Journal of Statistical Software

April 2012, Volume 47, Issue 5. http://www.jstatsoft.org/

High-Dimensional Bayesian Clustering with Variable Selection: The R Package bclust

Vahid Partovi Nia
Ecole Polytechnique de Montréal

Anthony C. Davison
Ecole Polytechnique Fédérale de Lausanne

Abstract

The R package **bclust** is useful for clustering high-dimensional continuous data. The package uses a parametric spike-and-slab Bayesian model to downweight the effect of noise variables and to quantify the importance of each variable in agglomerative clustering. We take advantage of the existence of closed-form marginal distributions to estimate the model hyper-parameters using empirical Bayes, thereby yielding a fully automatic method. We discuss computational problems arising in implementation of the procedure and illustrate the usefulness of the package through examples.

Keywords: agglomerative clustering, Bayesian clustering, Bayesian variable selection, dendrogram, hierarchical clustering, R, spike-and-slab model.

1. Introduction

The purpose of cluster analysis is to partition observations into groups such that observations belonging to the same group are more similar than observations belonging to different groups. There are various ways of attributing observations to clusters, but one may classify them into two broad categories: distance-based (nonparametric) and model-based (parametric) techniques. Our approach lies between these: we use a model to define a distance and we implement hierarchical clustering as used in distance-based methods.

Hierarchical clustering using various distance measures is implemented in the R progamming language (R Development Core Team 2012) in packages such as **cluster** (Mächler *et al.* 2012) and has two variants, agglomerative clustering and divisive clustering, implemented in the `agnes` and `diana` functions of this package respectively. Agglomerative clustering begins with each observation as a separate cluster, successively merges the closest clusters using a dissimilarity measure, and stops when there is just one cluster. Divisive clustering starts with all the observations in one cluster and divides it until each observation forms a single

cluster. However, hierarchical clustering is not the only way of grouping data. Another widely used technique is partitioning clustering, as embodied in the k-means algorithm, `kmeans`, of the package **stats**. A more robust variant, k-medoids, is coded in the `pam` function in the package **cluster**. Unfortunately, partitioning approaches can be hard to visualize, though some graphical tools are available in the packages **flexclust** (Leisch 2010) and **cclust** (Dimitriadou 2009). The dendrogram, a tree representation that provides a visual guide to the groupings as the number of clusters changes, is usually unavailable in partitioning algorithms. Many graphical tools are provided in the **ape** package (Paradis *et al.* 2004). One partitioning method is Bayesian mixture modeling, which often requires Markov chain Monte Carlo simulation, an example being the finite Gaussian mixtures of the **bayesm** package (Rossi 2011). Partial tree representation of Markov chain Monte Carlo groupings is feasible through **labeltodendro** package (Partovi Nia and Stephens 2012).

In many scientific domains modern technology provides data on many more variables than individuals. Cluster analysis is widely used in such cases, and a common difficulty is to provide reasonable statistical models for these low-sample-size-high-dimensional situations. Statistical analysis of such data is difficult partly because of overfitting, for which two main solutions have been proposed: the data are projected to a smaller dimension, or analysis is based only on relevant variables. Our approach is related to the second solution.

In the high-dimensional datasets now arising in biological applications, the key information on clustering may be hidden in a small subset of the variables (Cheeseman and Stutz 1996), and inclusion of other variables may mask the underlying structure. Many model-based clustering procedures depend on ratios of probability densities. When the data dimension greatly exceeds the number of individuals, the probability that two individuals will lie close enough to be considered part of the same cluster approaches zero, if substantial variation occur across all variables (Hall *et al.* 2005; Ahn *et al.* 2007). Thus variable selection or projection into a subspace seems necessary when clustering high-dimensional datasets, and this complicates matters further.

A variable may be considered useful for clustering if it defines a mixture, so variable selection in clustering requires the fitting of mixtures with unknown numbers of components on an unknown number of variables (Kim *et al.* 2006). Researchers have dealt with this in different ways. McLachlan *et al.* (2002) apply forward selection of variables using univariate significance tests of a single component against mixtures of two components. Wang and Zhu (2008) and Bondell and Reich (2008) implement variable selection using a penalized likelihood. Friedman and Meulman (2004) assign different weights to each variable as a measure of its importance, and have implemented this in the **COSA** software. Witten and Tibshirani (2010) similarly perform variable selection and provide importance measures by penalization of the dissimilarity matrix, implemented in the **sparcl** R package (Witten and Tibshirani 2011). Bergé *et al.* (2012) implemented clustering and discriminant analysis of high-dimensional data in the **HDclassif** R package using a new parametrization of the Gaussian mixture model which combines the idea of dimension reduction and model constraints on the covariance matrices. Hoff (2006) and Booth *et al.* (2008) suggest stochastic search to find the optimal clustering. Raftery and Dean (2006) fit a finite Gaussian mixture model and select variables using an approximate Bayes factor; see the R package **clustvarsel** (Dean and Raftery 2009). Tadesse *et al.* (2005) suggest use of a reversible jump algorithm for their Bayesian model. Another approach is dimension reduction by principal components analysis (Ghosh and Chinnaiyan 2002), but this may not show which variables are more effective for clustering or carry the

best information about the cluster topology (Chang 1983). Liu *et al.* (2003) combine principal components analysis with variable selection and propose a Gibbs sampler to determine the number of components to be used. In the present paper we use Bayesian variable selection through spike-and-slab models (Mitchell and Beauchamp 1988; George and McCulloch 1997). Our suggested model imposes independence of variables, so selection of variables marginally or conditional on the previous selected variables coincide.

Most of the model-based clustering R packages are inappropriate for high-dimensional data except for **HDclassif**. The **bclust** package version 1.3, built for R 2.15.0, is intended to fill this gap. Unlike **HDclassif** the **bclust** package implements a Bayesian approach to clustering, with priors for model parameters and for the allocation of subjects to groups. The model and its priors are chosen so that the marginal posterior is analytically tractable, providing a fast algorithm. The marginal posterior is taken as the natural measure of the appropriateness of a grouping. The clustering that maximizes the marginal posterior is taken to be optimal. Since it is not easy to find the maximum a posteriori grouping over all possible partitions, we propose an approximation. The agglomerative path is used to approximate the maximum a posteriori clustering. This gives a visual guide to some of the other possible data allocations, through a dendrogram. The R package implementing the methodology described in this work is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=bclust`.

2. Bayesian model for clustered data

Suppose that T clustering individuals are grouped into C clusters. The univariate random variable y_{vct} is the data of clustering individual t ($t = 1, \ldots, T_c$) in cluster c ($c = 1, \ldots, C$) measured on the continuous variable v ($v = 1, \ldots, V$). If there are T_c observations in cluster c ($T = \sum_{c=1}^{C} T_c$), then the data distribution is the same if the observations in cluster c are arbitrarily reordered. Thus $f(y_{1c} \ldots y_{T_c c})$ is an exchangeable distribution, and by the general representation theorem (Bernardo and Smith 1994, Chapter 4), there is a conditional distribution $f(y_c \mid \xi_c)$ and a prior distribution function $F(\xi_c)$ such that

$$f(y_{1c}, \ldots, y_{T_c c}) = \int \prod_{t=1}^{T_c} f(y_{tc} \mid \xi_c) dF(\xi_c), \tag{1}$$

suggesting use of a Bayesian model. We propose a linear model for the data:

$$y_{vct} = \mu + \delta_v \gamma_{vc} \theta_{vc} + \eta_{vct}, \tag{2}$$

where η_{vc} and θ_{vc} are continuous random variables, γ_{vc} and δ_v are binary random variables, and μ is a constant. The random variable η_{vct} is noise, supposed to be independent of θ_{vc} and sampled from a Gaussian distribution with zero mean and variance $\sigma_\eta^2 \geq 0$. The random variable θ_{vc} disappears when $\gamma_{vc} = 0$ or $\delta_v = 0$ but is present when $\gamma_{vc} = \delta_v = 1$. We assume that γ_{vc} and δ_v are independent of each other and follow Bernoulli distributions with probabilities p and q, respectively.

In Equation 2, μ represents an overall value for all the variables and individuals. Without loss of generality, our model presupposes that all the variable-wise centers equal zero; thus we suggest subtracting the median of each variable before using our software. The random

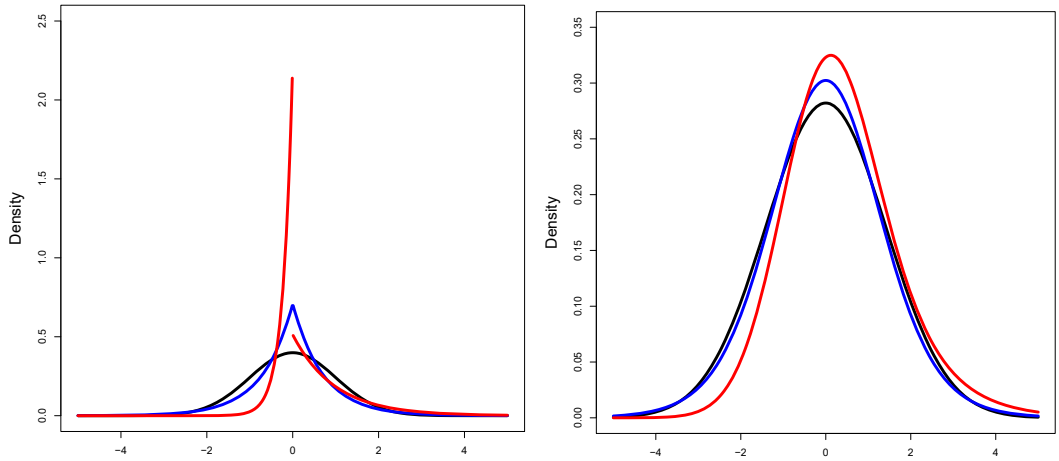

Figure 1: Distributions of underlying effects and of measurements. Left: the standard Gaussian density (black), the symmetric Laplace density (blue), and the asymmetric Laplace density with $\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 10$ (red), all having zero median and unit variance, giving examples of the distribution of θ_{vc}. Right: the marginal density of a measurement y_{vct} when the variable and the variable-class combination is active, obtained by convolving a standard Gaussian density with the densities on the left.

variables θ_{vc} are the cluster effects, and give different means on variable v for different clusters. If θ_{vc} appears for at least one cluster, then the variable v is important, and the importance of variables for clustering is coded in the Bernoulli random variable δ_v. One may interpret the probabilities q and p as the proportions of important variables and of the appearance of different cluster means for an important variable.

We propose two families for θ_{vc}: a Gaussian distribution with mean zero and variance $\sigma^2_\theta > 0$; and an asymmetric Laplace distribution (Bhowmick *et al.* 2006), used to model heavy tailed and asymmetric effects, centered at zero, and with left-tail and right-tail variances $\sigma^2_{\theta_L} > 0$ and $\sigma^2_{\theta_R} > 0$; see Figure 1. Both the Gaussian and the asymmetric Laplace families produce closed form marginal densities for the observations, yielding a fast algorithm. The slab density of y_{vct} in (2) when the effects follow the asymmetric Laplace distribution is plotted in Figures 1 and 2, and has the form

$$\begin{aligned} f_1(y_{vct}) \quad = \quad & \frac{1}{2\sigma_{\theta_L}} \exp\left(\frac{y_{vct}}{\sigma_{\theta_L}} + \frac{\sigma^2_\eta}{2\sigma^2_{\theta_L}}\right) \Phi\left(-\frac{y_{vct}}{\sigma_\eta} - \frac{\sigma_\eta}{\sigma_{\theta_L}}\right) \\ & + \frac{1}{2\sigma_{\theta_R}} \exp\left(-\frac{y_{vct}}{\sigma_{\theta_R}} + \frac{\sigma^2_\eta}{2\sigma^2_{\theta_R}}\right) \Phi\left(\frac{y_{vct}}{\sigma_\eta} - \frac{\sigma_\eta}{\sigma_{\theta_R}}\right). \end{aligned}$$

In Bayesian variable selection the term spike-and-slab distribution is typically used for the prior distribution. We use the terms spike for a distribution which is concentrated about zero and slab for the distribution with tails much more dispersed than the spike density, whether it is a prior or a marginal density.

If $\delta_v = \gamma_{vc} = 1$, then the variable v and the variable-cluster combination v, c are active. In this case the marginal variance of data in variable-cluster combination v, c equals $\sigma^2_\theta + \sigma^2_\eta$, defining a slab distribution, otherwise the marginal variance equals σ^2_η, giving a spike distribution, see

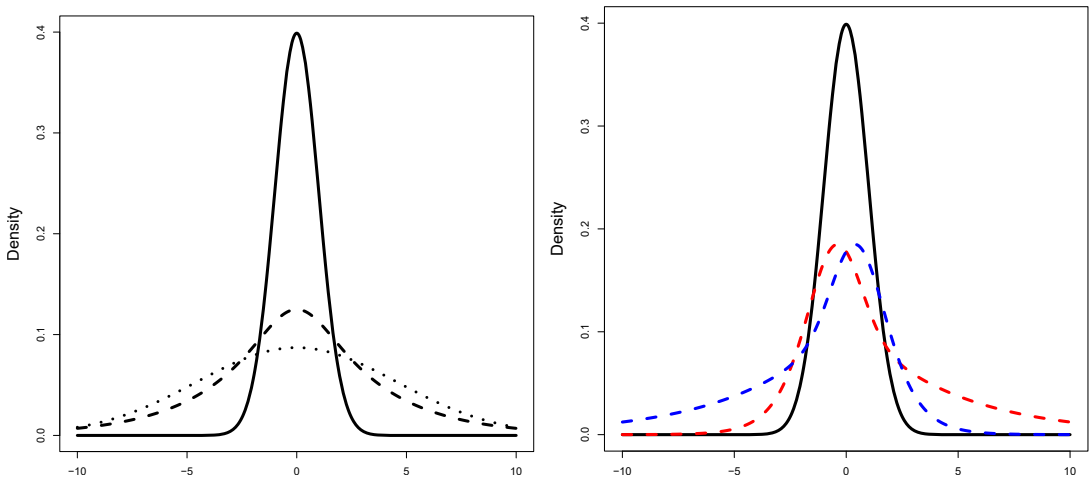

Figure 2: Marginal densities of measurements. Left: examples of a Gaussian spike (solid) and slab (dotted and dashed) densities; the dotted density is obtained by adding a Gaussian effect to a Gaussian noise variable; the dashed density is derived by adding a symmetric Laplace effect to Gaussian noise. Right: Gaussian spike and right-skewed (red dashed) and left-skewed (blue dashed) densities obtained by adding an asymmetric Laplace effect to Gaussian noise.

Figure 2. Hence model (2) always gives a Gaussian spike distribution, but depending on the distribution of θ_{vc}, it provides a symmetric or an asymmetric slab distribution.

Sometimes clustering individuals include replicate data. One may omit the replication information and consider each replicate as a clustering individual, but after the data are grouped some of the replicates may then fall into different groups, and this is undesirable. We therefore propose to generalize the model (2) by assuming another level of variability between replicates of a clustering individual. If there are R_{ct} replicates of clustering individual t grouped in cluster c, then we propose to generalize (2) to

$$y_{vctr} = \mu + \delta_v \gamma_{vc} \theta_{vc} + \eta_{vct} + \varepsilon_{vctr}, \quad r = 1, \ldots, R_{ct}. \tag{3}$$

This reduces to (2) if $R_{vc} = 1$ for all $v = 1, \ldots, V$ and $c = 1, \ldots, C$, so below we focus on (3). Comparing (3) with (2), we see that if $R_{vc} = 1$ for all $c = 1, \ldots, C, v = 1, \ldots, V$, then the model is identifiable only with respect to $\sigma_\eta^2 + \sigma_\varepsilon^2$. This is important when marginal maximum likelihood is used to estimate the model hyper-parameters for unreplicated data using the replicated model (3). In such cases we set $\sigma_\eta^2 = 0$ and estimate σ_θ^2.

Suppose that the letter y with fewer indices corresponds to an appropriate vector of data. For instance, y_v denotes the data vector for variable v and y_{vc} corresponds to the data vector for variable v and cluster c. The marginal density of the Bayesian model (3) can be obtained by replacing

$$\xi_c = (\theta_{1c}, \theta_{vc}, \ldots, \theta_{Vc}, \eta_{1c1}, \ldots, \eta_{1cT_c}, \ldots \eta_{vct}, \ldots, \eta_{VcT_c}), \tag{4}$$

in (1) and evaluating the integral. The marginal density equals (Partovi Nia 2009)

$$f(y) = \prod_{v=1}^{V} f(y_v), \tag{5}$$

in which the marginal density for each variable is a convex combination of the spike-and-slab densities, given by

$$\begin{aligned} f(y_v) &= qf(y_v \mid \delta_v = 1) + (1-q)f(y_v \mid \delta_v = 0), \\ f(y_v \mid \delta_v = 0) &= \prod_{c=1}^{C}\prod_{t=1}^{T_c} f_0(y_{vct}), \\ f(y_v \mid \delta_v = 1) &= \prod_{c=1}^{C}\left\{ pf_1(y_{vc}) + (1-p)\prod_{t=1}^{T_c} f_0(y_{vct}) \right\}, \end{aligned} \tag{6}$$

where

$$\begin{aligned} f_0(y_{vct}) &= (2\pi)^{-R_{ct}/2}\sigma^{1-R_{ct}}(R_{ct}\sigma_\eta^2 + \sigma_\varepsilon^2)^{-1/2} \\ &\quad \times \exp\left\{ -\frac{1}{2\sigma_\varepsilon^2}\left(\sum_{r=1}^{R_{ct}} y_{vctr}^2 - R_{ct}\overline{y}_{vct}^2 \right) - \frac{(\overline{y}_{vct} - \mu)^2}{2(\sigma_\eta^2 + \sigma_\varepsilon^2/R_{ct})} \right\}, \end{aligned} \tag{7}$$

is independent of the distribution assumed for the cluster effects θ_{vc}, but f_1 depends on their distribution. If the effect has a Gaussian distribution, then f_1 corresponds to a multivariate Gaussian density with mean vector $\mu\mathbf{1}$ and covariance matrix $\mathbf{\Sigma}$, where $\mathbf{\Sigma}$ is of dimension $\sum_{t=1}^{T_c} R_{ct} \times \sum_{t=1}^{T_c} R_{ct}$ with $\sigma_\varepsilon^2 + \sigma_\eta^2 + \sigma_\theta^2$ on the main diagonals, and the off-diagonal elements are equal to $\sigma_\eta^2 + \sigma_\theta^2$ for replications of the same individual and to σ_θ^2 for observations from different individuals.

When the effects are distributed according to the asymmetric Laplace distribution with variance $\sigma_\theta^2 = \sigma_{\theta_L}^2 + \sigma_{\theta_R}^2$, the rates of the left- and right-tail exponential distributions forming the Laplace density being $\sigma_{\theta_L}^{-1}$ and $\sigma_{\theta_R}^{-1}$, then

$$f_1(y_{vc}) = k_0(k_L I_L + k_R I_R), \tag{8}$$

where

$$\begin{aligned} k_0 &= (2\pi\sigma_\varepsilon^2)^{-\sum_{t=1}^{T_c} R_{ct}/2}(2\pi\sigma_\eta^2)^{-T_c/2}(2\pi\sigma_\eta^2/T_c)^{1/2} \\ &\quad \times (2\pi)^{T_c/2}|\mathbf{A}|^{-1/2}\exp\left\{ -\frac{1}{2\sigma_\varepsilon^2}\sum_{r=1}^{R_{ct}}\sum_{t=1}^{T_c} y_{vctr}^2 \right\}, \\ k_L &= (2\sigma_{\theta_L})^{-1}\exp\left(\frac{\sigma_\eta^2}{2T_c\sigma_{\theta_L}^2} - \frac{\mu}{\sigma_{\theta_L}} \right), \\ I_L &= \exp\left(\frac{1}{2}\mathbf{b}_L^\top \mathbf{A}^{-1}\mathbf{b}_L \right)\Phi\left(\frac{c_L + \mathbf{d}_L^\top \mathbf{A}^{-1}\mathbf{b}_L}{\sqrt{1 + \mathbf{d}_L^\top \mathbf{A}^{-1}\mathbf{d}_L}} \right), \\ k_R &= (2\sigma_{\theta_R})^{-1}\exp\left(\frac{\sigma_\eta^2}{2T_c\sigma_{\theta_R}^2} + \frac{\mu}{\sigma_{\theta_R}} \right), \\ I_R &= \exp\left(\frac{1}{2}\mathbf{b}_R^\top \mathbf{A}^{-1}\mathbf{b}_R \right)\Phi\left(\frac{c_R + \mathbf{d}_R^\top \mathbf{A}^{-1}\mathbf{b}_R}{\sqrt{1 + \mathbf{d}_R^\top \mathbf{A}^{-1}\mathbf{d}_R}} \right). \end{aligned}$$

Here $\mathbf{d}_L, \mathbf{d}_R, \mathbf{b}_L, \mathbf{b}_R$ are all vectors of length T_c. The vectors $\mathbf{d}_L$ and $\mathbf{d}_R$ with equal elements $-T_c^{-1/2}\sigma_\eta^{-1}$ and $T_c^{-1/2}\sigma_\eta^{-1}$, the vectors $\mathbf{b}_L$ and $\mathbf{b}_R$ consisting of elements $R_{ct}\overline{y}_{vct}\sigma^{-2}+T_c^{-1}\sigma_{\theta_L}^{-1}$ and $R_{ct}\overline{y}_{vct}\sigma^{-2}-T_c^{-1}\sigma_{\theta_R}^{-1}$, respectively. The constants c_L and c_R are

$$c_L = T_c^{1/2}\sigma_\eta^{-1}\{\mu - \sigma_\eta^2/(T_c\sigma_{\theta_L})\}, \quad c_R = -T_c^{1/2}\sigma_\eta^{-1}\{\mu + \sigma_\eta^2/(T_c\sigma_{\theta_R})\}.$$

The square matrix $\mathbf{A}_{T_c\times T_c}$ is positive definite with determinant $|\mathbf{A}|$ consisting of main diagonals $R_{ct}\sigma_\varepsilon^{-2} + \sigma_\eta^{-2} - T_c^{-1}\sigma_\eta^{-2}$ and equal off-diagonals $-T_c^{-1}\sigma_\eta^{-2}$.

An immediate consequence of (6) is resistance of the clustering method to the noise variables, because the data density has two parts. The first is the density of data when the clustering parameter θ_{vc} appears, the so-called slab density f_1. This guides the clustering procedure when a variable is important for clustering. The second, the spike density f_0, is the density of the data when the clustering parameter θ_{vc} disappears. This part down-weights the effect of useless variables in clustering and provides a valid clustering procedure when the number of noise variables increases. In the extreme case if the data density consists only of f_0s, that is with probability one $\delta_v = 0$ for all $v = 1, \ldots, V$, or equivalently $\gamma_{vc} = 0$ for all v and $c = 1, \ldots, C$, the data play no role in grouping and the clustering posterior equals the prior.

We take the log Bayes factors $\log B_\delta = \log f(y_v \mid \delta_v = 1) - \log f(y_v \mid \delta_v = 0)$ as a measure of the importance of the variable v and $\log B_\gamma = \log f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) - \log f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 0)$ as a measure of importance for the variable-cluster combination v, c. The posterior odds is a Bayesian measure of uncertainty for testing two hypotheses and equals the prior odds times the Bayes factor. The posterior odds are only related to data through the Bayes factor and are understood as a data-based measure of the evidence when comparing two hypotheses (Kass and Raftery 1995).

3. Bayesian clustering paradigm

In Bayesian clustering, the allocation of observations to clusters is regarded as a statistical parameter. Therefore a Bayesian model such as (3) is assumed for data conditional on the grouping structure, and a prior distribution must be adopted for the clusters. Then a search algorithm, often using Markov chain Monte Carlo simulation, is applied to find the maximum a posteriori grouping. In the **bclust** package, similar to **HBC** (Savage *et al.* 2009), a **Bioconductor** (Gentleman *et al.* 2004) package, we use an agglomerative search method because it provides a visual guide to the other possible groupings through a dendrogram.

Suppose a data allocation $\mathcal{C}$ groups the observations into C clusters, of sizes $T_1, \ldots, T_C$, with total $T = \sum_{c=1}^{C} T_c$ clustering individuals. We assume a multinomial-Dirichlet distribution (Heard *et al.* 2006) as the allocation prior

$$f(\mathcal{C}) \propto \frac{(C-1)!T_1!\ldots T_C!}{T(T+C-1)!}. \tag{9}$$

The clustering posterior is

$$f(\mathcal{C} \mid y) = k^{-1} f(y \mid \mathcal{C}) f(\mathcal{C}), \tag{10}$$

in which $f(y \mid \mathcal{C})$ is the marginal density of the data for the known allocation $\mathcal{C}$ derived in (5), and $k > 0$ is a fixed value for given data. The normalizing constant k plays no role in agglomerative clustering and may be omitted in numerical calculations.

Function	Description
`bclust`	Bayesian agglomerative clustering using the spike-and-slab model.
`bdiscrim`	Discriminates using the spike-and-slab model.
`ditplot`	Visualizes data using `image` plot.
`dptplot`	Visualizes data using `profileplot`.
`imp`	Calculates variable and variable-cluster importances.
`loglikelihood`	Computes the marginal log likelihood for the spike-and-slab model.
`meancss`	Computes the mean and the corrected sum of squares for `loglikelihood`.
`profileplot`	Visualizes replicated data.
`teethplot`	Visualizes grouping on `image` or `profileplot`.
`viplot`	Visualizes variable importances.

Table 1: Summary of the functions in the **bclust** package.

In order to apply Bayesian agglomerative clustering we start with each individual as a single cluster: the number of clusters equals the total number of individuals, $C = T$, and the number of individuals in cluster c is $T_c = 1$, for all $c = 1, \ldots, C$. In the first step, all pairwise merges are considered. For each pairwise merge, the clustering posterior (10) is calculated and the merge that maximizes (10) is applied. We keep $g_c = \log f(\mathcal{C} \mid y)$, the log posterior for the best merge having c clusters, to use as the dendrogram height. If the best merge according to (10) is to join cluster c_1 to c_2 to create the new cluster c, then of course $T_c = T_{c_1} + T_{c_2}$. The algorithm then considers all pairwise merges again, and continues until all clusters are merged and all individuals are in one cluster.

The best grouping found using the posterior as the objective function on the agglomerative path is the one that maximizes g_c across $c = 1, \ldots, T$. Clearly the groupings associated to g_c are sorted in agglomerative order with increasing c, so a dendrogram representation is possible. In order to draw a dendrogram a monotone height function is required, but g_c is not necessarily monotone and we use the following transformation. Write $g_{\max} = \max(g_c)$, and suppose that $c_{\max} = \text{argmax}(g_c)$ is the number of clusters that maximises g_c. For $c < c_{\max}$ we define the height of the dendrogram to be $h_c = g_c - g_{\max}$, which is negative, and for $c > c_{\max}$, we take $h_c = g_{\max} - g_c$, which is positive. By definition, h_c is monotone if g_c is unimodal, which is usually the case, and cutting the dendrogram at zero height gives the grouping that maximizes g_c. However plotting a dendrogram object in R requires non-negative heights, so we replace h_c by $h_c - \min(h_c)$.

4. Computational issues in code implementation

In order to accelerate the numerical computations, most of the **bclust** package is written in standard C and output is imported into R to benefit from its visualization facilities. However, some of the required routines were already available in Fortran, and are called using the `F77_CALL` function of the **BLAS** C library. Table 1 summarizes the main functions in the package **bclust** and Figure 3 denotes their dependencies.

The main difficulty of agglomerative clustering is fast evaluation of the data joint density $f(y \mid \mathcal{C})$. When the number of clusters is C, $C(C-1)/2$ merges are considered and because C varies from 1 to T, the total number of evaluations is $\sum_{C=1}^{T} C(C-1)/2$, which is of order

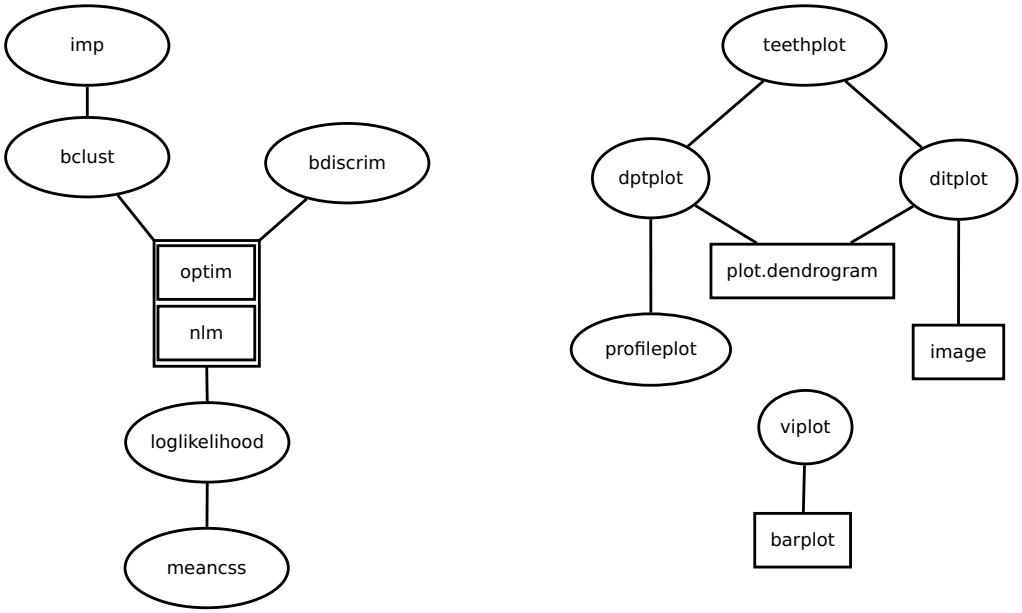

Figure 3: Diagram of function dependencies. Left panel: functions used for computations. Right panel: functions used for visualization. Ellipses denote functions developed in the **bclust** package and rectangles denote pre-existing R functions.

$O(T^3)$. This can be improved if a Lance-Williams type relationship (Lance and Williams 1967) holds for the posterior function. On the other hand, because the model (3) imposes independent variables, $f(y \mid \mathcal{C})$ reduces to $\prod_{v=1}^{V} f(y_v)$ and hence agglomerative clustering is of order $O(VT^3)$, linear in the number of variables V. This is encouraging because in high dimensional settings T is small but V is large, so our algorithm is rather fast. However, evaluation of $f(y \mid \mathcal{C})$ may be time-consuming for large V or T, and computational acceleration is then required.

In order to decide which clusters must be merged, we need to evaluate individual densities for each variable (6). The density evaluation becomes computationally expensive if C is large, as in the early stages of agglomerative clustering. A simple trick to rapidly evaluate $f(y_v \mid \mathcal{C})$ is to use the fact that in the agglomerative method only two clusters will be joined, so the evaluation of the density of two clusters with the past values of $f(y_{vc} \mid \mathcal{C})$ suffices for evaluation of the new $f(y \mid \mathcal{C})$. Every time that we evaluate $f(y \mid \mathcal{C})$, only the joint density of the merging clusters is calculated and $f(y \mid \mathcal{C})$ is reconstructed by multiplying the lacking components.

The individual density $f(y_v)$ for the Gaussian and the asymmetric Laplace model is composed of products and therefore it is best computed on the log scale. If we write

$$l_0 = \sum_{t=1}^{T_c} \log f_0(y_{vct}), \quad l_1 = \log f_1(y_{vc}), \tag{11}$$

then

$$\log f(y_v \mid \delta_v = 1) = \sum_{c=1}^{C} \log\{p \exp(l_0) + (1-p)\exp(l_1)\}.$$

When l_0 and l_1 are both very small or very large, the computation of

$$l = \sum_{c=1}^{C} \log\{p \exp(l_0) + (1-p)\exp(l_1)\} \tag{12}$$

is troublesome, and computer memory may overflow or l may be evaluated as zero. To avoid this we evaluate l after factorizing $\exp(l_1)$ as

$$l = l_1 + \log\left\{p + (1-p)\exp(l_0 - l_1)\right\}. \tag{13}$$

This expression is appropriate when $l_1 > l_0$, because the exponent function in (13) doesn't explode. There is an obvious variant when $l_0 \geq l_1$. A similar trick is applied for the evaluation of $\log f(y_v)$ using $\log f(y_v \mid \delta_v = 1)$ and $\log f(y_v \mid \delta_v = 0)$.

In the Gaussian effects model, $\log f_1(y_{vc})$ corresponds to logarithm of a d-variate Gaussian density with mean $\mu\mathbf{1}$ and covariance matrix $\mathbf{\Sigma}$, where $d = \sum_{t=1}^{T_c} R_{ct}$, $\mathbf{1}$ is a unit vector of length d and $\mathbf{\Sigma}$ is a $d \times d$ positive definite matrix, that is,

$$\log f_1(y_{vc}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}(y_{vc} - \mu\mathbf{1})^\top \mathbf{\Sigma}^{-1}(y_{vc} - \mu\mathbf{1}). \tag{14}$$

Evaluation of this density requires computation of the Mahalanobis distance and the log determinant of $\mathbf{\Sigma}$. In order to efficiently compute them, let the upper-triangular matrix $\mathbf{B}_{d\times d}$ denote the Cholesky decomposition of $\mathbf{\Sigma}$, that is $\mathbf{B}^\top\mathbf{B} = \mathbf{\Sigma}$. The Cholesky decomposition of a positive definite matrix is efficiently implemented in Fortran and is available in the function `dpbtrf` of the **LAPACK** library (Anderson *et al.* 1999). Because $\mathbf{B}$ is upper-triangular, a solution to the system of linear equations

$$\mathbf{B}\mathbf{x} = (y_{vc} - \mu\mathbf{1}) \tag{15}$$

is easily obtained by back-solving using the **LAPACK** function `dtrtrs`. Hence, $\mathbf{x} = \mathbf{\Sigma}^{-\frac{1}{2}}(y_{vc} - \mu\mathbf{1})$ might be used to evaluate the Mahalanobis distance as

$$\mathbf{x}^\top\mathbf{x} = \sum_{i=1}^{d} x_i^2 = (y_{vc} - \mu\mathbf{1})^\top \mathbf{\Sigma}^{-1}(y_{vc} - \mu\mathbf{1}), \tag{16}$$

in which x_i represents the ith element of the vector $\mathbf{x}$.

Once the Cholesky decomposition of $\mathbf{\Sigma}$ is computed, the eigenvalues λ_i are also available. Denoting the diagonal elements of $\mathbf{B}$, by b_{ii}, we have $b_{ii} = \lambda_i^{1/2}$, and hence

$$\log|\mathbf{\Sigma}| = \sum_{i=1}^{d}\log\lambda_i = 2\sum_{i=1}^{d}\log b_{ii}. \tag{17}$$

The log density can be obtained by replacing the Mahalanobis distance (16) and the log determinant (17) in (14), yielding

$$\log f_1(y_{vc}) = -\frac{d}{2}\log 2\pi - \sum_{i=1}^{d}\log b_{ii} - \frac{1}{2}\sum_{i=1}^{d} x_i^2.$$

We need to apply this procedure for all vectors of data y_{vc} ($v = 1, \ldots, V$, $c = 1, \ldots, C$). We can save computational time for data in the same cluster but another variable, say $y_{v'c}$ ($v' \neq v$), because for $y_{v'c}$, the covariance matrix $\mathbf{\Sigma}$ and hence $\mathbf{B}$ are unchanged, so we do not need to re-calculate the Cholesky decomposition of $\mathbf{\Sigma}$. However, the back-solving must be updated according to the new data in $\mathbf{Bx} = y_{v'c} - \mu\mathbf{1}$, and the Mahalanobis distance must be recomputed using the new $\mathbf{x}$.

In the asymmetric Laplace model the density $f_1(y_{vc})$ given in (8) has a more complicated form. However, the computational difficulty arises only in the calculation of

$$|\mathbf{A}|,\ \mathbf{b}_L^\top \mathbf{A}^{-1}\mathbf{b}_L,\ \mathbf{b}_R^\top \mathbf{A}^{-1}\mathbf{b}_R,\ \mathbf{d}_L^\top \mathbf{A}^{-1}\mathbf{b}_L,\ \mathbf{d}_R^\top \mathbf{A}^{-1}\mathbf{b}_R,\ \mathbf{d}_L^\top \mathbf{A}^{-1}\mathbf{d}_L,\ \mathbf{d}_R^\top \mathbf{A}^{-1}\mathbf{d}_R, \tag{18}$$

and of Φ, the standard Gaussian cumulative distribution function. The cumulative Gaussian distribution function is available in the `Rmath C` library and evaluation of the quantities in (18) is similar to the Gaussian case. First we calculate the upper-triangular Cholesky decomposition of $\mathbf{A}_{d\times d}$, say $\mathbf{B}_{d\times d}$, in which $d = T_c$. Hence

$$\log|\mathbf{A}| = 2\sum_{i=1}^{d} \log b_{ii},$$

and we find the vectors $\mathbf{x}_{\mathbf{b}_L}$, $\mathbf{x}_{\mathbf{b}_R}$, $\mathbf{x}_{\mathbf{d}_L}$, $\mathbf{x}_{\mathbf{d}_R}$ by back-solving the systems of linear equations

$$\mathbf{B}\mathbf{x}_{\mathbf{b}_L} = \mathbf{b}_L,\ \mathbf{B}\mathbf{x}_{\mathbf{d}_L} = \mathbf{d}_L,\ \mathbf{B}\mathbf{x}_{\mathbf{b}_R} = \mathbf{b}_R,\ \mathbf{B}\mathbf{x}_{\mathbf{d}_R} = \mathbf{d}_R.$$

Therefore, the required quantities are

$$\begin{aligned}
\mathbf{b}_L^\top \mathbf{A}^{-1}\mathbf{b}_L &= \mathbf{x}_{\mathbf{b}_L}^\top \mathbf{x}_{\mathbf{b}_L},\ \mathbf{b}_R^\top \mathbf{A}^{-1}\mathbf{b}_R = \mathbf{x}_{\mathbf{b}_R}^\top \mathbf{x}_{\mathbf{b}_R},\ \mathbf{d}_L^\top \mathbf{A}^{-1}\mathbf{b}_L = \mathbf{x}_{\mathbf{d}_L}^\top \mathbf{x}_{\mathbf{b}_L},\\
\mathbf{d}_R^\top \mathbf{A}^{-1}\mathbf{b}_R &= \mathbf{x}_{\mathbf{d}_R}^\top \mathbf{x}_{\mathbf{b}_R},\ \mathbf{d}_L^\top \mathbf{A}^{-1}\mathbf{d}_L = \mathbf{x}_{\mathbf{d}_L}^\top \mathbf{x}_{\mathbf{d}_L},\ \mathbf{d}_R^\top \mathbf{A}^{-1}\mathbf{d}_R = \mathbf{x}_{\mathbf{d}_R}^\top \mathbf{x}_{\mathbf{d}_R}.
\end{aligned}$$

The value of the log density is readily obtained. For data in the same cluster but a different variable, say $y_{v'c}$, the quantities $\mathbf{A}, \mathbf{d}_L$, and $\mathbf{d}_R$ are unchanged. Hence, we just need to update $\mathbf{x}_{\mathbf{b}_L}$ and $\mathbf{x}_{\mathbf{b}_R}$, replace them in (19) and can then evaluate $f_1(y_{v'c})$ with less computational effort. The positive definite matrix $\mathbf{A}$ is exchangeable and therefore $|\mathbf{A}|$ and $\mathbf{A}^{-1}$ are available analytically, but using the analytical forms doesn't accelerate the algorithm much.

5. Code analysis on simulated data

In order to analyze our computer code, a simple factorial experiment was performed with the number of variables V set to $50, 100, 200, 300, 500, 1000$ and the number of individuals T set to $10, 20, 30, 40, 50, 100, 200, 300$. The experiment was run on a desktop PC with Intel Core Duo processor 1.8 MHz, 1 GB RAM and Linux Ubuntu operating system. Each design was fitted 5 times using the Gaussian and the asymmetric Laplace models and the time in seconds required for agglomerative clustering was saved.

Least squares estimates of the parameters $(\beta_0, \beta_1, \beta_2)^\top$ for the linear model $\log_{10}\text{time} = \beta_0 + \beta_1 \log_{10} V + \beta_2 \log_{10} T$ are $(-6.62, 0.98, 3.19)^\top$ for the Gaussian model and $(-6.22, 0.98, 2.96)^\top$ for the asymmetric Laplace model. As expected, $\beta_1 \approx 1$ and $\beta_2 \approx 3$ for both models. The value of β_2 for the asymmetric Laplace model is smaller than that for the Gaussian model, suggesting a more efficient algorithm is implemented for the asymmetric Laplace model especially for

V	50			500			1000		
T	20	50	100	20	50	100	20	50	100
Gaussian	0.1	2.7	30.3	1.4	25.1	273.2	2.8	49.9	542.3
Asymmetric Laplace	0.2	2.9	24.6	1.8	26.8	215.9	3.6	53.3	428.6

Table 2: Average clustering time (in seconds) for different number of variables V and the number of clustering individuals T.

large T; see also Table 2. The fitted linear model can be used to predict the time required for agglomerative clustering for large T or V. However, β_0 is computer-dependent. On the equipment mentioned above, the time needed for clustering $T = 100$ individuals measured on $V = 5000$ variables is about 39 minutes for the Gaussian model and about 33 minutes for the asymmetric Laplace model.

6. Clustering toy examples

In this section only fits using the Gaussian model are presented. The asymmetric Laplace model leads to very similar results provided similar hyper-parameter values are used, but hyper-parameter estimation using the asymmetric Laplace model is often more difficult.

In order to demonstrate the usefulness of the `bclust` package first we cluster a toy data set consisting of a cluster of 20 observations independently and identically sampled from a standard bivariate Gaussian distribution with correlation $\rho = 0.9$, and another cluster of 20 Gaussian variates with mean $(4, 0)^\top$, unit variances and negative correlation $\rho = -0.9$. This gives a data set in which one variable is more useful for clustering than another. A scatterplot of the data is shown in Figure 4. The generated data violate the variable independence assumption of model (3), but, provided the cluster centers are separated reasonably well, ignoring the dependence has little effect on the estimated grouping and the algorithm yields convincing results.

The `bclust` function is the main command of the **bclust** package that implements the Bayesian clustering described earlier. The essential arguments of the command are a numeric matrix, with subjects in rows and variables in columns, and the hyper-parameter values. The commands presuppose that data are unreplicated, the clustering effect distribution is Gaussian, and the hyper-parameters are in a vector with a specific order and are transformed as $(\log \sigma_\varepsilon^2, \log \sigma_\eta^2, \log \sigma_\theta^2, \mu, \operatorname{logit} p, \operatorname{logit} q)^\top$.

The command for generating the toy data, performing Bayesian clustering, and plotting the output using the hyper-parameters $\sigma_\varepsilon^2 = 1, \sigma_\eta^2 \approx 0, \sigma_\theta^2 = 16, \mu = 0, p = 0.5$, and $q = 0.5$ is

```
R> set.seed(150)
R> library("MASS")
R> library("bclust")
R> x <- rbind(mvrnorm(20, c(0, 0), matrix(c(1, 0.9, 0.9, 1), 2, 2)),
+    mvrnorm(20, c(4, 0), matrix(c(1, -0.9, -0.9, 1), 2, 2)))
R> cluster.labels <- paste(c(rep(1, 20), rep(2, 20)))
R> cluster.obj <- bclust(x, transformed.par = c(0, -50, log(16), 0, 0, 0),
+    labels = cluster.labels)
R> plot(cluster.obj)
```

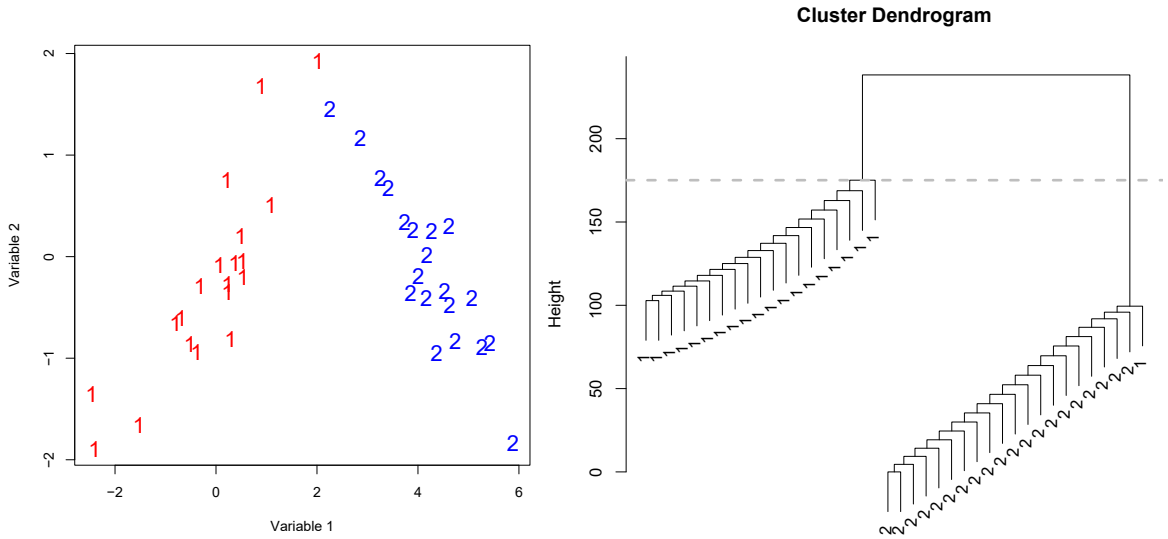

Figure 4: Left panel: scatter plot of the bivariate toy data; two clusters are generated, the first cluster from a standard bivariate Gaussian distribution with correlation 0.9 the second with mean $(4, 0)^\top$, $\rho = -0.9$, and unit marginal variances. Right panel: The output of `bclust` fit on the toy example visualized using the generic `plot` command.

```
R> abline(h = cluster.obj$cut, lty = 2, col = "gray", lwd = 3)
```

Here we set the hyper-parameters to specified values, but later we show how they can be estimated from data. The output of the code appears in the right panel of Figure 4.

The result of the `bclust` command is a `bclustvs` object, similar to the existing R class `hclust`, but including extra information needed to produce appropriate graphs.

Calculation of the importances is provided in the `imp` function. This imports a `bclustvs` object and gives the importance measures, the log Bayes factor of variables, $\log B_\delta$, and the log Bayes factor of variable-cluster combinations, $\log B_\gamma$, for the maximum a posteriori clustering found by the agglomerative method. Negative values of the importances give negative evidence that the variable v or the variable-cluster v, c participate in the optimal clustering. The package provides the following command for plotting variable importances:

```
R> viplot(imp(cluster.obj)$var)
```

The **bclust** package is not designed to handle low dimensional datasets like the bivariate toy data. In order to demonstrate its usefulness in a high-dimensional situation we add 98 standard Gaussian noise variables to the bivariate toy example, yielding a data set with 40 observations and 100 dimensions; we use the same hyper-parameter values as in the bivariate case. The package includes a dendrogram-image-teeth plot, `ditplot`, of a `bclustvs` object and draws the dendrogram tree, the image plot of the unreplicated data, and the optimal grouping, in the same figure.

```
R> x <- cbind(x, matrix(rnorm(3920), 40, 98))
R> cluster.obj <- bclust(x, transformed.par = c(0, -50, log(16), 0, 0, 0),
+    labels = cluster.labels)
```

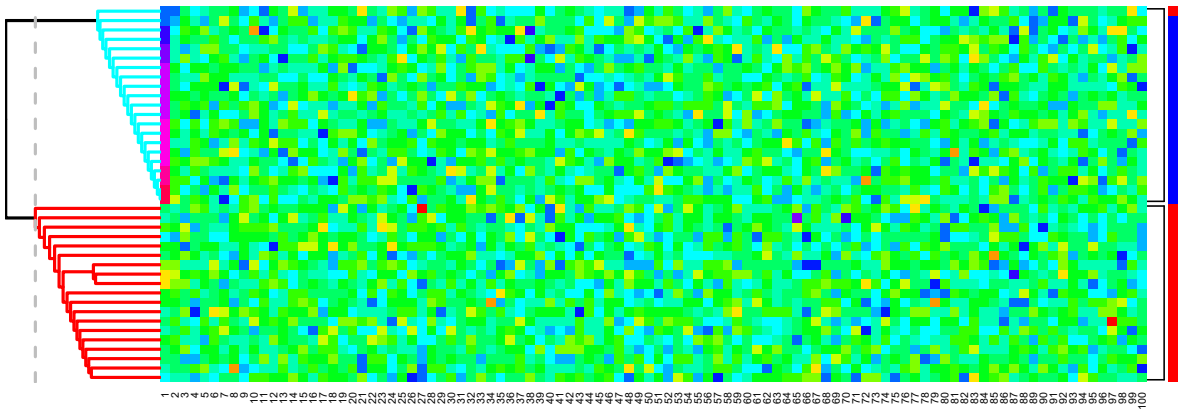

Figure 5: The output of `bclust` fit on the bivariate toy data (Figure 4 left panel), with 98 standard Gaussian noise variables added. The resulting `bclustvs` object is visualized using the `ditplot` command.

```
R> plotcol <- rep(c(2, 4), each = 20)
R> ditplot(cluster.obj, plot.width = 12, horizbar.distance = 0,
+     dendrogram.lwd = 2, xlab.cex = 0.6, ylab.cex = 0.6,
+     vertbar = plotcol, vertbar.col = plotcol, ylab.mar = 0)
```

The output of the code is shown in Figure 5.

On the left of the figure is the dendrogram tree. To its immediate right is the image plot of the data, with clustering individuals in rows and variables in columns. The package uses the `rainbow` color scheme as the default coloring scheme for the image plot: the minimum value appears blue, the maximum value appears magenta, and intermediate values are shown with colors that depend on their closeness to the limiting values. The image plot is followed by a teeth plot showing the optimal grouping found by cutting the tree. On the extreme right of the figure is a vertical bar which can be used to represent another arbitrary grouping, here taken to be the correct data clustering available in the `plotcol` numeric vector.

Our next example includes data simulated from the replicated Gaussian model (3) with $R_{ct} = 4$ and $T = 10$ grouped in four clusters with model hyper-parameters $\sigma^2_\varepsilon = 1, \sigma^2_\eta = 3, \sigma^2_\theta = 25, \mu = 0, p = 0.5, q = 0.5$. Figure 6 shows the `profileplot` of the data with red blobs for activated variable-cluster combinations attached to a `teethplot` and a horizontal bar declaring the activated variables, shown in red. The `profileplot` command of the **bclust** package is a handy tool suitable for presentation of replicated data.

Fitting the model (3) requires specification or estimation of the model hyper-parameters. For real data we propose estimation using maximum marginal likelihood before applying agglomerative clustering, i.e., considering every individual as a separate cluster. The `loglikelihood` command calculates the marginal log likelihood for a given dataset. One may adopt the `optim` or the `nlm` R functions to maximize the log likelihood and then cluster the data using the estimated hyper-parameters. Estimation of the model hyper-parameters using maximum likelihood for the simulated data yields $\sigma^2_\varepsilon = 1.03, \sigma^2_\eta = 1.13, \sigma^2_\theta = 25.77, \mu = 0.14, p = 0.60$ and $q = 0.41$.

Simulated Gaussian data is provided in supplementary materials in `simG.RData`. Assuming

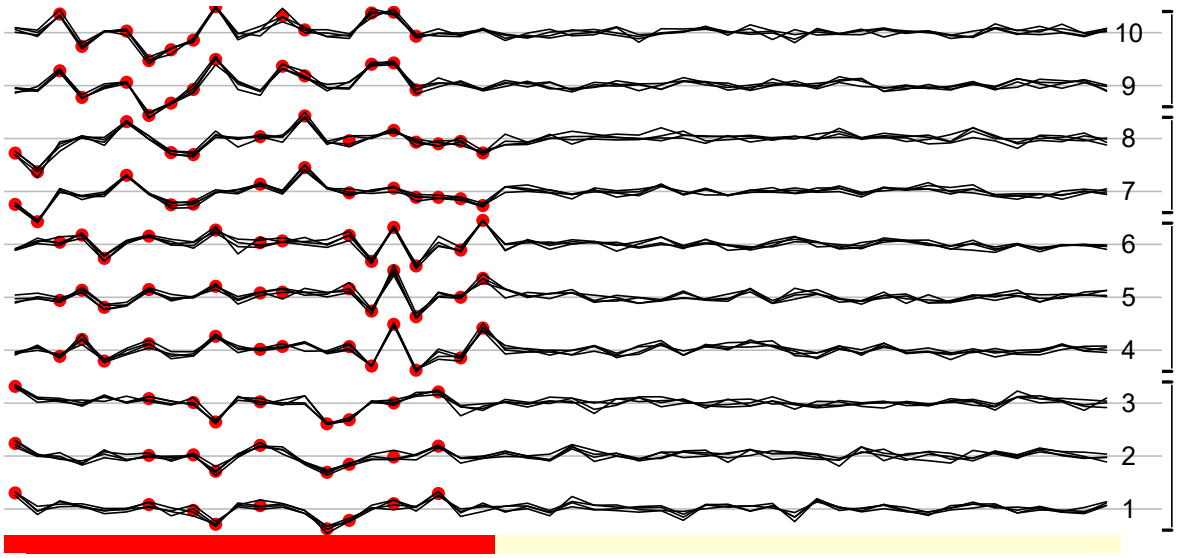

Figure 6: Profile plot of data with four clusters, simulated using the Gaussian model with ten clustering individuals, each having four replicates over 50 variables. The hyper-parameter values are $\sigma_\varepsilon^2 = 1, \sigma_\eta^2 = 1, \sigma_\theta^2 = 25, \mu = 0, p = 0.5, q = 0.5$. The 22 activated variables are shown using the red horizontal bar. Activated variable-cluster combinations are shown using the red blobs on the profiles for each group.

that the simulated data are stored in `x`, sample code for hyper-parameter estimation and fitting the Bayesian clustering with the Gaussian model is

```
R> library("bclust")
R> load("simG.RData")
R> x.id <- rep(1:10, each = 4)
R> meansumsq <- meancss(x, x.id)
R> optimfunc <- function(phi) {
+    -loglikelihood(x.mean = meansumsq$mean, x.css = meansumsq$css,
+      repno = meansumsq$repno, transformed.par = phi)
+  }
R> x.tpar <- optim(rep(0, 6), optimfunc, method = "BFGS")$par
R> bclust.obj <- bclust(x, rep.id = x.id, transformed.par = x.tpar)
R> dptplot(bclust.obj, scale = 20, horizbar.plot = TRUE,
+    varimp = imp(bclust.obj)$var, horizbar.distance = 0, dendrogram.lwd = 2)
```

See Figure 7 for the output.

Like `ditplot`, the function `dptplot` is intended to facilitate visualization of a `bclustvs` object, but when data are replicated. This function attaches a dendrogram plot to a profile plot and a teeth plot with an optional horizontal bar for variable importances and an optional vertical bar to show an arbitrary grouping. The pre-specified heat colors for the variable importances are determined by the scale proposed by Kass and Raftery (1995), blank for variables having negative Bayes factors and a heat color for positive ones. In the profile plot the variables are sorted automatically according to their importances.

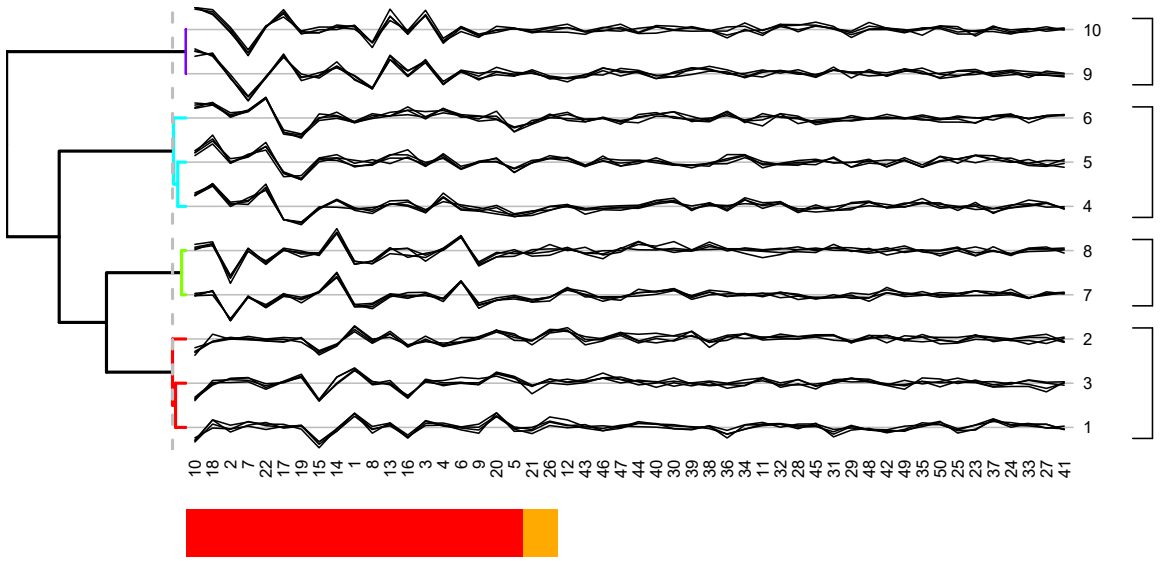

Figure 7: The output of `bclust` fit on simulated data visualized by `dptplot` command, see also Figure 6. The left side of the figure includes the posterior-based dendrogram cut at the maximum posteriori point, data are shown in the middle, and the maximum a posteriori grouping on the right side.

The `viplot` figure, which uses red for variables having a positive Bayes factor (important), and white for variables having a negative Bayes factor (unimportant), shows that one of the inactive variables has a slightly positive Bayes factor and two out of the 22 activated variables are thought to be unimportant:

```
R> viplot(imp(bclust.obj)$var, col = as.numeric(imp(bclust.obj)$var > 0) * 2)
```

The output of the former code appears in Figure 8.

7. Clustering real data

Many branches of science produce high-dimensional data for which classification of a new observation to existing groups or clustering individuals is of interest. In most cases a list of potentially important variables is available and effective discriminating or clustering variables are demanded. Here we consider a subset of the replicated metabolomic data of Messerli *et al.* (2007) available in the **bclust** package as the `gaelle` data set.

The metabolite data consist of 14 mutant samples of the plant *Arabidopsis thaliana*. Values of 43 potentially important metabolites are measured for each sample using GC-MS technology. These metabolites are supposed to depend on the genetic changes. The data involve two mutants defective in starch bio-synthesis, `pgm` and `isa2`; four defective in starch degradation `sex1`, `sex4`, `mex1`, and `dpe2`; a mutant for comparison that accumulates starch as a pleitropic effect, `tpt`; four uncharacterized mutants, `deg172`, `deg263`, `ke103`, and `sex3`; and three wild type plants, `WsWT`, `RLDWT`, and `ColWT`. There are four replicates of all samples except the last, for which there are three.

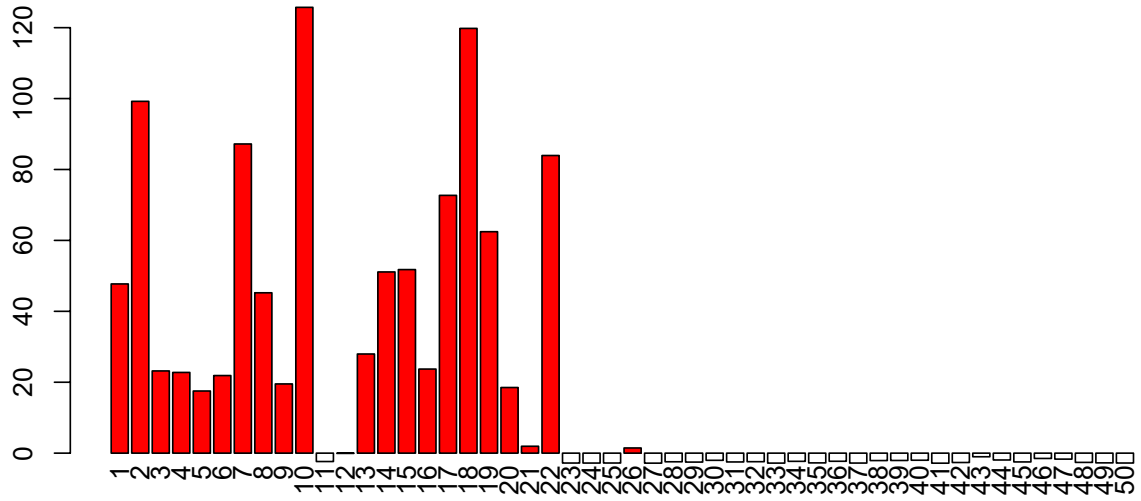

Figure 8: The bar plot of the variable Bayes factors using `viplot` for the simulated data; see also Figure 6.

When a small proportion of variables are active, i.e., q is small, it is often difficult to estimate the model hyper-parameters using maximum marginal likelihood. We therefore fix $q = 1$ and estimate the remaining parameters. The resulting procedure is equivalent to setting $\delta_v = 1$, for all $v = 1, \ldots, V$ in (3). One may use this modified model without variable selection to discriminate or cluster subjects. This model is still resistant to the noise variables, thanks to the variable-cluster selector γ_{vc}, but it does not include variable selection and therefore no variable importance can be computed. Simulation shows that parameter estimation and clustering performs better when $q = 1$. When noise dominates, estimation of all parameters simultaneously is troublesome, and usually fixing $q = 1$ provides a better result also.

If variable selection is needed when pq is small or the noise level is high, we can help the optimization routine by fixing the model parameters $\sigma_\varepsilon^2, \sigma_\eta^2, \sigma_\theta^2, \mu$ to their values already obtained by fixing $q = 1$, and then estimate only p and q. With the `gaelle` data the following sample code uses these fudges to give $\sigma_\varepsilon^2 = 0.16, \sigma_\eta^2 = 0.37, \sigma_\theta^2 = 5.10, \mu = 0.08, p = 0.03$ for the model with $q = 1$, and then the second step gives $p = 0.46, q = 0.16$ for the variable selection model. Iterating between these two estimation steps typically gives estimates that do not vary significantly. Confidence intervals for parameters can be found using the delta method or a profile likelihood; the latter helps the user to see whether the parameters are well-estimated. The profile likelihood curves for p and q are flat when pq is very small or the noise level dominates, i.e., σ_θ^2 is tiny compared with σ_η^2 and σ_ε^2. We do not suggest using our package in these cases without careful attention to parameter estimation. Optimization of the marginal likelihood for the asymmetric Laplace model is often more difficult. We suggest checking the convergence of the optimization routine for the estimated parameters before performing clustering, discrimination, and calculating the importances.

```
R> library("bclust")
R> data("gaelle")
R> x <- gaelle
R> x.id <- rep(1:14, c(3, rep(4, 13)))
```

```
R> meansumsq <- meancss (x, x.id)
R> optimfunc <- function(phi) {
+    -loglikelihood(x.mean = meansumsq$mean, x.css = meansumsq$css,
+      repno = meansumsq$repno, transformed.par = phi, var.select = FALSE)
+  }
R> xinit.tpar <- optim(rep(0, 5), optimfunc, method = "BFGS")$par
R> optimfunc <- function(phi) {
+    -loglikelihood(x.mean = meansumsq$mean, x.css = meansumsq$css,
+      repno = meansumsq$repno, transformed.par = c(xinit.tpar[1:4], phi))
+  }
R> x.tpar <- c(xinit.tpar[1:4], optim(rep(0, 2), optimfunc,
+    method = "BFGS")$par)
```

The first three rows of the `gaelle` data set include replications of the uncharacterized plant `ColWT`. The posterior discrimination percentages of `ColWT` using flat prior probabilities without and with variable selection are:

```
R> x.labels <- c("ColWT", "d172", "d263", "isa2", "sex4", "dpe2", "mex1",
+    "sex3", "pgm", "sex1", "WsWT", "tpt", "RLDWT", "ke103")
R> bdiscrim(training = x[-(1:3), ], training.id = (x.id[-(1:3)] - 1),
+    training.labels = x.labels[-1], predict = x[1:3, ],
+    transformed.par = xinit.tpar, var.select = FALSE)$probs * 100

        d172 d263 isa2 sex4 dpe2 mex1 sex3 pgm sex1 WsWT  tpt RLDWT ke103 New
ColWT.1  9.6 10.6    0  8.6    0    0  9.4 0.4  4.3 16.7 14.6  15.7     3 7.1

R> bdiscrim(training = x[-(1:3), ], training.id = (x.id[-(1:3)] - 1),
+    training.labels = x.labels [-1], predict = x[1:3, ],
+    transformed.par = x.tpar)$probs * 100

        d172 d263 isa2 sex4 dpe2 mex1 sex3 pgm sex1 WsWT  tpt RLDWT ke103 New
ColWT.1  2.8  3.8    0  1.7    0    0  6.4 0.2  0.5 36.8 15.2  14.7  11.7 6.2
```

The `bdiscrim` function discriminates an individual to one of the previously seen groups using the model (3). It also calculates the probability that the discriminating individual belongs to none of these groups above, denoted by `New`, and computes the log Bayes factors for δ_v and γ_{vc} in order to measure the importances of variable v and variable-class combination v, c, respectively.

The clusterings for the `gaelle` data with and without variable selection are similar, so we report only the result of the variable selection model. The figure below confirms that our Bayesian clustering method clusters the data into five groups and finds 11 of the 43 variables to be important.

```
R> bclust.obj <- bclust(x, rep.id = x.id, transformed.par = x.tpar,
+    labels = x.labels)
R> dptplot(bclust.obj, scale = 10, horizbar.plot = TRUE,
+    varimp = imp(bclust.obj)$var, horizbar.distance = 5, dendrogram.lwd = 2)
```

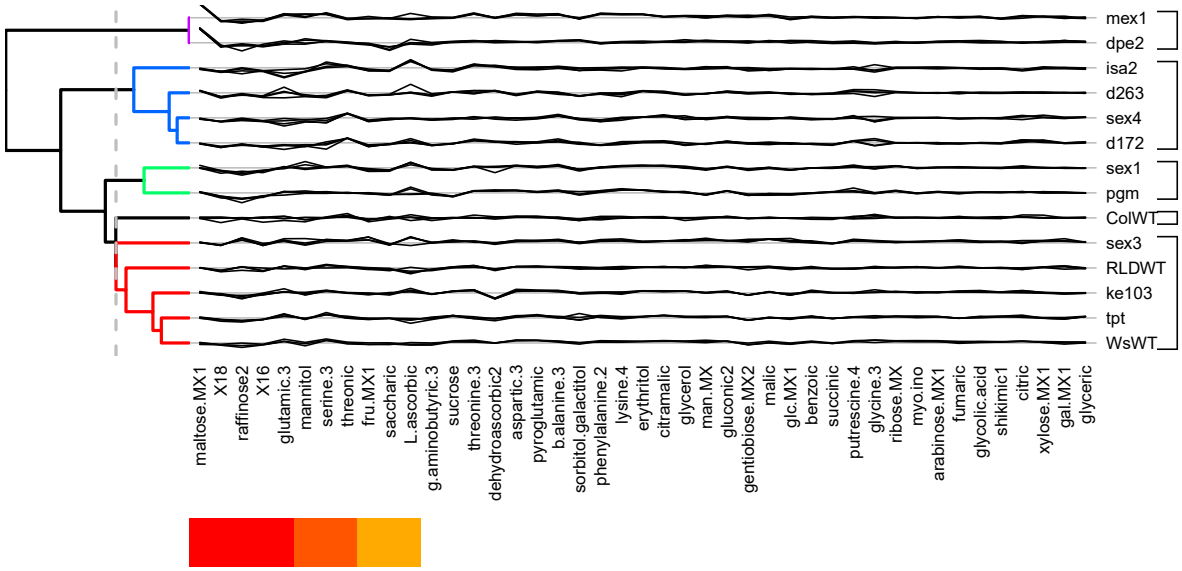

Figure 9: The `bclust` fit on the `gaelle` data, visualized using `dptplot`.

The output of the latter code appears in Figure 9.

Acknowledgments

The authors thank Nicole Sierro, Gaëlle Messerli, Samuel Zeeman, and Kjell Konis. Vahid Partovi Nia was supported by the Swiss SNF Fellowship PBELP2-125531S in the context of the Swiss National Center for Competence in Research in Plant Survival (http://www.unine.ch/nccr/), the NSERC Discovery Grant 341315/2006, and the VAHSR&D Grant IIR 07-229.

References

Ahn J, Marron JS, Muller KM, Chi YY (2007). "The High-Dimension, Low-Sample-Size Geometric Representation Holds Under Mild Conditions." *Biometrika*, **94**(3), 760–766.

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J, Croz JD, Greenbaum A, Hammarling S, McKenney A, Sorensen D (1999). ***LAPACK*** *Users' Guide*. 3rd edition. Society for Industrial and Applied Mathematics, Philadelphia.

Bergé L, Bouveyron C, Girard S (2012). "**HDclassif**: An R Package for Model-Based Clustering and Discriminant Analysis of High-Dimensional Data." *Journal of Statistical Software*, **46**(6), 1–29. URL http://www.jstatsoft.org/v46/i06/.

Bernardo JM, Smith AFM (1994). *Bayesian Theory*. John Wiley & Sons, New York.

Bhowmick D, Davison AC, Goldstein DR, Ruffieux Y (2006). "A Laplace Mixture Model for Identification of Differential Expression in Microarray Experiments." *Biostatistics*, **7**, 630–641.

Bondell HD, Reich BJ (2008). "Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR." *Biometrics*, **64**, 115–123.

Booth JG, Casella G, Hobert JP (2008). "Clustering Using Objective Functions and Stochastic Search." *Journal of the Royal Statistical Society B*, **70**, 119–139.

Chang WC (1983). "On Using Principal Components Before Separating a Mixture of Two Multivariate Normal Distributions." *Applied Statistics*, **32**, 267–275.

Cheeseman P, Stutz J (1996). "Bayesian Classification (**AutoClass**): Theory and Results." In *Advances in Knowledge Discovery and Data Mining*, pp. 153–180. The MIT Press.

Dean N, Raftery AE (2009). ***clustvarsel**: Variable Selection for Model-Based Clustering*. R package version 1.3, URL http://CRAN.R-project.org/package=clustvarsel.

Dimitriadou E (2009). ***cclust**: Convex Clustering Methods and Clustering Indexes*. R package version 0.6-16, URL http://CRAN.R-project.org/package=cclust.

Friedman JH, Meulman JJ (2004). "Clustering Objects on Subsets of Attributes." *Journal of the Royal Statistical Society B*, **66**, 815–849.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). "**Bioconductor**: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology*, **5**, R80. URL http://genomebiology.com/2004/5/10/R80.

George EI, McCulloch RE (1997). "Approaches for Bayesian Variable Selection." *Statistica Sinica*, **7**, 339–373.

Ghosh D, Chinnaiyan AM (2002). "Mixture Modelling of Gene Expression Data From Microarray Experiments." *Bioinformatics*, **18**, 275–286.

Hall P, Marron JS, Neeman A (2005). "Geometric Representation of High Dimension, Low Sample Size Data." *Journal of the Royal Statistical Society B*, **67**, 427–444.

Heard NA, Holmes CC, Stephens DA (2006). "A Quantitative Study of Gene Regulation Involved in the Immune Response of *Anopheline* Mosquitoes: An Application of Bayesian Hierarchical Clustering of Curves." *Journal of the American Statistical Association*, **101**, 18–29.

Hoff PD (2006). "Model-Based Subspace Clustering." *Bayesian Analysis*, **1**, 321–344.

Kass AE, Raftery AE (1995). "Bayes Factors." *Journal of the American Statistical Association*, **90**, 773–795.

Kim S, Tadesse MG, Vannucci M (2006). "Variable Selection in Clustering via Dirichlet Process Mixture Models." *Biometrika*, **93**, 877–893.

Lance GN, Williams WT (1967). "A General Theory of Classificatory Sorting Strategies 1. Hierarchical Systems." *The Computer Journal*, **9**, 373–380.

Leisch F (2010). "Neighborhood Graphs, Stripes and Shadow Plots for Cluster Visualization." *Statistics and Computing*, **20**, 457–469.

Liu JS, Zhang JL, Palumbo MJ, Lawrence CE (2003). "Bayesian Clustering with Variable and Transformation Selections." In JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, M West (eds.), *Bayesian Statistics 7*, pp. 249–275. Oxford University Press.

Mächler M, Rousseeuw P, Struyf A, Hubert M (2012). "Cluster Analysis Basics and Extensions." R package version 1.14.2, URL http://CRAN.R-project.org/package=cluster.

McLachlan GJ, Bean RW, Peel D (2002). "A Mixture Model-Based Approach to the Clustering of Microarray Expression Data." *Bioinformatics*, **18**, 413–422.

Messerli G, Partovi Nia V, Trevisan M, Kolbe A, Schauer N, Geigenberger P, Chen J, Davison AC, Fernie AR, Zeeman SC (2007). "Rapid Classification of Phenotypic Mutants of Arabidopsis via Metabolite Fingerprinting." *Plant Physiology*, **143**, 1481–1492.

Mitchell TJ, Beauchamp JJ (1988). "Bayesian Variable Selection in Linear Regression." *Journal of the American Statistical Association*, **83**, 1023–1036.

Paradis E, Claude J, Strimmer K (2004). "**ape**: Analyses of Phylogenetics and Evolution in R Language." *Bioinformatics*, **20**, 289–290.

Partovi Nia V (2009). *Fast High-Dimensional Bayesian Classification and Clustering*. Ph.D. thesis, Ecole Polytechnique Fédérale de Lasuanne. URL http://library.epfl.ch/en/theses/?nr=4482.

Partovi Nia V, Stephens DA (2012). *Probability: Interpretation, Theory and Applications*, chapter Dendrogram Represnetation of Stochastic Clustering. Nova Publishers, New York.

Raftery AE, Dean N (2006). "Variable Selection for Model-Based Clustering." *Journal of the American Statistical Association*, **101**, 168–178.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rossi P (2011). ***bayesm**: Bayesian Inference for Marketing/Micro-econometrics*. R package version 2.2-4, URL http://CRAN.R-project.org/package=bayesm.

Savage RS, Heller K, Xu Y, Ghahramani Z, Truman WM, Grant M, Denby KJ, Wild DL (2009). "R/**BHC**: Fast Bayesian Hierarchical Clustering for Microarray Data." *BMC Bioinformatics*, **10**, 242.

Tadesse MG, Sha N, Vannucci M (2005). "Bayesian Variable Selection in Clustering High-Dimensional Data." *Journal of the American Statistical Association*, **100**, 602–617.

Wang S, Zhu J (2008). "Variable Selection for Model-Based High-Dimensional Clustering and Its Application to Microarray Data." *Biometrics*, **64**, 440–448.

Witten DM, Tibshirani R (2011). ***sparcl**: Perform Sparse Hierarchical Clustering and Sparse K-Means Clustering*. R package version 1.0.2, URL http://CRAN.R-project.org/package=sparcl.

Witten DM, Tibshirani RJ (2010). "A Framework for Feature Selection in Clustering." *Journal of the American Statistical Association*, **105**, 713–726.

Affiliation:

Vahid Partovi Nia
Department of Mathematics and Industrial Engineering
Ecole Polytechnique de Montréal
2900 Edouard-Montpetit
H3T 1J4 Montréal, Canada
E-mail: vahid.partovinia@polymtl.ca

Anthony C. Davison
Ecole Polytechnique Fédérale de Lausanne
EPFL-FSB-MATHAA-STAT, Station 8
1015 Lausanne, Switzerland
E-mail: anthony.davison@epfl.ch

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 47, Issue 5 *Submitted:* 2010-01-15
April 2012 *Accepted:* 2012-04-05