

Package ‘vabayelMix’

April 19, 2009

Type Package

Title Variational Bayesian Mixture Modelling

Version 0.3

Date 2006-02-25

Author Andrew E Teschendorff

Maintainer <aet21@cam.ac.uk>

Depends mclust

Description Performs inference of a gaussian mixture model within a bayesian framework using an optimal separable approximation to the posterior density. The optimal posterior approximation is obtained using a variational approach.

License GPL (>= 2)

Repository CRAN

Date/Publication 2006-02-26 20:43:01

R topics documented:

pack	2
UseBasicPrior	2
vabayelMix	4

Index	6
--------------	----------

 pack

Profile analysis using Clustering and Kurtosis

Description

For a data matrix, selects features with kurtosis values in a specified range. Optionally, it further selects features based on whether their profiles are mixtures of gaussians or not.

Usage

```
pack(data.m, kurt.range=c(-2, 0), cluster=T, method=c("bic", "vb"));
```

Arguments

data.m	Data matrix with features along the rows.
kurt.range	Desired range of kurtosis values.
cluster	Logical, to indicate whether additional cluster learning step is desired.
method	Character specifying model selection to be used (bic=EM-algorithm + BIC, vb=variational Bayesian + evidence bound).

Value

A list with the following components:

out	Matrix with rows labeling selected features and columns labeling kurtosis, cluster size and index position in data.m
class	A list with non-null elements giving the clustering classification of the selected features.

Author(s)

Andrew Teschendorff <aet21@hutchison-mrc.cam.ac.uk>

 UseBasicPrior

Prior Function for Variational Gaussian Mixture Model

Description

This function implements an uninformative prior distribution for the cluster centers and variances, but allows the user to define prior weights for the clusters.

Usage

```
UseBasicPrior(data, weights.v)
```

Arguments

<code>data</code>	A matrix with columns representing variables and rows observations. Algorithm clusters observations.
<code>weights.v</code>	A vector of relative prior weights for the clusters.

Details

`weights.v` is a vector of length `Ncat`, the maximum number of clusters to look for.

Value

A list with following components. The first four are matrices of dimension `Ncat` x `Ndim`, `dapi` is a vector of length `Ncat`.

<code>mean</code>	the means of the cluster mean gaussian priors.
<code>varm</code>	the inverse variances for the cluster mean gaussian priors.
<code>ivara</code>	parameters for the gamma prior distribution of the inverse variances of the clusters. See references.
<code>ivarb</code>	parameters for the gamma prior distribution of the inverse variances of the clusters. See references.
<code>dapi</code>	weight vector specifying prior knowledge about the number of clusters.

Author(s)

Andrew Teschendorff<act21@hutchison-mrc.cam.ac.uk>

References

- 1 D.J.MacKay: Developments in probabilistic modelling with neural networks-ensemble learning. In Neural Networks: Artificial Intelligence and Industrial Applications. Proceedings of the 3rd Annual Symposium on Neural Networks Nijmegen, Netherlands, Berlin Springer, 191-198 (1995).
- 2 J.W.Miskin : Ensemble Learning for Independent Component Analysis, PhD thesis University of Cambridge December 2000.
- 3 A. E. Teschendorff,...et al.: A variational bayesian mixture modelling framework for cluster analysis of gene expression data. Submitted to Bioinformatics.

Description

Learns a gaussian mixture model from data using an optimal separable approximation to the posterior density. The optimisation uses a variational procedure and implements an iterative ensemble learning algorithm. The algorithm gives a framework in which to infer the number of clusters in the data set. Prior information may be incorporated through specification of hyperparameters in a prior distribution. Current version implements a gaussian mixture model where the covariances matrices are diagonal.

Arguments

<code>data</code>	A matrix of dimension $N_s \times N_{dim}$ containing the data to be clustered. Algorithm clusters rows of matrix and treats columns as dimensions.
<code>prior</code>	A list of various elements containing prior information as obtained for example by using <code>UseBasicPrior</code> . List elements are <code>prior\$mean</code> , <code>prior\$ivarm</code> , <code>prior\$ivara</code> , <code>prior\$ivarb</code> and <code>prior\$dapi</code> . The first four are matrices of dimension $N_{cat} \times N_{dim}$, <code>prior\$dapi</code> is a vector of length N_{cat} . <code>prior\$mean</code> contains the means of the cluster mean gaussian priors. <code>prior\$ivarm</code> contains the inverse variances for the cluster mean gaussian priors. <code>prior\$ivara</code> and <code>prior\$ivarb</code> contain the parameters for the gamma prior distribution of the inverse variances of the clusters. <code>prior\$dapi</code> is a weight vector specifying prior knowledge about the number of clusters. If <code>prior</code> is unspecified a complete uninformative prior is implemented that assumes rows to be mean normalised to zero.
<code>Ncat</code>	The maximum number of clusters or categories to look for in the data set. Algorithm switches off clusters it doesn't need. See References.
<code>nruns</code>	Number of ensemble learning optimisation runs to be performed. Each optimisation run uses a different (random) starting point.
<code>npick</code>	The <code>npick</code> runs (out of <code>nruns</code>) that best optimise the cost function. See References.
<code>MaxIt</code>	Maximum number of iterations to be performed for a single optimisation run.
<code>conv.tol</code>	Threshold tolerance level for establishing convergence of iterations.
<code>nCV</code>	Number of consecutive iterations to consider in establishing convergence of the run at level <code>conv.tol</code> .
<code>verbatim</code>	Logical. If true prints out estimates and cost function value per iteration.

Value

A list with the following components:

`estvals` A list with components:

mean	Means of gaussian posterior. Matrix of dimension Ncat x Ndim. A row containing all zeros means that component is absent.
ivarm	Inverse variances of gaussian posterior. Matrix of dimension Ncat x Ndim.
ivara,ivarb	Parameters of gamma posterior. Matrices of dimension Ncat x Ndim.
dapi	Parameters of dirichlet posterior giving weights of components. A value of 1 means that component is absent.
wcl	A matrix of dimension npick x Ns. Each row gives cluster assignment of each row of data. Clusters are labeled by integers.
probs	A list of length npick, each list element is a matrix of dimension Ns x Ncat containing the probabilities of membership to clusters.
costs	A vector of length nruns specifying converged values of cost function.
conv	A binary vector of length nruns specifying if that run converged (0) or not (1).

Author(s)

Andrew Teschendorff{aet21@hutchison-mrc.cam.ac.uk}

References

- 1 D.J.MacKay: Developments in probabilistic modelling with neural networks-ensemble learning. In Neural Networks: Artificial Intelligence and Industrial Applications. Proceedings of the 3rd Annual Symposium on Neural Networks Nijmegen, Netherlands, Berlin Springer, 191-198 (1995).
- 2 J.W.Miskin : Ensemble Learning for Independent Component Analysis, PhD thesis University of Cambridge December 2000.
- 3 A. E. Teschendorff,...et al.: A variational bayesian mixture modelling framework for cluster analysis of gene expression data. Submitted to Bioinformatics.

Examples

```

NsTot <- 100;
Nspg <- 50;
Ng <- 2;
deg.idx <- 1 ;
data <- matrix( nrow=NsTot, ncol=Ng);
for( s in 1:Nspg ){
  data[s,] <- rnorm(Ng,0,0.25);
}
for( s in (Nspg+1):NsTot){
  data[s,] <- rnorm(Ng,0,0.25);
  data[s,deg.idx] <- rnorm(1,2,0.25);
}
types.idx <- c(rep(1,50),rep(2,50));
useprior.l <- UseBasicPrior(data,rep(1,4));
vbmix <- vabayelMix(data, prior=NA, Ncat=4, nruns=10, npick=2,MaxIt=500, conv.tol=0.001, nCV=10)
# or could use
# vbmix <- vabayelMix(data, prior=useprior.l, Ncat=4, nruns=10, npick=2,MaxIt=500, conv.tol=0.001, nCV=10)
plot(1:NsTot,vbmix$wcl[1,],type="h",col=types.idx);

```

Index

*Topic **cluster**

pack, [1](#)

UseBasicPrior, [2](#)

vabayelMix, [3](#)

pack, [1](#)

UseBasicPrior, [2](#)

vabayelMix, [3](#)