td>
<td>x-axis for the curve.</td>
</tr>
<tr>
<td>xlab</td>
<td>label for x-axis.</td>
</tr>
<tr>
<td>ylab</td>
<td>label for y-axis.</td>
</tr>
<tr>
<td>main</td>
<td>main label for the plotting window.</td>
</tr>
</tbody>
</table>

**Value**

a new plotting window displaying the curve.

**Author(s)**

Pierrick Bruneau

**Examples**

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

**Description**

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

**Usage**

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

**Arguments**

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

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

Author(s)

Pierrick Bruneau

See Also

nnet


Examples

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

displayScatter | displayScatter

displayscatter

displayScatter

displayscatter

displayscatter

displayscatter

displayscatter

displayscatter

displayscatter

displayscatter

displayscatter

displayscatter

displayscatter

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 compo-
nents 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.
complabels character vector containing text strings to be printed over Gaussian envelopes.
comppstrokes 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 dimensionality, this argument specifies the axes to be printed.
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

Examples
displayScatter(irisdata, NULL, irislabels)

displaySVM(svm.model, dataframe, displayPoints = TRUE, subset = NULL, steps = 100, alpha = 0.4, lwd = 1)

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

# 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

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

Tipping, M. E. and Bishop, C. M. (1999) _Probabilistic principal component analysis_ , Journal of
the Royal Statistical Society - B Series, Volume 61, Number 3, Pages 611-622.

See Also

mppca newMppca

Examples

```r
temp <- eigenMppca(pcapen[[2]])
```

description

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

Usage

```r
extractSimpleModel(model = model, labels = FALSE)
```

Arguments

- **model**: variational posterior.
- **labels**: boolean indicating whether to extract a label vector. If TRUE, model, a list object,
  should also contain a data attribute, used to build label vector.

Value

GMM object.

Author(s)

Pierrick Bruneau
**gaussianKL**

**See Also**

varbayes subVarbayes

**Examples**

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

---

**gaussianKL**

**Description**

computes \(KL(\ N(0, \Sigma_0) \parallel N(0, \Sigma_1))\).

**Usage**

```r
gaussianKL(N0, N1)
```

**Arguments**

- `N0`  
  \(\Sigma_0\)
- `N1`  
  \(\Sigma_1\)

**Value**

KL value.

**Author(s)**

Pierrick Bruneau

**See Also**

klmc

**Examples**

```r
temp <- gaussianKL(gmmpen[[1]]$cov[[1]], gmmpen[[1]]$cov[[2]])
```
**generate2Dtransform**  

**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**

```r
  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)`
getBic

**Arguments**

- `npoints` number of points to generate (i.e. in a matrix with elements as rows).
- `dim` number of variables to generate.
- `span` \([-\text{span}, \text{span}]\) 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**

- matrix with well separated elements as its rows.

**Author(s)**

- Pierrick Bruneau

**Examples**

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

---

**getBic**

**getBic**

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

- BIC estimate.

**Author(s)**

- Pierrick Bruneau
References


See Also

ggetDataLikelihood varbayes

Examples

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

ggetColor

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

g getColor(index)

Arguments

index integer input index.

Value

color name.

Author(s)

Pierrick Bruneau

Examples

ggetColor(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

getCode(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 <- classicEM(irisdata, 4)
getCode(temp$labels, irislabels)
# [1] 0.1524832
Description

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

Usage

gedataLikelihood(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 <- classicEM(irisdata, 4)
temp2 <- getLabels(temp, irisdata)
```

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

Usage
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
temp <- getQforComp(pcapen[[1]]$wmean[[2]], quick=TRUE)

desc <- getResp(pendat, pcapen[[1]])

desc <- 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]])
```
gmmgen

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

Usage
gmmdensity(mod, data)

Arguments
mod reference GMM.
data matrix of row-elements.

Value
numeric vector containing densities.

Author(s)
Pierrick Bruneau

See Also
gmmgen

Examples
temp <- gmmgen(gmmpen[[1]], 50)
temp2 <- gmmdensity(gmmpen[[1]], temp[[1]])

gmmdensity

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

Usage
gmmdensity(mod, data)

Arguments
mod reference GMM.
data matrix of row-elements.

Value
numeric vector containing densities.

Author(s)
Pierrick Bruneau

See Also
gmmgen

Examples
temp <- gmmgen(gmmpen[[1]], 50)
temp2 <- gmmdensity(gmmpen[[1]], temp[[1]])
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 <- gmmsock(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 <- gmmpen(gmmpen[[1]], 1000)
```

**gmmToMppca**

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

Usage
```
gmmToMppca(model, alpha = 500)
```
**gramschmidt**

**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]])`

---

**gramschmidt**

**Description**
performs Gram-Schmidt orthogonalization on mat.

**Usage**
- `gramschmidt(mat)`

**Arguments**
- `mat`  matrix object to orthogonalize.

**Value**
- orthogonalized matrix.

**Author(s)**
Pierrick Bruneau

**See Also**
mppca newMppca
gridGen

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()

handdat

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 ...
Source

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

References


Examples

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

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 1 1 ...

Examples

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

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/
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[123,], 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[123,], handvoid, handnonvoid, handdomains)
### `imglabs`

**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 ...`

**Examples**

```
imglabs[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

```r
imgnames[1:10]
```

Description

updates a reference MPPCA model with an input distribution.

Usage

```r
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`
Examples

```
# commented for packaging needs (requires approx. 5s)
#temp <- incremMerge(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

```
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 1 ...

Examples

```
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[[Q]]Dwmean[[R]])
# [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(mod1 || mod2).
mod2  GMM parameter to JS(mod1 || mod2).
nsamp number of samples used to build estimate.

Value

JS divergence value.

Author(s)

Pierrick Bruneau

See Also

klmc gaussianKL

Examples

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

Description

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

Usage

jsut(mod1, mod2)

Arguments

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

Value

JS divergence value.

Author(s)

Pierrick Bruneau
References


See Also

klut jsmc

Examples

\[
\text{temp <- jsut(gmmpen[,1], gmmpen[,2])}
\]

Description

computes Monte Carlo estimate of KL divergence between GMM.

Usage

\[
\text{klmc(mod1, mod2, nsamp = 5000)}
\]

Arguments

\[
\begin{align*}
\text{mod1} & \quad \text{GMM parameter to KL(mod1 || mod2).} \\
\text{mod2} & \quad \text{GMM parameter to KL(mod1 || mod2).} \\
\text{nsamp} & \quad \text{number of samples used to build estimate.}
\end{align*}
\]

Value

KL value.

Author(s)

Pierrick Bruneau

See Also

jsmc gaussianKL

Examples

\[
\text{temp <- klmc(gmmpen[,1], gmmpen[,2])}
\]
**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**

```r
temp <- klut(gmmpen[[1]], gmmpen[[2]])
```
l2norm

Description
computes Euclidian norm of vec.

Usage
l2norm(vec)

Arguments
vec numeric vector.

Value
norm value.

Author(s)
Pierrick Bruneau

Examples
temp <- l2norm(gmpen[[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.
mixKnn

**Value**

classification error ratio in [0,1].

**Author(s)**

Pierrick Bruneau

**See Also**

mixKnn vbcomp

**Examples**

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

**Description**

performs k-nearest neighbors over a collection of GMM. It uses jsmc 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, KLparam = 500)

**Arguments**

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

**Value**

classification error ratio in [0,1].

**Author(s)**

Pierrick Bruneau
See Also

mergeClassif constrClassif sampleClassif

Examples

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

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 = .1, maxit = NULL)

Arguments

mods input MPPCA that concatenates the set of components to aggregate.
ncomp number of components in the posterior.
thes threshold for lower bound variations between 2 iterations. Convergence is de-
cided 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

tmp <- newMppca()
for(i in 1:3) tmp <- appendMppca(tmp, pcapen[[i]])
temp2 <- mppca(tmp, 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, outputting 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)

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


See Also

mppca varbayes

Examples

temp <- mppcaToGmm(pcapen[[1]])
multinomial

Description
samples from a k-multinomial.

Usage
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
weights <- c(0.3, 0.5, 0.2)
multinomial(weights, 3)
# [1] 0 1 0

mvndensity

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

Usage
mvndensity(mean, cov, data)
Arguments

- **mean**: numeric vector, mean of the multivariate normal.
- **cov**: covariance matrix of the multivariate normal.
- **data**: matrix of row-elements.

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

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

```r
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
```

Description

displays mvn envelopes. For internal usage in graphical functions.

Usage

```r
mymvn2plot(w, mu, sigma, k = 15, alone = FALSE, col = NA,
alpha = 0.8, alpha = 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.
- `alpha`: alpha coefficient for a component with a color.
- `alpha`: alpha coefficient for a component with no color.
- `lty`: line type for the ellipsis.
Description

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

Usage

mySmoothScatter(data, model, xlim, ylim)

Arguments

data  matrix of row-elements to plot.
model Optional Gaussian components to plot.
xlim  optional bound for plotting.
ylim  optional bound for plotting.

Description

creates an empty GMM data structure.

Usage

newGmm()

Value

list object with the following members:

w numeric vector containing the component weights of the mixture model.
mean list with respective means (numeric vectors) as elements.
cov  list with respective covariance matrices as elements.
a 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 vbconstr.

Author(s)

Pierrick Bruneau
newMppca

See Also

varbayes vbconstr

Examples

temp <- newGmm()

newMppca

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.
emean 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
References


See Also

mppca mmppca

Examples

```
temp <- newMppca()
```

normalizeVariable

**normalizeVariable**

normalizes a variable (numeric vector) in [0,1].

**Usage**

`normalizeVariable(v)`

**Arguments**

- `v` a numeric vector.

**Value**

normalized numeric vector.

**Author(s)**

Pierrick Bruneau

**Examples**

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

**Description**

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

**Usage**

`normMppca(mppca)`

**Arguments**

- `mppca`: 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 <- normMppca(temp)
```

---

**pca**

**Description**

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

**Usage**

`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

temp <- pcap(irisdata, 3)

describe(pcapen, comp)
pendat

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)

penlab

Description
vector of numeric labels associated to pendat.

Format
The format is: int [1:2000] 5 3 8 6 0 9 1 8 1 9 ...

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

References

Examples
displayScatter(data=pendat, labels=penlab)
**Description**

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

**Usage**

` pixmapToVector(p) `  

**Arguments**

- `p`: pixmapGrey object.

**Value**

numeric vector containing pixel intensities.

**Author(s)**

Pierrick Bruneau

**See Also**

` pixmapGrey reBuild readPixmapFile `  

**Examples**

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

---

**Description**

3D density plot of a 2D GMM.

**Usage**

```r
plotGmm(mod, steps=200)
```
**randomGmm**

**Arguments**
- `mod` GMM object to plot
- `steps` specifies the horizontal and vertical amount of vertices used to build the wire-frame plot.

**Value**
- a new plotting window with the 3D density plot.

**Author(s)**
- Pierrick Bruneau

**See Also**
- displayScatter

**Examples**

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

**Description**

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

**Usage**

```r
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
Examples

temp <- randomGmm()

Description

performs DCT on a real vector.

Usage

Rdct(vect)

Arguments

vect vector of real values.

Value

vector of DCT transformed values.

Author(s)

Pierrick Bruneau

See Also

Rdct2D

Examples

temp <- Rdct(irisdata[,1])
\textbf{Rdct2D} \hfill Rdct2D

\textbf{Description}
performs 2D DCT on a real matrix.

\textbf{Usage}
\texttt{Rdct2D(mat)}

\textbf{Arguments}
\begin{itemize}
  \item \texttt{mat} \hspace{1cm} matrix of real values.
\end{itemize}

\textbf{Value}
matrix of DCT transformed values.

\textbf{Author(s)}
Pierrick Bruneau

\textbf{See Also}
Rdct RinvDct2D

\textbf{Examples}
\begin{verbatim}
temp <- Rdct2D(irisdata)
\end{verbatim}

\textbf{rDirichlet} \hfill rDirichlet

\textbf{Description}
samples from the Dirichlet distribution.

\textbf{Usage}
\texttt{rDirichlet(K, alpha = 0.1)}

\textbf{Arguments}
\begin{itemize}
  \item \texttt{K} \hspace{1cm} order of the sample.
  \item \texttt{alpha} \hspace{1cm} alpha parameter of the distribution (i.e. alpha repeated K times).
\end{itemize}
Value

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

Author(s)

Pierrick Bruneau

See Also

dDirichlet

Examples

temp <- rDirichlet(4)

Description

reads the vector of numeric labels contained in a binary file. Labels are associated to handwritten digits, thus in [0-9].

Usage

readLabelFile(name)

Arguments

name path to the file.

Value

vector of digit labels.

Author(s)

Pierrick Bruneau

References


See Also

readPixmapFile
Examples

```r
# use with path to actual train-... file
# temp <- readLabelFile("data/train-labels-idx1-ubyte")
```

Description

extracts a list of pixmap objects from the handwritten digits file format provided in references.

Usage

```r
readPixmapFile(name)
```

Arguments

- **name**: path to the file.

Value

a list of pixmapGrey objects.

Author(s)

Pierrick Bruneau

References


See Also

- pixmapGrey

Examples

```r
# use with path to actual train-... file
#temp <- readPixmapFile("data/train-images-idx3-ubyte")
```
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)
```
RinvDct2D

Description
performs inverse 2D DCT on a real matrix.

Usage
RinvDct2D(mat)

Arguments
mat matrix of real values.

Value
matrix of inverse DCT transformed values.

Author(s)
Pierrick Bruneau

See Also
Rdct Rdct2D

Examples
temp <- RinvDct2D(Rdct2D(irisdata))

sampleClassif

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, 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

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)

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.
setDomain

Author(s)

Pierrick Bruneau

Examples

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 new 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**

**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

**Examples**
```
temp <- rnorm(10)
temp2 <- sort_index(temp)
```

---

**spiralgen**

**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**

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**

temp <- subGmm(gmmpen[[1]], inds=1:3)
Description

removes unused components and factor columns from model.

Usage

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

Arguments

model MPPCA model to be shrunken.
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

shrunk 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
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, outputting a GMM.

Usage
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.
mmaxit 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.
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

classicEM extractSimpleModel

Examples

temp <- varbayes(irisdata, 20)
vbcomp

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, outputting 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

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.
thsres threshold for lower bound variations between 2 iterations. Convergence is de-
cided 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

models with a variational-Bayes approach_, Pattern Recognition, Volume 43, Pages 850-858.

See Also

vbcomp extractSimpleModel

Examples

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

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
temp <- extractSimpleModel(vbpen[[2]])

ZtoLabels

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, 32
    handdat, 34
    handdomains, 35
    handlab, 35
    handnonvoid, 36
    handvoid, 36
    imglabels, 37
    imgmods, 37
    imgnames, 38
    irisdata, 39
    irislabels, 39
    pcapen, 56
    pendat, 57
    penlab, 57
    vbpen, 76

appendToGmm, 4
appendToList, 4
appendToMppca, 5

binnedEntropy, 6
buildFrame, 6

circlegen, 7
classicEM, 8
constrClassif, 9
covgen, 10

dat1sample, 11
dat2sample, 12
dat3sample, 13
datagen, 14
dDirichlet, 15
displayGraph, 15
displayNnet, 16
displayScatter, 17
displaySVM, 18

eigenMppca, 19
extractSimpleModel, 20
gaussianKL, 21
generate2Dtransform, 22
generateSparsePoints, 22
getBic, 23
ggetColor, 24
getCouple, 25
ggetDataLikelihood, 26
getLabels, 26
getQforComp, 27
getResp, 28
getVarbayesResp, 29
gmmDensity, 30
gmmgen, 30
gmmkmsock, 31
gmmpen, 32
gmmToMppca, 32
gramschmidt, 33
gridGen, 34

handdat, 34
handdomains, 35
handlab, 35
handnonvoid, 36
handvoid, 36

imglabels, 37
imgmods, 37
imgnames, 38
incremMerge, 38
irisdata, 39
irislabels, 39
isNonVoid, 40

jsmc, 40
jsut, 41

klmc, 42
klut, 43

l2norm, 44
mergeClassif, 44
mixKnn, 45
mpppca, 46
mppca, 47
mppcaToGmm, 48
multinomial, 49
mvndensity, 49
mvngen, 50
mymvn2plot, 51
mySmoothScatter, 52

newGmm, 52
newMppca, 53
normalizeVariable, 54
normMppca, 55

pca, 55
pcapen, 56
pendat, 57
penlab, 57
pixmapToVector, 58
plotGmm, 58

randomGmm, 59
Rdct, 60
Rdct2D, 61
rDirichlet, 61
readLabelFile, 62
readPixmapFile, 63
reBuild, 64
RGBtoLab, 65
RinvDct2D, 66

sampleClassif, 66
semispheregen, 67
setDomain, 68
sort_index, 69
spiralgen, 69
subGmm, 70
subMppca, 71
subVarbayes, 72

varbayes, 72
vbcomp, 74
vbconstr, 75
vbpen, 76

Ztolabels, 76