BHC

March 24, 2012

`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) or time-series data.

Usage

```
bhc(data, itemLabels, nFeatureValues, timePoints, dataType,
    noise, numReps, noiseMode, robust, numThreads, verbose)
```

Arguments

`data` A 2D array containing discretised data. The dimensions of `data` should be `nDataItems * nFeatures`, and the algorithm will cluster the data items.

`itemLabels` A character array containing `nDataItems` entries, one for each data item in the analysis. The leaf nodes of the output dendrogram will be labelled with these labels.

`nFeatureValues`
Deprecated. This is a legacy argument, retained for backwards compatibility. Any value passed to it will have no effect.

`timePoints` An array of length `nFeatures`, containing the time points of the measurements.

`dataType` A string specifying the data type. Either "`multinomial`", "`time-course`", or "`cubicspline`".

`noise` Noise term for each gene, required only if `noiseMode=1`. The noise term for each gene is calculated as

$$\frac{\sum(\text{residuals}^2)}{(\text{number of observations for gene} - 1)(\text{number of replicates})},$$

where (number of observations for gene) is typically (number of time points * number of replicates).

numReps	Number of replicates per observation.
noiseMode	Noise mode. If 0 then fitted noise; 1 fixed noise; 2 estimated noise from replicates.
robust	0 to use single Gaussian likelihood, 1 to use mixture likelihood.
numThreads	The BHC library has been parallelised using OpenMP (currently on UN*X systems only). Specify here the number of threads to use (the default value is 1).
verbose	If set to TRUE, the algorithm will output some information to screen as it runs.

Details

Typical usage for the multinomial case:

```
bhc(data, itemLabels).
```

To use the squared-exponential covariance:

```
bhc(data, itemLabels, 0, timePoints, "time-course",
    noise, numReps, noiseMode),
```

and the cubic spline covariance:

```
bhc(data, itemLabels, 0, timePoints, "cubicspline",
    noise, numReps, noiseMode).
```

Value

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

Author(s)

Rich Savage, Emma Cooke, and Robert Darkins (binomial case originally written 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);

Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements, Cooke et al, currently under review

See Also

hclust

Examples

```
##BUILD SAMPLE DATA AND LABELS
data         <- matrix(0,15,10)
itemLabels   <- vector("character",15)
data[1:5,]   <- 1 ; itemLabels[1:5]   <- "a"
data[6:10,]  <- 2 ; itemLabels[6:10]  <- "b"
data[11:15,] <- 3 ; itemLabels[11:15] <- "c"
timePoints   <- 1:10 # for the time-course case

##DATA DIMENSIONS
nDataItems <- nrow(data)
nFeatures  <- ncol(data)

##RUN MULTINOMIAL CLUSTERING
hc1 <- bhc(data, itemLabels, verbose=TRUE)
plot(hc1, axes=FALSE)

##RUN TIME-COURSE CLUSTERING
hc2 <- bhc(data, itemLabels, 0, timePoints, "time-course",
          numReps=1, noiseMode=0, numThreads=2, verbose=TRUE)
plot(hc2, axes=FALSE)

##OUTPUT CLUSTER LABELS TO FILE
WriteOutClusterLabels(hc1, "labels.txt", verbose=TRUE)

##FOR THE MULTINOMIAL CASE, THE DATA CAN BE DISCRETISED
newData      <- data[] + rnorm(150, 0, 0.1);
percentiles  <- FindOptimalBinning(newData, itemLabels, transposeData=TRUE, verbose=TRUE)
discreteData <- DiscretiseData(t(newData), percentiles=percentiles)
discreteData <- t(discreteData)
hc3          <- bhc(discreteData, itemLabels, verbose=TRUE)
plot(hc3, axes=FALSE)
```

Index

∗Topic **cluster**
bhc, 1

bhc, 1

DiscretiseData *(bhc)*, 1

FindOptimalBinning *(bhc)*, 1

hclust, *2*

WriteOutClusterLabels *(bhc)*, 1