Ca' Foscari
University
of Venice

Department
of Economics

Working Paper

Stefano Tonellato

Bayesian nonparametric clustering as a community detection problem

ISSN: 1827-3580
No. 28/WP/2019

Working Papers
Department of Economics
Ca' Foscari University of Venice
No. 20/WP/2019
ISSN 1827-3580

Bayesian nonparametric clustering as a community detection problem

Stefano Tonellato
Ca' Foscari University of Venice

Abstract
It is well known that a wide class of bayesian nonparametric priors lead to the representation of the distribution of the observable variables as a mixture density with an infinite number of components, and that such a representation induces a clustering structure in the observations. However, cluster identification is not straightforward a posteriori and some post-processing is usually required. In order to circumvent label switching, pairwise posterior similarity has been introduced, and it has been used in order to either apply classical clustering algorithms or estimate the underlying partition by minimising a suitable loss function. This paper proposes to map observations on a weighted undirected graph, where each node represents a sample item and edge weights are given by the posterior pairwise similarities. It will be shown how, after building a particular random walk on such a graph, it is possible to apply a community detection algorithm, known as map equation method, by optimising the description length of the partition. A relevant feature of this method is that it allows for both the quantification of the posterior uncertainty of the classification and the selection of variables to be used
for classification purposes.

Keywords
Dirichlet process priors, mixture models, community detection, entropy, variable selection

JEL Codes
C11, C38

Address for correspondence:
Stefano Tonellato
Department of Economics
Ca' Foscari University of Venice
Cannaregio 873, Fondamenta S.Giobbe
30121 Venezia - Italy
e-mail: stone@unive.it

The Working Paper Series
is available only on line
(http://www.unive.it/pag/16882/)
For editorial correspondence, please contact:
wp.dse@unive.it

Department of Economics
Ca' Foscari University of Venice
Cannaregio 873, Fondamenta San Giobbe
30121 Venice Italy
Fax: ++39 041 2349210

Bayesian nonparametric clustering as a community detection problem

Stefano Tonellato
Department of Economics
Università Ca' Foscari Venezia

Abstract

It is well known that a wide class of bayesian nonparametric priors lead to the representation of the distribution of the observable variables as a mixture density with an infinite number of components, and that such a representation induces a clustering structure in the observations. However, cluster identification is not straightforward a posteriori and some post-processing is usually required. In order to circumvent label switching, pairwise posterior similarity has been introduced, and it has been used in order to either apply classical clustering algorithms or estimate the underlying partition by minimising a suitable loss function. This paper proposes to map observations on a weighted undirected graph, where each node represents a sample item and edge weights are given by the posterior pairwise similarities. It will be shown how, after building a particular random walk on such a graph, it is possibile to apply a community detection algorithm, known as map equation method, by optimising the description length of the partition. A relevant feature of this method is that it allows for both the quantification of the posterior uncertainty of the classification and the selection of variables to be used for classification purposes.

Keywords

Dirichlet process priors, mixture models, community detection, entropy, variable selection.

JEL Codes

C11, C38

1 Introduction

Cluster analysis, or unsupervised learning, aims to detecting homogeneous groups within heterogeneus collections of items and it plays an important role in a wide range of scientific disciplines. The earliest contributions on this topic date back to the third decade of the past century (Zubin, 1938; Tyron, 1939). Partitional and hierarchical algorithms have been proposed later on and thorough treatments can be found in Hartigan (1975); Kaufman and Rousseeuw (1990) and Gordon (1999). These classical clustering methods have been widely used in exploratory data analysis, but due to their heuristic nature they are not immediately suitable for inferential purposes. More recently, thanks to the increasing availability of computing power, clustering methods based on statistical models have been developed, with finite mixture distributions playing a central role (McLachlan and Peel, 2000; Frühwirth-Schnatter, 2006). Within the K-component mixture model based clustering, a particularly popular classification method (Fraley and Raftery, 2002) is the one that maximises the posterior probability of allocating each sample item to one of K clusters, which are identified with the K mixture components. The number of components and the specific parametrisation of the components are determined by minimising the BIC criterion. In the paper, we shall refer to this method by using the acronymous MAP. The possibility of implementing MCMC algorithms and the development of Bayesian nonparametric methods have had a strong impact in model based clustering. Some of the most popular Bayesian non parametric prior processes allow for the representation of the likelihood as a mixture with an infinite number of components and random weights. This is true for the Dirichlet process prior, for the Pitman-Yor process prior and for a wide class of normalised random measures (Sethuraman, 1994; Ishwaran and James, 2001; Lijoi and Prünster, 2010). Such a property induces naturally a clustering structure in the model and implies a prior probability distribution on sample partitions. Unfortunately, the exchangeability assumption makes the mixture components unidentifiable, and therefore the identification of clusters is not straightforward. The implementation of MCMC algorithms, however, produces a sample from the posterior distribution of the partitions which can be used in different ways for clustering purposes. Medvedovic and Sivaganesan (2002) and Medvedovic and Guo (2004) estimate the posterior pairwise similarity, i. e. the posterior probability, τ_{ij}, that items i and j are generated by the same mixture component through the proportion of simulated partitions allocating i and j to the same cluster. They can then define the pairwise dissimilarity, $1 - \tau_{ij}$, and apply classical clustering algorithms. Under alternative approaches, the underlying partition is seen as an unknown parameter and a posterior estimate is obtained by serching for the posterior mode (Dahl, 2009), maximising a given criterion, like the expected adjusted Rand index in Fritsch and Ickstadt (2009), or minimising a loss function (Lau and Green, 2007; Wade and Ghahramani, 2018). Wade and Ghahramani (2018) show that the variation of information (VI, Meilă (2007)) and Binder's loss (Binder, 1978) are two metrics in the space of partitions. Furthermore, they show that these two metrics are aligned with the lattice structure of the space of partitions. These results allow them to estimate the underlying partition by minimising either loss function and quantify the posterior uncertainty by defining suitable posterior credible regions in the partition space, called credible balls. As pointed out by Frühwirth-Schnatter et al. (2018), their methodology can be utilised for both infinite and finite mixture models.

Graphs representing social, biological, technological and information networks are usually characterised by a strong inhomogeneity in the distribution of edges among nodes: high concetrations of edges within some particular groups of nodes and low concentrations of edges between thes groups may coexist. Such a feature is called community structure or clustering and community detection has attracted a lot of attention among the researchers analysing network systems, and in particular statistical physicists (Fortunato, 2010). In this paper we propose to look at the sample as a weighted undirected graph whose nodes represent sample items and edge weights are given by the posterior pairwise similarities induced by a Bayesian nonparametric model. Under this perspective, clustering can be seen as a community detection problem. In particular, we shall consider the approach based on the map equation (Rosvall and Bergstrom, 2008; Rosvall et al., 2009). The optimal partition will be defined as that partition that minimises the expected description length of a particular random walk defined on such a graph. In section 2 we shall review some of the main features of the map equation method and of a wide class of Bayesian nonparametric models. Section 3 will illustrate how the optimal clustering can be achieved and posterior uncertainty can be evaluated. In section 4 we suggest an algorithm aimed to selecting a subset of variables for clustering purposes in a multivariate context. Examples on simulated and real data will also be given in sections 3 and 4.

2 Review

In this section we give a short description of the community detection based on the map equation and recall some features of a wide class of Bayesian nonparametric models that are particularly relevant for our purposes.

2.1 The map equation

Rosvall and Bergstrom (2008) and Rosvall et al. (2009) used maps in order to describe a flow dynamics across the links and nodes in both directed and undirected weighted networks representing interactions among the sub-units of a system. In order to understand the flow of information on the network, they propose to identify the modules, i.e. the clusters of nodes, among which information flows quickly and easily by representing the whole system through a map. Such a map divides the network in two levels of description:

a) unique names are retained for the clusters within the network;

b) names associated with fine-grain details, i.e. the nodes, may be reused in different clusters.

The analogy with maps where city names are unique, but street names may be reused from one city to another is straightforward. Such coding problem can be seen as the description of a random walker spending long periods of time within certain clusters. Defining an efficient code is equivalent to identifying a module (i.e. cluster) partition that allows for the minimisation of the description length. Let $\mathcal{M}$ represent a partition of n nodes in K groups. Using a binary code in order to label nodes and clusters, the average description length of a step of the random walker is

$$L(\mathcal{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{k=1}^{K} p^k_{\circlearrowright} H(\mathcal{P}^k), \tag{1}$$

where the first term represents the entropy of the movement between two modules, and the second the entropy of movements within modules. More precisely:

$$\begin{aligned} q_{\curvearrowright} &= \text{probability of leaving a cluster;} \\ H(\mathcal{Q}) &= \text{average length of the code used to specify cluster names;} \\ H(\mathcal{P}^k) &= \text{average length of codewords in cluster } k\text{;} \\ p^k_{\circlearrowright} &= \text{probability of staying in or leaving cluster } k. \end{aligned}$$

All these probabilities can be easily computed once the transition probability of the random walk, i.e. the probability of moving from node i to node j, $i, j = 1, \ldots, n$, is properly defined. In Rosvall and Bergstrom (2008) such transition probabilities are proportional to the weights of the edges connecting i to its neighbours. Rosvall and Bergstrom (2008) call equation (1) the map equation. A more detailed description of the quantities appearing in (1) will be given in section 3.

The optimal partition $\mathcal{M}^*$ minimising (1) provides the minimum expected description length of the random walk on the network. The algorithm proposed by Rosvall et al. (2009) in order to minimise (1) is based on a slight modification of the one introduced in Blondel et al. (2008), which they call the core algorithm. The core algorithm works as follows.

Stage 1 Neighboring nodes are joined into modules, which subsequently are joined into supermodules and so on. First, each node is assigned to its own module. Then, in random sequential order, each node is moved to the neighboring module that results in the largest decrease of the map equation. If no move results in a decrease of the map equation, the node stays in its original module. This procedure is repeated, each time in a new random sequential order, until no move generates a decrease of the map equation.

Stage 2 At this stage the network is rebuilt, with the modules of the last level forming the nodes at this level. And exactly as at the previous level, the nodes are joined into modules. This hierarchical rebuilding of the network is repeated until the map equation cannot be reduced further. Except for the random sequence order, this is the algorithm described in Blondel et al. (2008).

The core algorithm reminds an agglomerative clustering. In fact, when two modules are merged in a single one at stage 2, they can never be separated in this algorithm. Therefore Rosvall et al. (2009) propose to improve its accuracy by introducing two further movements after stage 2 is completed:

Submodule movements. First, each cluster is treated as a network on its own and the main algorithm is applied to this network. This procedure generates one or more submodules for each module. Then all submodules are moved back to their respective modules of the previous step. At this stage, with the same partition as in the previous step but with each submodule being freely movable between the modules, the main algorithm is re-applied.

Single node movements. First, each node is re-assigned to be the sole member of its own module, in order to allow for single-node movements. Then all nodes are moved back to their respective modules of the previous step. At this stage, with the same partition as in the previous step but with each single node being freely movable between the modules, the main algorithm is re-applied.

These movements are repeated sequentially until no improvement in the expected description length are achieved, or when the improvement does not exeed a fixed tolerance.

As an example, consider a graph with adjacency matrix

$$\mathbf{W} = \begin{bmatrix} 0.00 & 0.00 & 0.90 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.01 & 0.00 \\ 0.90 & 0.50 & 0.00 & 0.00 & 0.01 \\ 0.00 & 0.01 & 0.00 & 0.00 & 0.90 \\ 0.00 & 0.00 & 0.01 & 0.90 & 0.00 \end{bmatrix}, \tag{2}$$

where the element $s_{i,j}$ represents the similarity between nodes i and j, $i, j = 1, \ldots, 5$. The optimal clustering is quite obviously given by the partition $\mathcal{M}^* = \{\{1, 2, 3\}, \{4, 5\}\}$ with $L(\mathcal{M}^*) = 1.784$. Figure 1 shows the corresponding graph and the clusters composing $\mathcal{M}^*$. Both computations and graphical representation are based on the R package `igraph` (Csardi and Nepusz, 2006).

2.2 Dirichlet process mixture model and some generalisations

A prominent class of models in Bayesian nonparametrics is based on the Dirichlet process prior (Ferguson, 1973) and is known as Dirichlet process mixture (Antoniak, 1974). In this model, the observable random variables, $\mathbf{Y}_i$, $i = 1, \ldots, n$, are assumed to be exchangeable and generated by the following hierarchical model:

$$\begin{aligned} \mathbf{Y}_i|\theta_i &\overset{ind}{\sim} p(\cdot|\theta_i),\ \theta_i \in \Theta \\ \theta_i|G &\overset{iid}{\sim} G \\ G &\sim DP(\alpha, G_0), \end{aligned}$$

where $DP(\alpha, G_0)$ denotes a Dirichlet process (DP) with base measure G_0 and concentration parameter $\alpha > 0$. Since the DP generates almost surely discrete random measures on the parameter space Θ, ties among the parameter values have positive probability, leading to a batch of clusters of the parameter vector $\theta = [\theta_1, \ldots, \theta_n]^T$. Exploiting the Pólya urn representation of the DP (Blackwell and MacQueen, 1973), the model can be rewritten as

$$\begin{aligned} \mathbf{Y}_i|z_i, \theta^*_{z_i} &\overset{ind}{\sim} p(\cdot|\theta^*_{z_i}),\ \theta^*_{z_i} \in \Theta & (3) \\ \theta^*_{z_i} &\overset{iid}{\sim} G_0 & (4) \\ z_1 &= 1 & \\ p(z_i = j|\mathbf{z}_{<i}) &= \begin{cases} \frac{\alpha}{\alpha+i-1} & j = \max(\mathbf{z}_{<i}) + 1 \\ \frac{n_j}{\alpha+i-1} & j \in \{\max(\mathbf{z}_{<i})\}, \end{cases} \quad i > 1 & (5) \\ z_i &\perp \theta^*_j \quad \forall i, j, & (6) \end{aligned}$$

where $\mathbf{z}_{<i} = \{z_1, \ldots, z_{i-1}\}$, and n_j is the number of θ_i's equal to θ^*_j, $j \in \{k\}$. In this model representation, the parameter θ can be expressed as $(\mathbf{z}, \theta^*)$, with $\mathbf{z} = \{z_1, \ldots, z_n\}$, $\theta^* = [\theta^*_1, \ldots, \theta^*_k]^T$, $k = \max(\mathbf{z}) \leq n$ with $\theta^*_j \overset{iid}{\sim} G_0$, and $\theta_i = \theta^*_{z_i}$. The labels in $\mathbf{z}$ identify a partition of $\{1, \ldots, n\}$ in k clusters, $\mathcal{M} = \{C_1, \ldots, C_k\}$ with prior probability (Dahl, 2009)

$$p(\mathcal{M}) \propto \alpha^k \prod_{j=1}^{k} \Gamma(|C_j|).$$

The model can be represented in the following, equivalent way:

$$
\begin{aligned}
p(y|P) &= \int K(y|\theta)dG(\theta),\\
G(\theta) &= \sum_{j=0}^{\infty} \omega_j \delta_{\theta_j},\\
\theta_j &\overset{iid}{\sim} G_0,
\end{aligned}
$$

where the ω_j and θ_j are independently distributed (Sethuraman, 1994). This representation holds for a class of Bayesian nonparametric priors more general than the Dirichlet process, including the Pitman-Yor process and normalised random measures (Lijoi and Prünster, 2010). Wade and Ghahramani (2018) provide a nice review of Bayesian nonparametric clustering. Here we want to highlight a common feature of the MCMC algorithms that have been defined in the literature: each one of them produces a sample of partitions from their posterior distribution that can be used to estimate the pairwise posterior similarity that will be discussed in the next section.

In the examples given in the following sections, we shall consider only models for continuous variables, with Dirichlet process priors with Gaussian base measure G_0. Nonetheless, the clustering method we are going to suggest can be straightforwardly extended to the more general class of Bayesian nonparametric models mentioned above as well as to finite mixtures.

3 The clustering method

3.1 Posterior similarity

We can state that two individuals, i and j, are similar if $\mathbf{y}_i$ and $\mathbf{y}_j$ are generated by the same mixture component, i.e. if $z_i = z_j$. Label switching prevents us from identifying mixture components, but not from assessing similarities among individuals. In fact, any of the MCMC algorithms currently available (see, for instance: Neal (2000); Walker (2007); Jain and Neal (2007); Papaspiliopoulos and Roberts (2008); Griffin and Walker (2011)), allows us to estimate the pairwise posterior similarity τ_{ij}. The posterior probability that x_i and x_j are generated by the same component, i.e. the posterior probability of the event $\{z_i = z_j\}$, can be estimated as

$$
\hat{\tau}_{ij} = \frac{1}{R}\sum_{r=1}^{R} I\left(z_i^{(r)}, z_j^{(r)}\right), \tag{7}
$$

where $z_i^{(r)}$ is the component label associated to the i-th sample item at the r-th run of the MCMC algorithm, $r = 1, \ldots, R$, $I(x, y) = 1$ if $x = y$ and $I(x, y) = 0$ otherwise. We can then define a similarity matrix $\mathbf{S}$ with ij-th element $s_{ij} = \hat{\tau}_{ij}$.

3.2 The map equation based on posterior similarity

The matrix $\mathbf{S}$ can be used to build the weighted undirected graph $G = (V, E)$, where each node in the set V represents a sample item, i. e. $V = \{1, \ldots, n\}$, and the set E,

$E \subseteq V \times V$, contains all the edges in G. Furthermore, the weight of the generic edge (i, j) is given by $w_{ij} = s_{ij}$ if $i \neq j$, and $w_{ij} = 0$ otherwise (the reason of this constraint will be explained below). The weight matrix will be denoted by $\mathbf{W} = \mathbf{S} - \mathbf{I}_n$.

We can then define a random walk $\mathcal{X}$ on G, with state space V. Let d_i represent the degree of vertex i, i.e.

$$d_i = \sum_{j=1}^{n} w_{ij}, \ i = 1, \ldots, n$$

and $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$, the n-dimensional diagonal matrix with the vertex degrees on the main diagonal. We define the transition matrix of $\mathcal{X}$ as

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W},$$

hence the probability of moving from i to j is given by,

$$p_{ij} = \frac{w_{ij}}{d_i}. \tag{8}$$

It follows that $p_{ii} = 0$ due to the constraint $w_{ii} = 0$: no self-loops are allowed under this assumption.

If G is connected, then $\mathcal{X}$ and its invariant distribution is $\pi = [\pi_1, \ldots, \pi_n]^T$, with

$$\pi_i = \frac{d_i}{\sum_{i,j} w_{ij}}, \ i = 1, \ldots, n \tag{9}$$

(Lovász, 1996). The random walk we have just defined represents an artificial stochastic flow such that the probability of moving from i to j is proportional to w_{ij}, i.e. to the similarity between i and j. Such a dynamics induces some high density subsets of V, i.e. subsets where the random walker spends a long time before moving to other clusters, separated by low weight edges. In such a context, we can utilise the community detection algorithms discussed in section 2.1. In order to do that, we shall now describe in detail how the quantities appearing in (1) can be computed. Let $\mathcal{M}$ represent an arbitrary partition of V in K clusters, $\mathcal{M} = \{C_1, \ldots, C_K\}$. The probability of leaving C_k is given by,

$$q_{k\curvearrowright} = \sum_{i \in C_k} \pi_i \sum_{j \notin C_k} p_{ij}, \quad k = 1, \ldots, K. \tag{10}$$

It follows that the probability of leaving one of the K clusters is given by

$$q_{\curvearrowright} = \sum_{k=1}^{K} q_{k\curvearrowright}.$$

The average length of the code used to specify cluster names is then equal to

$$H(\mathcal{Q}) = -\sum_{k=1}^{K} \frac{q_{k\curvearrowright}}{q_{\curvearrowright}} \log \left(\frac{q_{k\curvearrowright}}{q_{\curvearrowright}} \right).$$

The labels of the nodes in C_k are used at a rate

$$p_{\circlearrowright}^{k} = \sum_{i \in C_k} \pi_i + q_{k\curvearrowright},$$

i.e. $p^k_{\circlearrowright}$ is given by the fraction of time spent in C_k plus the probability of leaving C_k. Hence, the entropy for the k-th cluster codebook is

$$H(\mathcal{P}^k) = -\frac{q_{k\curvearrowright}}{q_{k\curvearrowright} + \sum_{i \in C_k} \pi_i} \log \left(\frac{q_{k\curvearrowright}}{q_{k\curvearrowright} + \sum_{i \in C_k} \pi_i} \right)$$
$$- \sum_{i \in C_k} \frac{\pi_i}{q_{k\curvearrowright} + \sum_{j \in C_k} \pi_j} \log \left(\frac{\pi_i}{q_{k\curvearrowright} + \sum_{j \in C_k} \pi_j} \right).$$

Then (1) can be written as

$$L(\mathcal{M}) = \left(\sum_{k=1}^{K} q_{k\curvearrowright} \right) \log \left(\sum_{k=1}^{K} q_{k\curvearrowright} \right) - 2 \sum_{k=1}^{K} q_{k\curvearrowright} \log(q_{k\curvearrowright})$$
$$- \sum_{i=1}^{n} \pi_i \log(\pi_i) + \sum_{k=1}^{K} \left(q_{k\curvearrowright} + \sum_{i \in C_k} \pi_i \right) \log \left(q_{k\curvearrowright} + \sum_{i \in C_k} \pi_i \right). \tag{11}$$

It is straightforward to verify that once the posterior similarity is computed using (7), all these quantities can be easily determined. Following (8) and (10), we can write, as in Rosvall and Bergstrom (2008):

$$q_{k\curvearrowright} = \sum_{i \in C_k} \sum_{j \notin C_k} \frac{w_{ij}}{\sum_{s,h} w_{sh}}$$

and all the terms in (11) can be rewritten as appropriate transformations of the weights in $\mathbf{W}$.

3.3 Quantifying uncertainty

Once a similarity matrix, $\mathbf{S}$, is defined, we can compute the entropy, $L(\mathcal{M})$, associated to any arbitrary partition. We can then define the equivalence relation between partitions, $\equiv$, such that for any pair of partitions, $\mathcal{M}_1$ and $\mathcal{M}_2$,

$$\mathcal{M}_1 \equiv \mathcal{M}_2 \Leftrightarrow L(\mathcal{M}_1) = L(\mathcal{M}_2) \tag{12}$$

A priori, the model specified by equations (3)-(6), assumes that any pair, $(\mathbf{y}_i, \mathbf{y}_j)$, of observations are generated by the same mixture component with a constant probability, $p(\alpha)$, independent of i and j. Henceforth, the prior similarity matrix $\tilde{\mathbf{S}}$ has elements $s_{ij} = p(\alpha)$ for $i \neq j$ and $s_{ii} = 1$. From the construction of the random walk on the graph G we illustrated in the previous subsection, it is straightforward to verify that the probability of moving from node i to node j is $\frac{1}{n-1}$, for $i \neq j$, and 0 otherwise, independently of i, j and α.

Proposition 1. *If $\mathbf{W}$, the transition matrix of the random walk $\mathcal{X}$ on the graph G, is such that $w_{ij} = \frac{1}{n-1}$ for $i \neq j$ and $w_{ij} = 0$ otherwise, the partition minimising (11) contains the unique cluster $\{1, \dots, n\}$.*

Proof. Under the stated assumption, it is easy to verify that, from (3.2),

$$q_{k\curvearrowright} = \frac{n_k(n - n_k)}{n(n-1)}, \; k = 1, \dots, K$$

and, from (9), $\pi_i = \frac{1}{n}$, $i = 1, \ldots, n$. Hence, (11) can be rewritten as

$$L(\mathcal{M}) = \left(\sum_{k=1}^{K} \frac{n_k(n-n_k)}{n(n-1)} \right) \log \left(\sum_{k=1}^{K} \frac{n_k(n-n_k)}{n(n-1)} \right) - 2 \sum_{k=1}^{K} \frac{n_k(n-n_k)}{n(n-1)} \log \left(\frac{n_k(n-n_k)}{n(n-1)} \right)$$

$$- \sum_{i=1}^{n} \frac{1}{n} \log \left(\frac{1}{n} \right) - \sum_{k=1}^{K} \left(\frac{n_k(n-n_k)}{n(n-1)} + \frac{n_k}{n} \right) \log \left(\frac{n_k(n-n_k)}{n(n-1)} + \frac{n_k}{n} \right). \quad (13)$$

Putting $0 \log 0 = 0$, and defining $\mathcal{M}^*$ as the partition given by the unique cluster $\{1, \ldots, n\}$, we can notice that in the computation of $L(\mathcal{M}^*)$, the first, second and fourth terms on the r.h.s. of (13) are equal to zero. It follows that

$$L(\mathcal{M}^*) = - \sum_{i=1}^{n} \frac{1}{n} \log \left(\frac{1}{n} \right).$$

We can also notice that for any partition composed by K clusters, with $K > 1$, the first, second and fourth terms on the r.h.s. of (13) are positive. Hence, $\mathcal{M}^*$ minimises (13). □

It follows that, a priori, the optimal clustering allocates all individuals to a unique group, $\{1, \ldots, n\}$, independently of the value taken by α. Furthermore, two sample partitions will be equivalent whenever they are composed by the same number of clusters, provided that the clusters in the two partitions have the same cardinality. Notice that this does not mean that the probability distribution of the random variable $L(\mathcal{M})$ is independent of α. In fact, under the Dirichlet process prior, for small values of α, partitions with a small number of clusters have high probabilty, whereas high values of α determine high prior probabilities of partitions with a high number of clusters (Antoniak, 1974). It follows that, for any fixed sample size n, our prior optimal clustering will be $\mathcal{M}^* = \{1, \ldots, n\}$, as a consequence of the exchangeability assumption, independently of our state of uncertainty about the number of clusters, which is related to α. Such uncertainty will be represented by the behaviour of $L(\mathcal{M})$. For small values of α the probability mass of $L(\mathcal{M})$ will be concentrated on the values corresponding to partitions with a small number of clusters; for high values of α, it will be concentrated on the values taken on the partitions with high number of clusters which is anyway bounded above by the sample size, n. Both these circumstances are characterised by a relatively low uncertainty about the number of clusters for any fixed n. Intermediate values of α determine an increase in the dispersion of the probability mass of $L(\mathcal{M})$, which represents a higher uncertainty about the number of clusters. Using equation (5) we can generate independent partitions and compute the value taken by $L(\mathcal{M})$ on each of them. Figure 2 provides an example of how the prior uncertainty about $L(\mathcal{M})$ (and hence about the number of clusters) depends on the value taken by α. We can notice that, for a fixed sample size ($n = 200$), the values taken by $L(\mathcal{M})$ are increasingly concentrated around the minimum (maximum) as *alpha* decreases (increases).

Once the posterior similarity has been estimated, it is also possible to sample from the posterior distribution of $L(\mathcal{M})$. When MCMC algorithms are used to estimate a Bayesian nonparametric model, at each iteration a new, random, sample partition is generated and the corresponding value of $L(\mathcal{M})$ can be computed. It is then possible to relate the optimal partition $\mathcal{M}^*$ computed by the algorithm described in section 2.1 and its expected code length $L(\mathcal{M}^*)$ with the posterior distribution of $L(\mathcal{M})$. It is worth to notice that quite often MCMC algorithms visit only a few of the many possible partitions and that each of them is visited only once or twice in many cases. It follows that, if we denote the partition generated by the MCMC algorithm that provides the minimum code length by $\mathcal{M}^{(1)}$, we often shall find that $L(\mathcal{M}^*) < L(\mathcal{M}^{(1)})$. However, due to the euristic nature of the optimising algorithm, in some rare occasion we might find that $L(\mathcal{M}^{(1)}) < L(\mathcal{M}^*)$. We can then define our best approximation of the optimal partition as

$$\tilde{\mathcal{M}} = \underset{\{\mathcal{M}^*, \mathcal{M}^{(1)}\}}{\text{argmin}} \; L(\mathcal{M}). \tag{14}$$

3.4 Two examples

Galaxy data. The Galaxy data (Roeder, 1990), shown in Figure 3a, consists of $n = 82$ measured velocities (in 10^3 km/s), relative to our own galaxy, of galaxies from six well separated conic sections in the *Corona Borealis* region. Roeder (1990) estimates a number of clusters between 3 and 7, identifying clusters with mixture components. This dataset hase been widely studied and quite different conclusions have been drawn by different researchers, as well documented by Aitkin (2011), who provides evidence for a number of three clusters under an objective Bayesian perspective. We might then expect that our method should warn us about a high level of uncertainty in the clustering. We elicited the following prior distribution:

$$\begin{aligned} Y_i|\mu_i, \sigma_i^2 &\sim N(\mu_i, \sigma_i^2), \\ \mu_i, \sigma_i^2 &\sim DP(\alpha G_0), \\ \alpha &\sim Gamma(1, 1), \\ G_0 &= NIG(\mu_0, \kappa_0, a, b), \end{aligned}$$

with $\mu_0 = \bar{y}$, $\kappa_0 = 0.1$, $a = 2$ and $b = 0.5s^2$, where $\bar{y}$ and s^2 denote the sample mean and variance respectively. After 10000 burnin iterations, 10000 iterations of the Gibbs sampler have been run using algorithm 8 of Neal (2000), as implemented by the R package `DPpackage` (Jara, 2007; Jara et al., 2011).

Figure 3b shows the posterior density estimate and the optimal partition, obtained by optimising (11). Figure 4 shows the heat map representation of the posterior similarity matrix and the graph representing the optimal partition which, accordingly with the estimates given by Roeder (1990) and Aitkin (2011) about the most likely clustering, is composed by three clusters. It is worth to notice that the optimal partition, obtrained by minimising (11), coincides with the unique partition, among the ones generated by the MCMC algorithm, which gives the minimum value of the code length. In order to quantify the uncertainty about the clustering, we examine the posterior estimate of the cumulative distribution function of the code length associated to an arbitrary partition of the sample items. Such an estimate is given by the empirical distribution function of the code lengths computed on the partitions generated by the MCMC and it is shown in Figure 5b. Figure 5a show how many times the equivalence classes of partitions correponding to distinct expected code lengths have been visited by the Gibbs sampler. In the same figure the posterior percentiles of $L(\mathcal{M}$ of order 5, 10, 25, 50, 75 and 95 are represented as red segments on the abscissae. From Table 1 we can notice that the number of clusters in partitions belonging to different equivalence classes does not decrease with respect to the expected code length. We can also observe that the number of clusters in partitions of decreasing quality (i.e. increasing code length) may keep locally constant. This means that our posterior uncertainty concerns both the number of clusters and each cluster composition. We can quantify our uncertainty about the clustering, in the sense that we can state what is the posterior probability that a clustering generated by our model gives an expected code length not greater than a given threshold. In our example, we can state that, a posteriori, the probability of partitions with code length not greater than 6.29 is 0.25. We can compare partitions corresponding to different expected code length posterior quantiles, as shown in Figure 6, which shows the representative elements of the equivalence classes corresponding to the posterior precentiles shown in Table 1. Each partition shown in that figure is compared with the optimal partition in terms of the Adjusted Rand Index (ARI). The ARI (Hubert and Arabie, 1985) measures the agreement between two partitions: the closer the index is to 1, the stronger the agreement between the two partitions, a value equal to 1 indicating perfect agreement. In this applicaton, we can then state that the posterior probability of partitions with expected code length ranging between 6.14 and 6.29 is 0.25, that the agreement of each such partition with the optimal one is negatively associated with the expected code length, and for partitions corresponding to the posterior percentiles of $L(\mathcal{M})$ of order ranging between 5 and 95 such agreement, measured by the ARI, decreases from 1 to 0.197.

Simulated data. We now present, via simulation, an example of unsupervised classification characterised by a very low posterior uncertainty. We generated a sample of size $n = 200$ from the following bivariate Gaussian mixture:

$$f(\mathbf{y}) = \sum_{j=1}^{4} \frac{1}{4} N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}), \tag{15}$$

with $\boldsymbol{\mu}_j = \left[2(-1)^{\lfloor (j-1)/2 \rfloor}, 2(-1)^{(j-1)}\right]^T$ and $\boldsymbol{\Sigma} = 0.04\mathbf{I}_2$. The data are shown in Figure 7a. The fitted model is

$$\begin{aligned} \mathbf{Y}_i &\sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \\ (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) &\sim DP(\alpha G_0), \\ G_0 &\sim NIW(\boldsymbol{\mu}_0, \kappa_0, \nu, \boldsymbol{\Psi}), \\ \alpha &\sim Ga(2, 1), \end{aligned}$$

with $\boldsymbol{\mu}_0 = \bar{\mathbf{y}}$, $\kappa_0 = 0.1$, $\nu = 6$ and $\boldsymbol{\Psi} = \frac{1}{n-1}(\mathbf{y} - \bar{\mathbf{y}})^T(\mathbf{y} - \bar{\mathbf{y}})$. We ran 10000 iterations of the Gibbs sampler after discarding the first 10000 iterations. The data are shown in Figure 7, with the contour levels of the posterior estimate of the joint probability density function superimposed. Figure 7 shows also the heat map representation of the posterior pairwise similarities, the graph representation of $\tilde{\mathcal{M}}$, which coincides with the partition determined by the labelling of the mixture components of (15), and the empirical distribution function of the expected code lengths associated to the partitions visited by the Gibbs sampler. From Figure 7d we can notice that they range between 6.4 and 6.64, about 80% of them attaining the minimum, which corresponds to $L(\tilde{\mathcal{M}})$. Figure 8 shows the partition corresponding to the 95th posterior percentile of $L(\mathcal{M})$, taking the value 6.45. It is composed by 6 clusters, two of which are the singletons represented by the black square and dark green diamond on the top right region of the scatterplot. The ARI index measuring the agreement between this partition and the partition induced by the mixture components of (15) takes value 0.987. This behaviour is an example of a general property of this clustering method: when the posterior uncertainty about clustering is low, the posterior distribution of $L(\mathcal{M})$ tends to be strongly concentrated around the lower bound of its support, and the agreement between partitions associated to different posterior quantiles tends to be strong.

4 Multivariate data and variable selection

When dealing with multivariate data, we often need to decide wether using the whole dataset or only a subset of variables better suited for clustering purposes. Two particularly interesting recent contributions are Raftery and Dean (2006) and Yau and Holmes (2011). The former attempts to identify the variables that best contribute to clustering the dataset by modelling jointly the variables, say $\mathbf{Y} = [Y_1, \dots, Y_p]^T$, and the membership labels, say $\mathbf{z} = [z_1, \dots, z_n]^T$. This is done by separating the subset of variables that depend on the cluster labels, say $\mathbf{Y}^c$, from the subset of variables that, conditionally on $\mathbf{Y}^c$, are independent of $\mathbf{z}$ through the factorisation of the joint probability density function:

$$f(\mathbf{Y}) = f(\mathbf{Y}^{nc}|\mathbf{Y}^c)f(\mathbf{Y}^c|\mathbf{z}). \tag{16}$$

Alternative models are estimated through maximum likelihood and compared via the BIC. Estimation and model comparison are carried out through an iterative algorithm. The subset of variables best suited for classification purposes is the $\mathbf{Y}^c$ corresponding to the joint model that minimises the BIC. Yau and Holmes (2011) propose an interesting hierarchical Bayesian nonparametric model based on infinite Gaussian mixture. The advantage of their proposal is that the relevance of each variable can be evaluated by a parameter that represents the variance of the normalised differences between the centers of mixture components. One limitation of their proposal is that it requires clusters to be identified by component means only, whereas variances do not play a crutial role.

Here we shall propose an algorithm similar, under some respects, to the one of Raftery and Dean (2006). The key idea is that dealing with a dataset containing measurements on p variables, we look for a subset of p^* ($p^* < p$) variables providing a clustering with the lowest expected code length among the ones provided by any other subset of variables, using the method illustrated in the previous section. Unfortunately, applying a selection method based on this principle, we found that often the clustering obtained in this way was characterised by the presence of a high number of very small clusters and that this inconvenience occurs more often when the number of variable increases. Therefore, we deemed useful to consider a criterion linked to the expected code length, with a correction term penalising for the presence of small clusters. Before cosidering such a criterion, we recall the concept of silhouette width and average silhouette width.

4.1 Penalised code length

Rousseeuw (1987) introduced the silhouette widths with the purpose of evaluating clustering validity and, possibly, determining the number of clusters. For any partition $\mathcal{M}$, let us define A as the cluster to which item i has been assigned, C any other cluster different from A, and $d(i, C)$ the average dissimilarity of i to all the elements of C. We can define the following quantities:

$$\begin{aligned}
a_i &= \text{average dissimilarity of } i \text{ to all other objects of } A, \\
b_i &= \min_{C \neq A} d(i, C), \\
s_i &= \frac{b_i - a_i}{\max\{a_i, b_i\}}.
\end{aligned}$$

The quantity s_i is the silhouette width of i. From the definition, it follows straightforwardly that $-1 \leq s_i \leq 1$. When s_i is close to 1, i is very similar to the elements in A and well separeted form the cluster containing the items less dissimilar to i but not included in A ($a_i \ll b_i$). When s_i is close to -1, i is much more similar to the items in a cluster C than to the elements in A ($a_i \gg b_i$), hence it is badly classified. When $s_i = 0$, i is not well classified, in the sense that it is equally separated from the elements in A and the elements in the closest cluster different form A ($a_i = b_i$). The average silhouette width, $\bar{s} = \frac{1}{n} \sum_{i=1}^{n} s_i$, $-1 \leq \bar{s} \leq 1$, can then be used to give an overall evaluation of a partition $\mathcal{M}$: the closer $\bar{s}$ is to one, the better the partition $\mathcal{M}$. The average silhouette width is often used in order to determine the number G of clusters. In order to avoid the number of clusters increasing significantly, it is customary to set $s(i) = 0$ whenever $A = \{i\}$.

Let $\mathbf{Y}^{(p)} = [Y_1, \ldots, Y_p]^T$ be a p-variate random vector, and assume a model like the one described in equations (3)-(6) has been fitted to the data. We can then define the posterior dissimilarity matrix as

$$\mathbf{D}\left(\mathbf{Y}^{(p)}\right) = \mathbf{1}_n - \mathbf{S}\left(\mathbf{Y}^{(p)}\right), \tag{17}$$

where $\mathbf{1}_n$ denots the n-dimensional square matrix with elements identically equal to 1 and $\mathbf{S}\left(\mathbf{Y}^{(p)}\right)$ is the posterior similarity matrix introduced in subsection 3.1. Let $\tilde{\mathcal{M}}\left(\mathbf{Y}^{(p)}\right)$ represent the optimal partition obtained as in (14), $A \in \tilde{\mathcal{M}}\left(\mathbf{Y}^{(p)}\right)$ be the cluster to which item i has been allocated and $C \in \tilde{\mathcal{M}}\left(\mathbf{Y}^{(p)}\right)$ any other cluster. We can then define

$$\begin{aligned}
a_i\left(\mathbf{Y}^{(p)}\right) &= \frac{1}{|A| - 1} \sum_{j \in A} \mathbf{D}_{ij}\left(\mathbf{Y}^{(p)}\right) \\
b_i\left(\mathbf{Y}^{(p)}\right) &= \min_{C \in \tilde{\mathcal{M}}\left(\mathbf{Y}^{(p)}\right), C \neq A} \frac{1}{|C|} \sum_{j \in C} \mathbf{D}_{ij}\left(\mathbf{Y}^{(p)}\right) \\
s_i\left(\mathbf{Y}^{(p)}\right) &= \frac{b_i\left(\mathbf{Y}^{(p)}\right) - a_i\left(\mathbf{Y}^{(p)}\right)}{\max\{a_i\left(\mathbf{Y}^{(p)}\right), b_i\left(\mathbf{Y}^{(p)}\right)\}},
\end{aligned}$$

with $s_i\left(\mathbf{Y}^{(p)}\right)$ representing the posterior shilouette width of i, conditionally on $\mathbf{Y}^{(p)}$. It follows straightforwardly the we can define the posterior average width as

$$\bar{s}\left(\mathbf{Y}^{(p)}\right) = \frac{1}{n} \sum_{i=1}^{n} s_i\left(\mathbf{Y}^{(p)}\right).$$

We can finally define a new criterion, the penalised expected code length, given by

$$\mathcal{L}\left(\mathbf{Y}^{(p)}\right) = L\left(\tilde{\mathcal{M}}\left(\mathbf{Y}^{(p)}\right)\right) - \bar{s}\left(\mathbf{Y}^{(p)}\right). \tag{18}$$

Clearly, when looking for a subset of $p*$ variables among the p available ones, under the perspective we have illustrated so far, we should choose that $\mathbf{Y}^{(p^*)}$ minimising (18).

4.2 The algorithm

The greedy search algorithm we propose works as follows.

Step 1 Produce a clustering from each variable, as in the previous section, and select the variable that provides the lowest code length partition. Name $\mathcal{L}^*$ the minimum penalised code length you found and $Y^{(1)}$ the selected variable.

Step 2 If the number of selected variables is $p' = 1$, form $p - p'$ pairs of variables by coupling each of the unselected variables with $Y^{(1)}$, and for each pair compute the corresponding optimal clustering and its associated code length. Name $\tilde{\mathcal{L}}$ the minimum penalised expected code length you found, $Y^{(2)}$ the variable that, correspondingly, you coupled with $Y^{(1)}$.

If $\tilde{\mathcal{L}} < \mathcal{L}^*$, select $\mathbf{Y}^{(2)} = [Y^{(1)}, Y^{(2)}]^T$, set $\mathcal{L}^* = \tilde{\mathcal{L}}$ and go to Step 3.

Else, return $Y^{(1)}$, $\mathcal{L}^*$, $p^* = 1$ and stop.

Step 3 Let the number of selected variables be p', $1 < p' < p$. Form $p - p'$ subsets of size $(p'+1)$ by joining each unselected variable to $\mathbf{Y}^{(p')}$, and for each such subset compute the corresponding optimal clustering and its associated penalised code length. Name $\tilde{\mathcal{L}}$ the minimum penalised expected code length you found and $Y^{(p'+1)}$ the variable that, correspondingly, you joined to $\mathbf{Y}^{(p')}$.

If $\tilde{\mathcal{L}} < \mathcal{L}^*$, select the subset $\mathbf{Y}^{(p'+1)} = \left[\mathbf{Y}^{(p')^T}, Y^{(p'+1)}\right]^T$ and set $\mathcal{L}^* = \tilde{\mathcal{L}}$. and go to Step 4.

Else, go to step 4.

Step 4 Let the number of selected variables be $p'+1$. For $j = 1, \ldots, p'$ remove $Y^{(j)}$ from $\mathbf{Y}^{(p'+1)}$. Name $\tilde{\mathcal{L}}$ the minimum penalised code length you found and $\mathbf{Y}^{(p')}$ the new subset of variables.

If $\tilde{\mathcal{L}} \leq \mathcal{L}^*$ and $p' \leq p-2$ set $\mathbf{Y}^{(p')}$ as the new subset of selected variables and $\mathcal{L}^* = \tilde{\mathcal{L}}$, then go to Step 3.

Else if $\tilde{\mathcal{L}} > \mathcal{L}^*$ and $p' \leq p - 2$ go to step Step 3.

Else if $\tilde{\mathcal{L}} \leq \mathcal{L}^*$ and $p' + 1 = p$, return $\mathbf{Y}^{(p'+1)}$, $\mathcal{L}^*$, $p^* = p$ and stop.

Step 5 Iterate steps 3 and 4 until no improvements are achieved in $\mathcal{L}^*$, in which case return p^*, $\mathbf{Y}^*$ and $\mathcal{L}^*$.

The algorithm we have just illustrated allows us to find a subset of $p^* \leq p$ variables that best suite for clustering purposes; we can also rank these varables in the order they have been selected, which corresponds to a decreasing clustering capability. Differently from Yau and Holmes (2011), the clustering does not necessarely depend on the location only. On the other hand, our selection method requires that at each step a model is fitted to each one of the subsets of variables that are considered, in order to estimate a distinct posterior pairwise similarity. Parallel computing can be implemented, but still the algorithm is quite expensive, An advantage with respect to a selection method like the one by Raftery and Dean (2006) is that at each iteration we need only to model the marginal distribution of some subsets of variables, but we can omit the estimation of a term like $f(\mathbf{Y}^{nc}|\mathbf{Y}^{c})$ in (16 with a relevant gain in computational efficiency. It is important to emphasise that this advatage can be achieved only if each of the submodels, for $1 \leq p' < p$ represents a marginalisation of the model built for the whole set of p variables after that $p - p'$ of them have been discarded, otherwise the submodels would be incoherent.

4.3 A simulation study

We applied the algorithm to 100 independent samples of size $n = 200$ from the six-variate random vector $\mathbf{Y}$ such that

$$\begin{aligned} Y_1 \sim& N(0, 25) \\ (Y_2, Y_3) \sim& \sum_{j=1}^{4} \frac{1}{4} N(\boldsymbol{\mu_j}, \boldsymbol{\Sigma}_j) \\ (Y_4, Y_5 | \boldsymbol{\Sigma}) \sim& N(\mathbf{0}, \boldsymbol{\Sigma}) \\ Y_6 | Y_1, Y_2, Y_3, Y_4, Y_5 \sim& N(1 + 3Y_2, 0.64) \end{aligned}$$

where $\boldsymbol{\mu}_j = \left[2(-1)^{\lfloor \frac{j-1}{2} \rfloor}, 2(-1)^{(j-1)}\right]^T$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_3 = \mathbf{I}_2$, $\boldsymbol{\Sigma}_2 = 0.5\mathbf{I}_2$, $\boldsymbol{\Sigma}_4 = 1.5\mathbf{I}_2$, $\boldsymbol{\Sigma} \sim$ IW$(6, 12\mathbf{I}_2)$ and Y_1, $[Y_2, Y_3]^T$ and $[Y_4, Y_5]^T$ are independent. Figure 9 shows one of these independent samples. Obviously, the variables that carry the whole information about the clustering are either Y_2 and Y_3, which are generated by a 4-component Gaussian mixture, or Y_3 and Y_6; the triple $[Y_2, Y_3, Y_6]^T$ would carry the same information, but one variable would be redundant, Y_2 and Y_6 being strongly correlated.

For any subset $\mathbf{Y}^{(p)}$ of variables, $1 \leq p \leq 6$, we defined the following model:

$$\begin{aligned} \mathbf{Y}_i^{(p)} | \boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)} &\overset{ind}{\sim} N(\boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)}), \\ \left(\boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)}\right) | G &\overset{iid}{\sim} G \\ G &\sim DP(\alpha, G_0^{(p)}), \\ G_0^{(p)} &= NIW(\bar{\mathbf{y}}^{(p)}, \kappa_0, \nu_0, \mathbf{S}^{(p)}), \end{aligned}$$

where $\bar{\mathbf{y}^{(p)}}$ and $\mathbf{S}^{(p)}$ denote the sample mean and variance-covariance matrix respectively, $k_0 = 0.5$ and $\nu_0 = 10$. It is worth to notice that, for $1 \leq p \leq 5$, $G_0^{(p)}$ represents the marginal distribution of $G^{6)}$ after the exclusion of $6 - p$ variables, and this implies that all the submodels difined for different values of p are coherent. For each simulated dataset we implemented the algorithm illustrated above after estimating the model using the Gibbs sampler (Neal, 2000) with 10000 iterations after a burn in period of 10000 iterations.

Table 2 shows that only in 4 cases 4 variables have been selected,whereas in 84 and 12 samples, 2 and 3 variables were selected respectively. From Table 3, we can notice that in 78 samples Y_2 and Y_3 were the first two selected variables and in 22 cases the first two variables in the ranking were Y_3 and Y_6, meaning that the ferst two selected variables have always been chosen among the ones that actually provide information about the underlying clustering. Table 4 shows the frequency distribution of the number of clusters identified by our method: in 87 out of 100 independent simulations the number of clusters is correctly estimated, in eleven cases it is overestimated (five clusters) and only in two cases it is underestimated (three clusters). Figure 10 reports the boxplots of the ARI computed over all the partitions produced by our method and sharing the same number of clusters. We compared partitions produced by our method and the one given by the allocation of the simulated data to the mixture components. As we can notice from the figure, when 4 clusters are identified, the ARI ranges between about 0.7 and 0.95 whereas when 5 clusters are identified the ARI ranges between 0.8 and 0.95, whereas a sensibly lower agreement is observed on the two clusterings that underestimate the number of groups.

4.4 Examples

In the following we shall present two applications of the algorithm we illustrated in section 4.2 on real data. For each subset of variables we define the following model:

$$\begin{aligned}
\mathbf{Y}_i^{(p)}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i &\overset{ind}{\sim} N(\boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)}), \\
\left(\boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)}\right)|G &\overset{iid}{\sim} G \\
G &\sim DP(\alpha, G_0^{(p)}), \\
G_0 &= NIW(\bar{\mathbf{y}}^{(p)}, \kappa_0, \nu_0, \mathbf{S}^{(p)}),
\end{aligned}$$

where $\bar{\mathbf{y}}^{(p)}$ and $\mathbf{S}^{(p)}$ denote the sample mean and variance-covariance matrix respectively, $k_0 = 0.5$ in both examples, whereas $\nu_0 = 17$ in the first examlple and $\nu_0 = 10$ in the second one.

Wine data. The Wine dataset is available in the package `pdfCluster` of R (Azzalini and Menardi, 2014). It consists of 178 measurements on 14 variables. The first variable is a categorical one, and it identifies three types of wine: Barolo, Grignolino and Barbera. The remaining variables measure the quantities of 13 constituents: Alcohol, Malic acid, Ash, Alcalinity, Magnesium, Phenols, Flavanoids, Nonflavanoids, Proanthocyanins, Color intensity, Hue, OD280.OD315Dilution, Proline. The latter variables are continuous and we used them in order to assess wether the method we propose is able to identify the different types of wine. Six covariates were selected, in the following order: Malic, Proline, Color, Flavanoids and Alcalinity. The optimal partition, $\tilde{\mathcal{M}}$, is obtained through (14) and it is composed by four clusters. The top block of Table 5 is the confusion matrix that compares $\tilde{\mathcal{M}}$ with the true item labelling. Barolo and Barbera are quite clearly identified, beeing the items of these classes allocated for the most part to clusters 2 and 4; the items labelled as Grignolino, instead, are allocated over the four clusters, mostly to cluster 1 and, in a lesser extent to cluster 4 (which comprises also 8 items labelled as Barolo). The ARI measuring the agreement between $\mathcal{M}$ and the true labelling is equal to 0.65. We considered also the partitions obtained by applying the VI method of Wade and Ghahramani (2018) and by minimising Binder's loss function (Binder, 1978), using the same subset of variables. The confusion matrices comparing these partitions with the true labelling are reported in the second and third blocks of Table 5 respectively. We notice that the ARI index computed on the optimal partition produced by the VI is equal to 0.66. The same value is recorded for the clustering obtained by the minimisation of

Binder's loss, but in this case the number of clusters increases to seven. For completness, we implemented also the variable selection method of Raftery and Dean (2006) using the R package `clustvarsel` (Scrucca and Raftery, 2018), ending up by chosing a subset variables that coincides only partially with the one chosen by our method. The variables selected now are: Malic, Proline, Flavanoids, Color, Dilution and Hue. Coherently with the selection method, we fitted on this set of variables a finite Gaussian mixture via maximum likelihood estimation. A clustering $\mathcal{M}_{MAP}$ has been obtained by the MAP method as in Fraley and Raftery (2002), using the R package `mclust`. The confusion matrix comparing this clustering with the true labelling of sample items is reported in the last block of Table 5. We still get a partition in four clusters, and again Barolo and Bardolino are satisfactorily identified, whereas Grignolino is splitted over different clusters (mainly over clusters 2 and 3). The ARI takes value 0.720, which represents a slight improvement with respect to $\tilde{\mathcal{M}}$. Table 6 reports the confusion matrix between $\tilde{\mathcal{M}}$ and $\mathcal{M}_{MAP}$, which shows how similar the two partitions are ($ARI = 0.85$).

Bank note data. This data set (Flury and Riedwyl, 1988) consists of the measurement of six variables made on 100 genuine and 100 counterfeit old Swiss bank notes. The variables are: bank note length near the top, diagonal length, top margin width, bottom margin width, left edge width and right edge width (all measurements are taken in millimeters). The data are available in the R package `mclust`. Minimising the penalised expected code length, we excluded the left and right edge width. The optimal partition consists of three clusters: the former one, with 99 elements, coincides substantially with the subset of genuine bank notes, whereas the subset of counterfeit bank notes is splitted between the second (with 85 elements) and the third cluster (with 15 items). Using the same variables, we computed the optimal partitions via the VI method and the minimisation of Binder's loss. The former coincides with the partition we have just described, whereas the latter differs only for the presence of an additional cluster containing one item. As in the previous example, we considered Raftery and Dean's selection method, obtaining a partition in four clusters which could not clearly identify the two subsets. All these results are reported in Table 7.

5 Discussion

In this paper we have proposed a Bayesian nonparametric clustering based on the representation of sample items as nodes of a weighted graph where edge weights are given by the posterior pairwise similarities and on the minimisation of the expected description length of a sutably defined random walk on such a graph. Exploiting the MCMC output, we have shown that it is possible to quantify our state of uncertainty about the clustering. Furthermore, we have defined a greedy search algorithm which, in a multivariate setting, allows us to select a subset of variables which is better suited to cluster the sample items. The results obtained on simulated and real data suggest that our proposal is competitive with other recently introduced model based clustering methods.

In the examples we gave, we always considered continuous variables on which we fitted Gaussian models with Dirichlet process priors. The base measure has always been defined as a normal inverse Wishart distribution, with some hyperparameters estimated by sample moments, under an empirical Bayes perspective. We made these choises just to simplify the presentation, but here we stress that none of these restrictions is required in order to implement our clustering method.

One last feature of our proposal is its applicability to Bayesian finite mixture models.

References

Aitkin, M. (2011). How many components in a finite mixture? In K. Mengersen, C. Robert, and M. Titterington (Eds.), *Mixtures: estimation and applications*, pp. 123–144. John Wiley & Sons Inc, Chichester.

Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian non parametric problems. *Annals of Statistics. 2*, 1152–1174.

Azzalini, A. and G. Menardi (2014). Clustering via Nonparametric Density Estimation: The R Package pdfCluster. *Journal of Statistical Software 57*(11), 1–26.

Binder, D. A. (1978). Bayesian cluster analysis. *Biometrika 65*(1), 31–38.

Blackwell, D. and J. B. MacQueen (1973, 03). Ferguson Distributions Via Pólya Urn Schemes. *Ann. Statist. 1*(2), 353–355.

Blondel, V., J. Guillaume, R. Lambiotte, and E. Lefebvre (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment 2008*, P10008,.

Csardi, G. and T. Nepusz (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.

Dahl, D. B. (2009). Modal Clustering in a Class of Product Partition Models. *Bayesian Analysis 4*, 243–264.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The annals of statistics 1*(2), 209–230.

Flury, B. and H. Riedwyl (1988). *Multivariate Statistics: A practical approach.* London: Chapman & Hall.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports 486*(3), 75 – 174.

Fraley, C. and A. Raftery (2002). Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association 97*, 611–631.

Fritsch, A. and K. Ickstadt (2009, 06). Improved criteria for clustering based on the posterior similarity matrix. *Bayesian Anal. 4*(2), 367–391.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models.* Berlin: Springer.

Frühwirth-Schnatter, S., B. Grün, , and G. Malsiner-Walli (2018). Comment on the paper by Wade and Ghahramani. *Bayesian Analysis 13*(2), 601–603.

Gordon, A. (1999). *Classification* (2nd ed.). Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press.

Griffin, J. E. and S. G. Walker (2011). Posterior Simulation of Normalized Random Measure Mixtures. *Journal of Computational and Graphical Statistics 20*(1), 241–259.

Hartigan, J. A. (1975). *Clustering Algorithms*. New York, NY, USA: John Wiley & Sons, Inc.

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification 2*(1), 193–218.

Ishwaran, H. and L. F. James (2001). Gibbs Sampling Methods for Stick Breaking Priors. *Journal of the American Statistical Association 96*, 161–173.

Jain, S. and R. M. Neal (2007, 09). Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Anal. 2*(3), 445–472.

Jara, A. (2007). Applied Bayesian Non- and Semi-parametric Inference Using DPpackage. *R News 7*(3), 17–26.

Jara, A., T. Hanson, F. Quintana, P. Müller, and G. Rosner (2011). DPpackage: Bayesian Semi- and Nonparametric Modeling in R. *Journal of Statistical Software 40*(5), 1–30.

Kaufman, L. and P. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.

Lau, J. W. and P. J. Green (2007). Bayesian Model-Based Clustering Procedures. *Journal of Computational and Graphical Statistics 16*(3), 526–558.

Lijoi, A. and I. Prünster (2010). Models beyond the Dirichlet process. In N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker (Eds.), *Bayesian Nonparametrics*, Cambridge Series in Statistical and Probabilistic Mathematics, pp. 80–136. Cambridge University Press.

Lovász, L. (1996). Random Walks on Graphs: A Survey. In D. Miklós, V. T. Sós, and T. Szőnyi (Eds.), *Combinatorics, Paul Erdős is Eighty*, Volume 2, pp. 353–398. Budapest: János Bolyai Mathematical Society.

McLachlan, G. and D. Peel (2000). *Finite Mixture Models*. New York: Wiley.

Medvedovic, M. and J. Guo (2004). Bayesian Model-Averaging in Unsupervised Learning From Microarray Data. In *BIOKDD*, pp. 40–47.

Medvedovic, M. and S. Sivaganesan (2002). Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics 18*, 1194–1206.

Meilă, M. (2007). Comparing clusterings — an information based distance. *Journal of Multivariate Analysis 98*(5), 873 – 895.

Neal, R. M. (2000). Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics 9*(2), 249–265.

Papaspiliopoulos, O. and G. O. Roberts (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika 95*(1), 169–186.

Raftery, A. E. and N. Dean (2006). Variable Selection for Model-Based Clustering. *Journal of the American Statistical Association 101*(473), 168–178.

Roeder, K. (1990). Density Estimation With Confidence Sets Exemplified by Superclusters and Voids in the Galaxies. *Journal of the American Statistical Association 85*(411), 617–624.

Rosvall, M., D. Axelsson, and C. Bergstrom (2009). The map equation. *Eur. Phys. J. Special Topics 178*, 13–23.

Rosvall, M. and C. Bergstrom (2008). Maps of random walks on complex networks reveal community structure. *PNAS. 105*, 1118–1123.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*, 53 – 65.

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: Clustering, Classification and Censity Estimation Using Gaussian Finite Mixture Models. *The R Journal 8*(1), 205–233.

Scrucca, L. and A. E. Raftery (2018). clustvarsel: A Package Implementing Variable Selection for Gaussian Model-Based Clustering in R. *Journal of Statistical Software 84*(1), 1–28.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica 4*, 639–650.

Tyron, R. C. (1939). *Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality.* Edwards Brothers.

Wade, S. and Z. Ghahramani (2018). Bayesian Cluster Analysis: Point Estimation and Credible Balls (with Discussion). *Bayesian Analysis 13*(2), 559–626.

Walker, S. G. (2007). Sampling the Dirichlet Mixture Model with Slices. *Communications in Statistics - Simulation and Computation 36*(1), 45–54.

Yau, C. and C. Holmes (2011, 06). Hierarchical Bayesian nonparametric mixture models for clustering with variable relevance determination. *Bayesian Anal. 6*(2), 329–351.

Zubin, J. (1938). A technique for measuring like-mindedness. *The Journal of Abnormal and Social Psychology 33*(4), 508–516.

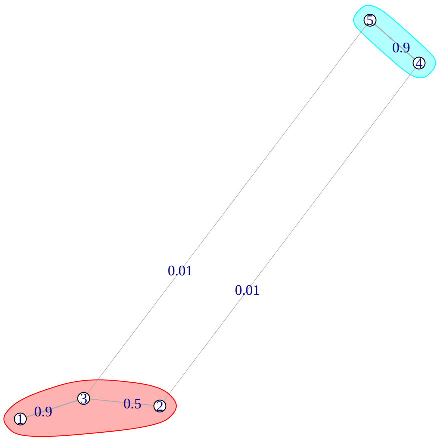
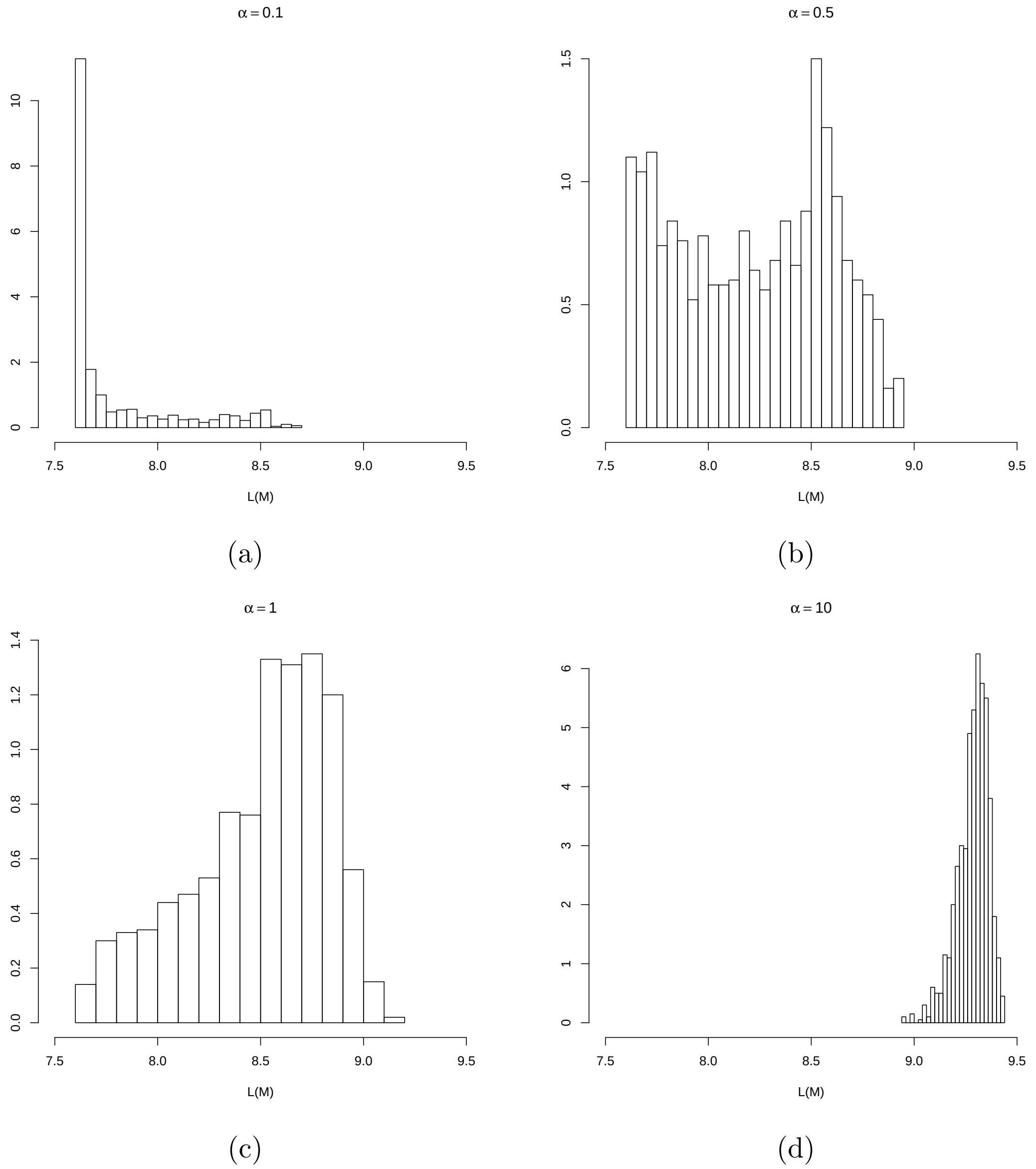


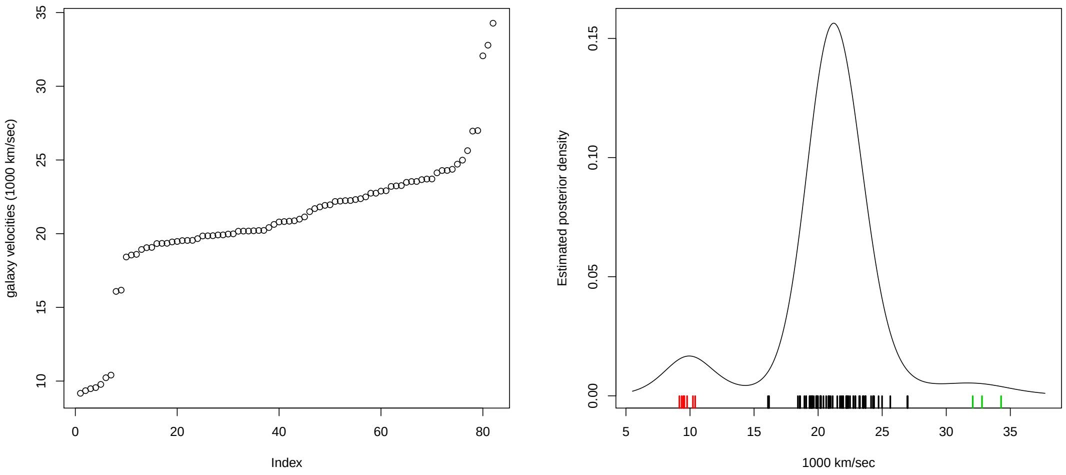
Figure 1: Representation of the graph associated to the adjacency matrix in (2); $\mathcal{M}^*$ is given by the highlighted clusters.



(a) (b)

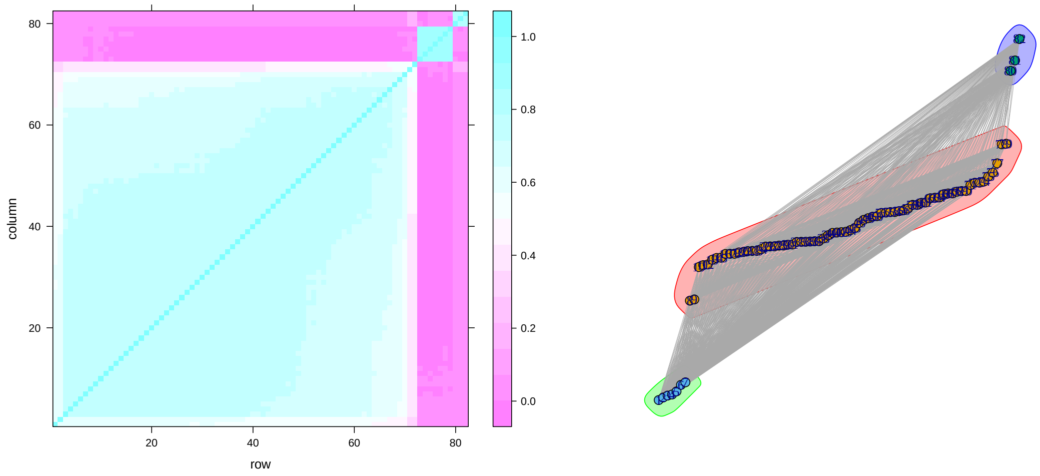
(c) (d)

Figure 2: Histograms of the values taken by $L(\mathbf{M})$ on 1000 random partitions of samples of size $n = 200$ generated by a Dirichlet process for different values of the concentration parameter α



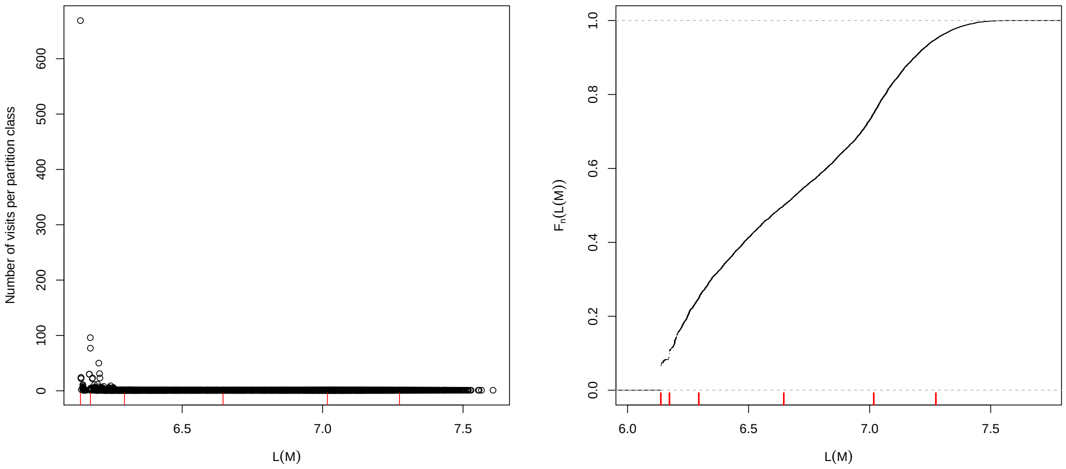
(a) (b)

Figure 3: (a) Galaxy data; (b) posterior estimate of the probability density function with observations on the abscissae, different colours representing distinct clusters in the optimal partition.



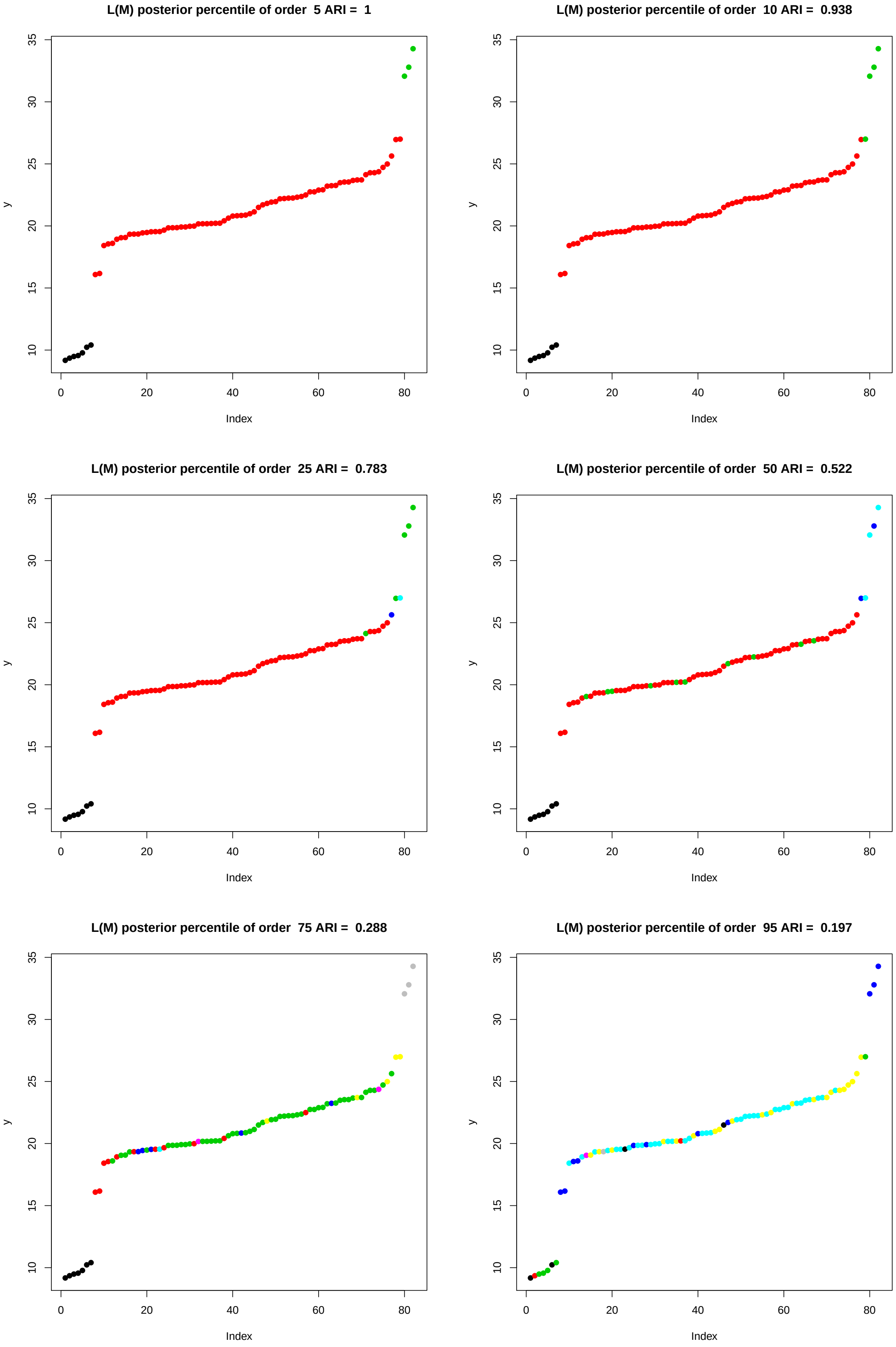
(a) (b)

Figure 4: (a) Heat map representation of the posterior similarity matrix; (b) Graph representation of the optimal partition.



(a) (b)

Figure 5: (a) Number of visits of each equivalence class in the partition space. (b) Posterior estimate of the cumulative distribution function of $L(M)$. The red segments on the abscissae represent the posterior percentiles of $L(M)$ of order 5, 10, 25, 5, 75 and 95.



L(M) posterior percentile of order 5 ARI = 1

L(M) posterior percentile of order 10 ARI = 0.938

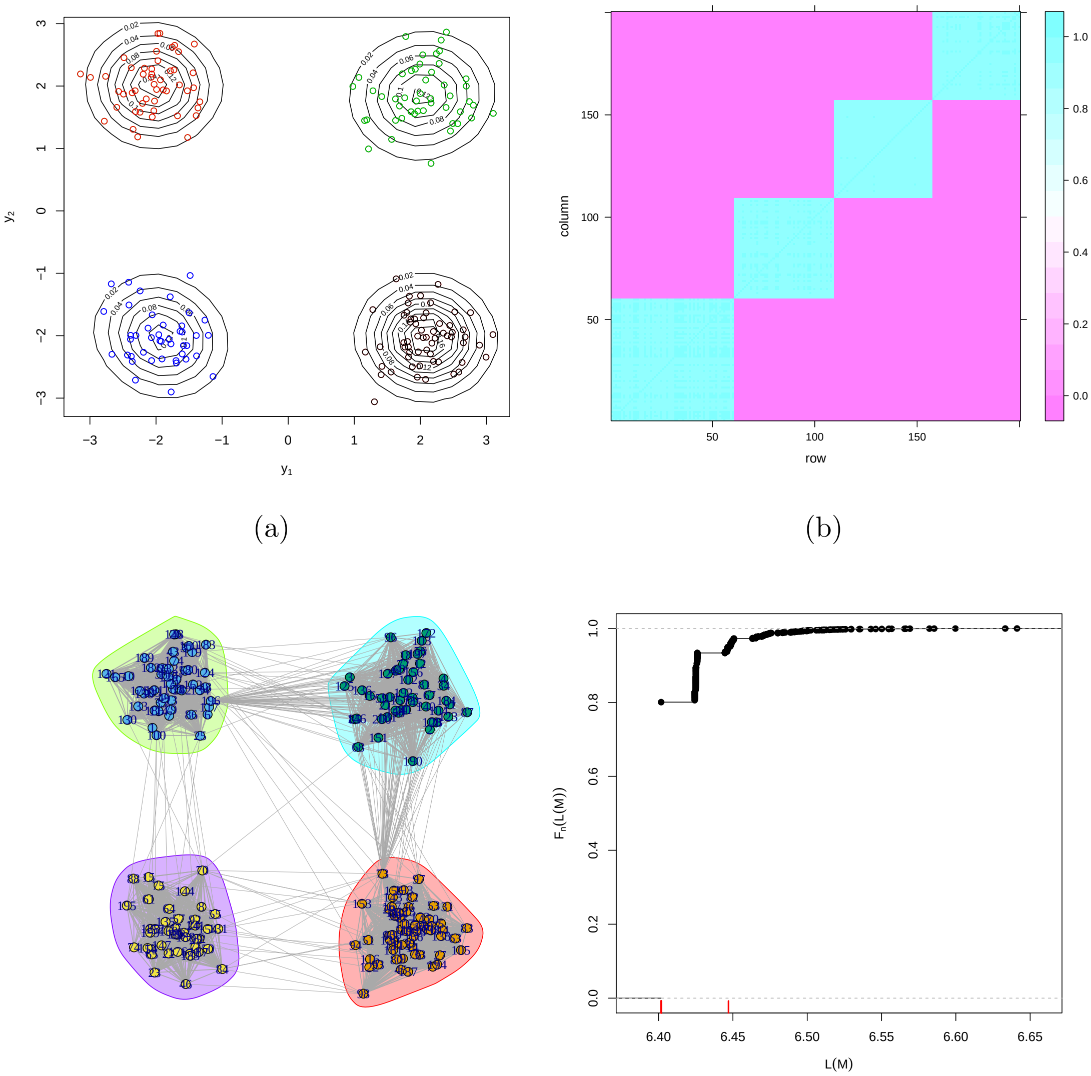
L(M) posterior percentile of order 25 ARI = 0.783

L(M) posterior percentile of order 50 ARI = 0.522

L(M) posterior percentile of order 75 ARI = 0.288

L(M) posterior percentile of order 95 ARI = 0.197

Figure 6: Partitions corresponding to the 5th, 10th, 25th, 50th, 75th and 95th posterior percentiles of $L(\mathcal{M})$. Clusters are identified by different colours. For each partition the ARI measures the agreement between the partition itself and $\tilde{\mathcal{M}}$.



(a)

(b)

(c)

(d)

Figure 7: (a) Simulated data discussed in Section 3.4 with posterior density estimate contour levels superimposed. (b) Heat map representation of the pairwise posterior similarity. (c) Graph representation of $\tilde{\mathcal{M}}$. (d) Empirical cumulative distribution function of the values of $L(\mathcal{M})$ sampled from the posterior distribution via Gibbs sampling.

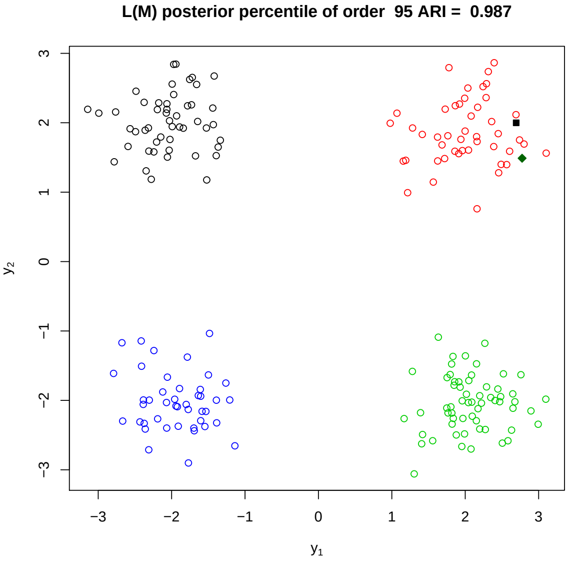


Figure 8: Partition corresponding to the posterior 95th percentile of $L(\mathcal{M})$ in the simulation discussed in Section 3.4. Clusters are identified by different colours, the black square and the green diamond on the top right side represent two singleton clusters.

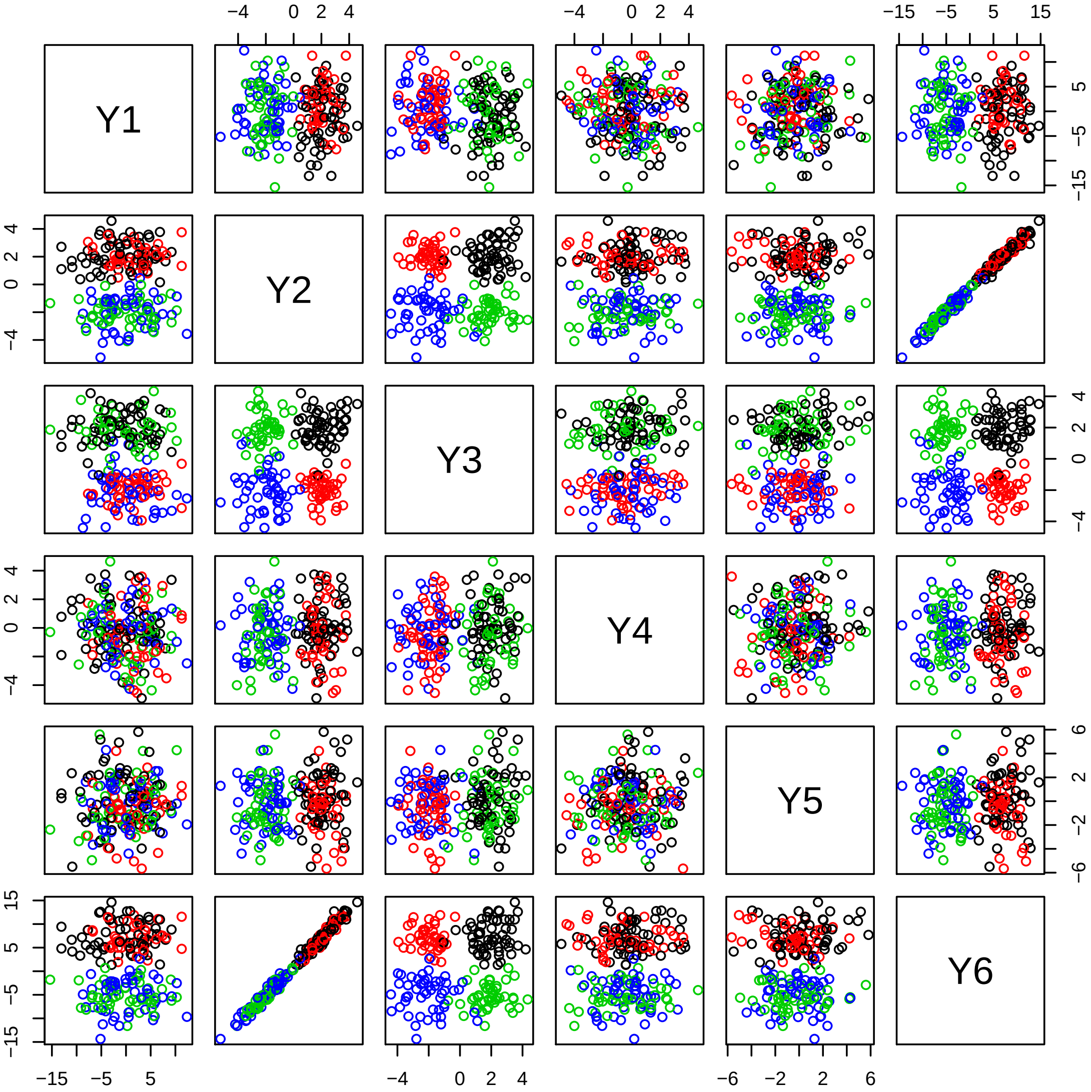


Figure 9: A simulated dataset

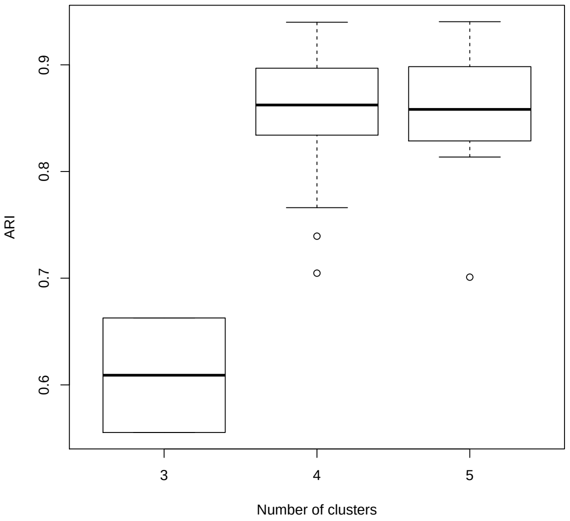


Figure 10: Boxplot of the ARI conditioned on the number of identified clusters.

	5	10	25	50	75	95
Code length percentile	6.14	6.17	6.29	6.65	7.02	7.27
Number of groups	3.00	3.00	5.00	5.00	8.00	12.00

Table 1: Number of groups corresponding to different code length percentiles.

Number of selected variables	2	3	4
Frequency	84	12	4

Table 2: Frequency of the number of selected variables in the simulation study of section 4.3.

	Y_2	Y_3	Y_6
Y_2	0	45	0
Y_3	33	0	5
Y_6	0	17	0

Table 3: Frequency of pairs of variables appearing as first (rows) and second (columns) selected ones in the simulation study of section 4.3.

Number of identified clusters	3	4	5
Frequency	2	87	11

Table 4: Frequency distribution of the number of identified clusters in the simulation study of section 4.3.

		Barolo	Grignolino	Barbera
$\tilde{\mathcal{M}}$ $ARI = 0.65$	1	50	3	0
	2	8	13	0
	3	1	49	1
	4	0	6	47
Wade and Ghahramani (2018) $ARI = 0.66$	1	50	3	0
	2	8	14	0
	3	1	49	1
	4	0	5	47
Binder (1978) $ARI = 0.66$	1	50	3	0
	2	1	13	0
	3	7	0	0
	4	1	48	1
	5	0	5	47
	6	0	1	0
	7	0	1	0
Raftery and Dean (2006) Scrucca et al. (2016) $ARI = 0.72$	1	51	3	0
	2	0	51	1
	3	8	17	0
	4	0	0	47

Table 5: Comparison of the clusterings produced by the different model based classification methods applied on the wine data

		MAP			
		1	2	3	4
$\tilde{\mathcal{M}}$	1	52	1	0	0
	2	0	2	19	0
	3	2	48	1	0
	4	0	1	5	47

Table 6: Comparison of the partitions obtained by our method (rows) and tha MAP (columns).

		Counterfeit	Genuine
$\tilde{\mathcal{M}}$ $ARI = 0.648$	1	0	99
	2	85	0
	3	15	1
Wade and Ghahramani (2018) $ARI = 0.648$	1	0	99
	2	15	1
	3	85	0
Binder (1978) $ARI = 0.648$	1	0	99
	2	15	1
	3	84	0
	X4	1	0
Raftery and Dean (2006) Scrucca et al. (2016) $ARI = 0.720$	1	1	20
	2	0	79
	3	15	1
	4	84	0

Table 7: Comparison