BHC

April 20, 2011

`BHC-package` *Bayesian Hierarchical Clustering*

Description

The BHC method performs bottom-up hierarchical clustering, using a Dirichlet Process (infinite mixture) to model uncertainty in the data and Bayesian model selection to decide at each step which clusters to merge. This avoids several limitations of traditional methods, for example how many clusters there should be and how to choose a principled distance metric. This implementation accepts multinomial (i.e. discrete, with 2+ categories) data.

Details

Package:	BHC
Type:	Package
Version:	1.0
Date:	2008-04-02
License:	GPL-3

The Bayesian Hierarchical Clustering (BHC) algorithm is a black-box, accessed via the bhc() function (see help for details on how to do this). All other functions in this package are accessed via bhc() and the user should not need to access them directly.

Author(s)

Rich Savage (C++ code originally written for binomial case by Yang Xu)

Maintainer: Rich Savage <r.s.savage@warwick.ac.uk>

References

Bayesian Hierarchical Clustering, Heller + Ghahramani, Gatsby Unit Technical Report GCNU-TR 2005-002 (2005); also see shorter version in ICML-2005; *R/BHC: fast Bayesian hierarchical clustering for microarray data*, Savage et al, BMC Bioinformatics 10:242 (2009)

Examples

```
require(graphics)
require(BHC)
require(affydata)
require(gcrma)

data(Dilution)
ai        <- compute.affinities(cdfName(Dilution))
Dil.expr  <- gcrma(Dilution,affinity.info=ai,type="affinities")
testData  <- exprs(Dil.expr)
keep      <- sd(t(testData))>0
testData  <- testData[keep,]
testData  <- testData[1:100,]
geneNames <- row.names(testData)

nGenes          <- (dim(testData))[1];
nFeatures       <- (dim(testData))[2];
nFeatureValues <- 4
##NORMALISE EACH EXPERIMENT TO ZERO MEAN, UNIT VARIANCE
for (i in 1:nFeatures){
    newData      <- testData[,i]
    newData      <- (newData - mean(newData)) / sd(newData)
    testData[,i] <- newData
}
##DISCRETISE THE DATA ON A GENE-BY-GENE BASIS
##(defining the bins by equal quartiles)
for (i in 1:nGenes){
  newData      <- testData[i,]
  newData      <- rank(newData) - 1
  testData[i,] <- newData
}
##PERFORM THE CLUSTERING
hc <- bhc(testData, geneNames, nFeatureValues=nFeatureValues)
plot(hc, axes=FALSE)
```

bhc — *Function to perform Bayesian Hierarchical clustering on a 2D array of discretised (i.e. multinomial) data*

Description

The method performs bottom-up hierarchical clustering, using a Dirichlet Process (infinite mixture) to model uncertainty in the data and Bayesian model selection to decide at each step which clusters to merge. This avoids several limitations of traditional methods, for example how many clusters there should be and how to choose a principled distance metric. This implementation accepts multinomial (i.e. discrete, with 2+ categories) data.

Usage

```
bhc(data, itemLabels, nFeatureValues, verbose = FALSE)
```

Arguments

`data` A 2D array containing discretised data, with values in the range 0 =< value =< nFeatureValues-1. The values start at zero because the analysis is performed in linked C++ code (which starts counting at zero). The dimensions of `data` should be nFeatures * nDataItems, and the algorithm will cluster the dataItems `itemLabels` A character array containing 'nDataItems' entries, one for each data item in the analysis. The leaf nodes of the output dendrogram will be labeled with these labels.

`nFeatureValues` The number of feature values in the (discretised) data, running from zero to nFeatureValues-1. Note that nFeatureValues=2 corresponds to binary data

`verbose` Logical. If set to TRUE, the algorithm will output some information to screen as it runs.

Value

a DENDROGRAM object (see the R stats package for details).

Author(s)

Rich Savage (C++ code originally written for binomial case by Yang Xu)

References

Bayesian Hierarchical Clustering, Heller + Ghahramani, Gatsby Unit Technical Report GCNU-TR 2005-002 (2005); also see shorter version in ICML-2005;*R/BHC: fast Bayesian hierarchical clustering for microarray data*, Savage et al, BMC Bioinformatics 10:242 (2009)

Examples

```
require(graphics)
require(BHC)
require(affydata)
require(gcrma)

data(Dilution)
ai        <- compute.affinities(cdfName(Dilution))
Dil.expr  <- gcrma(Dilution,affinity.info=ai,type="affinities")
testData  <- exprs(Dil.expr)
keep      <- sd(t(testData))>0
testData  <- testData[keep,]
testData  <- testData[1:100,]
geneNames <- row.names(testData)

nGenes         <- (dim(testData))[1];
nFeatures      <- (dim(testData))[2];
nFeatureValues <- 4
##NORMALISE EACH EXPERIMENT TO ZERO MEAN, UNIT VARIANCE
for (i in 1:nFeatures){
    newData      <- testData[,i]
    newData      <- (newData - mean(newData)) / sd(newData)
    testData[,i] <- newData
}
##DISCRETISE THE DATA ON A GENE-BY-GENE BASIS
##(defining the bins by equal quartiles)
for (i in 1:nGenes){
  newData      <- testData[i,]
  newData      <- rank(newData) - 1
  testData[i,] <- newData
}
##PERFORM THE CLUSTERING
```

```
hc <- bhc(testData, geneNames, nFeatureValues=nFeatureValues)
plot(hc, axes=FALSE)
##OUTPUT CLUSTER LABELS TO FILE
WriteOutClusterLabels(hc, "labels.txt", verbose=TRUE)
```

Index