High-Dimensional Clustering via Random Projections

(Article begins on next page)

30 April 2024

High-dimensional clustering via Random Projections

Laura Anderlucci[*1], Francesca Fortunato[1], and Angela Montanari[1]

[1]*Department of Statistical Sciences, University of Bologna, Italy.*

Abstract

This work addresses the unsupervised classification issue for high-dimensional data by exploiting the general idea of Random Projection Ensemble. Specifically, we propose to generate a set of low-dimensional independent random projections and to perform model-based clustering on each of them. The top B^* projections, i.e., the projections which show the best grouping structure, are then retained. The final partition is obtained by aggregating the clusters found in the projections via consensus. The performances of the method are assessed on both real and simulated datasets. The obtained results suggest that the proposal represents a promising tool for high-dimensional clustering. ***Keywords:*** High-dimensional clustering; random projections; model-based clustering

1 Introduction

Data clustering plays a key role in modern statistics as it represents one of the most effective tools to understand the structure underlying a given data set. The aim of clustering is essentially to categorize data into 'clusters' (or groups) such that observations belonging to the same cluster are more similar to each others than those in different groups (see, for recent comprehensive reviews, Xu and Tian, 2015; Hennig et al., 2015).

Many different approaches have been proposed to address this issue. In this paper, we focus on model-based ones, see Bouveyron and Brunet-Saumard (2014), McNicholas (2016) and McLachlan et al. (2019) for reviews on this topic and Bouveyron et al. (2019) for a recent book that covers topics on model-based clustering and classifications. Most popular model-based clustering methods are based on the Gaussian mixture model (GMM) (see, among others, Banfield and Raftery, 1993; McLachlan and Peel, 2000).

Clustering in low-dimensional spaces requires limited resources; the complexity of the problem indeed increases with the number of observed features, p. When dealing with high-dimensional data, the use of traditional unsupervised classification algorithms faces several limitations; in particular, the presence of noisy or irrelevant information

---

[*]mailto: `laura.anderlucci@unibo.it`

can mislead these methods due to the '*curse of dimensionality*', as coined by Bellman (1957). In order to overcome this problem, often dimension reduction procedures are applied before carrying out any clustering.

Generally, the term 'dimension reduction' refers to two different approaches; namely, it includes both *feature selection* methods that embed the high-dimensional points into a lower subspace by selecting some 'relevant' variables, and *feature extraction* algorithms which find an embedding by constructing new artificial features that are, for example, linear combinations of the original ones. Variable selection strategies have been frequently used to handle high-dimensional clustering issues, but feature extraction procedures could be generally more efficient. Feature selection techniques indeed may discard some potentially important variables, e.g. variables that are not predictive if individually considered, but that could provide significant benefits when taken in conjunction with other features.

Traditionally, variable combination methods involve the projection of high-dimensional data onto a lower subspace with the intent of capturing as much of the data variability as possible (e.g. Principal Component Analysis). Albeit this approach has been successfully used in many applications, its aim does not always coincide with that of a clustering task. In fact, the useful information about the group structure is not necessarily contained in the subspaces with the largest variance, as exposed by Chang (1983).

A recent approach for dimension reduction that has been gaining increasing attention is based on Random Projections (RPs, Vempala, 2004) and consists in mapping at random the original high-dimensional data onto a lower subspace by using a random matrix with orthogonal columns of unit length. Specifically, the key point of RP is that, regardless of the original data dimension, the final solution still preserves the global linear information almost perfectly. Such a result is guaranteed by the Johnson-Lindenstrauss (JL) Lemma (Johnson and Lindenstrauss, 1984), which states that any n-point set in p dimensions ($\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n]^\top$, $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$) can be linearly projected onto $d = O(\log(n)/\varepsilon^2)$ coordinates (with $d \ll p$), by using a random matrix $\mathbf{A} \in \mathbb{R}^{p \times d}$ with orthonormal columns, while preserving pairwise distances within a factor $1 \pm \varepsilon$, $\varepsilon \in (0, 1)$. More precisely, with high probability over the randomness of $\mathbf{A}$:

$$(1 - \varepsilon)||\mathbf{x}_i - \mathbf{x}_j||_2 \leq ||\mathbf{A}^\top \mathbf{x}_i - \mathbf{A}^\top \mathbf{x}_j||_2 \leq (1 + \varepsilon)||\mathbf{x}_i - \mathbf{x}_j||_2, \quad i, j = 1, \ldots, n \quad (1)$$

where $|| \cdot ||_2$ indicates the L_2 norm.

RPs are a particularly useful tool in high-dimensional settings and their application covers multiple areas of multivariate analysis (for a recent and detailed review see Cannings, 2021), from supervised classification (e.g., Cannings and Samworth, 2017; Durrant and Kabán, 2015) to linear regression (see, e.g., Ahfock et al., 2020; Slawski et al., 2018; Thanei et al., 2017), to sparse principal component analysis (Gataric et al., 2020), to dimension reduction (Bingham and Mannila, 2001) and to clustering (Dasgupta, 1999; Fern and Brodley, 2003; Boutsidis et al., 2010).

Bhattacharya et al. (2009) proved that also the Hellinger distance between any two

distributions P and Q, defined as

$$H(P,Q) = \frac{1}{2}\left(\|\sqrt{P} - \sqrt{Q}\|_2\right)^2,$$

admits a low distortion JL-type embedding. In the model-based clustering context, where data are considered as coming from a distribution that is a mixture of two or more components, this theorem directly implies that the distance between the density of any pair of components is preserved with arbitrarily small distortion. In other words, it states that if two component densities are sufficiently far apart in the high-dimensional space, then they are expected to be approximately the same also in the reduced d-dimensional space.

These interesting results motivated us to employ random projections within a model-based clustering framework. Specifically, inspired by the original idea of Cannings and Samworth (2017) for supervised classification, we propose to generate a set of B low dimensional independent random projections and to apply a Gaussian mixture model on each of them. Our Random Projection Ensemble Clustering (RPEClu) algorithm then obtains the final partition by combining via consensus the clustering results from the top B^* projections, i.e., the projections which show the best grouping structure according to a given criterion.

Other approaches have been proposed in the literature on model-based clustering with the aim of reducing the dimensionality; those based on on mixtures of factor analyzers have turned out to provide particularly good results in terms of dimension reduction (see, among others, Ghahramani and Hinton, 1997; McLachlan et al., 2003; Montanari and Viroli, 2011, and references therein). They assume that, within each mixture component, the data are generated according to a linear factor model and resort to the Expectation-Maximization algorithm (EM, Dempster et al., 1977) for parameter estimation.

McNicholas and Murphy (2008) developed a family of parsimonious Gaussian mixture models (PGMMs) for clustering that allows for constraints upon the component covariance structure in the most general mixture of factor analyzers model. Bouveyron et al. (2007) proposed a GMM which takes into account the specific subspace around which each cluster is located, therefore limiting the number of parameters to estimate. The parameter estimation is carried out via EM algorithm and the intrinsic dimension of each group is determined automatically thanks to either the Bayesian Information Criterion (BIC) or Cattell's scree-test (Cattell, 1966). In Sections 5 and 6 we compare our approach to Bouveyron's one.

The paper is organized as follows. Section 2 recalls the model-based clustering framework. In the same section, some popular dimension reduction procedures for high-dimensional clustering are briefly presented. In Section 3, the RPEClu algorithm is introduced and defined in detail. Section 4 is devoted to practical considerations about the computational complexity of the algorithm, the choice of the number of random projections and the dimension of the projected space. Section 5 presents a simulation study where the proposed methodology is compared with some benchmark clustering techniques. In Section 6, RPEClu is applied to two sets of high-dimensional real data. Section 7 explores some possible options to solve the problem of estimating the number

of clusters when G is not known in advance. A final discussion on the obtained results concludes the paper.

2 High-dimensional model-based clustering

In model-based clustering (Banfield and Raftery, 1993; McLachlan and Peel, 2000), data are assumed to derive from a common source with G different sub-populations. In particular, each sub-population is modelled separately (typically by members of the same parametric density family) and the overall population is but a mixture of them. The resulting model is a finite mixture and it is described by the following probability density function (pdf):

$$f(\mathbf{x}) = \sum_{k=1}^{G} \pi_k f_k(\mathbf{x}|\theta_k).$$

Here, f_k and θ_k are the density and the parameters of the k-th component of the mixture, respectively, whereas π_k is the prior probability that an observation belongs to the k-th component ($\pi_k > 0$, $\sum_{k=1}^{G} \pi_k = 1$). For clustering purposes, units are allocated to the component whose posterior probability is maximum.

A common choice for $f_k(\cdot)$ is the multivariate normal distribution, $\phi_k(\cdot)$, parameterized by its mean vector μ_k and its covariance matrix Σ_k:

$$\phi_k(\mathbf{x}|\mu_k, \Sigma_k) = (2\pi)^{-(p/2)} |\Sigma_k|^{-(1/2)} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \mu_k)^\top \Sigma_k^{-1} (\mathbf{x} - \mu_k) \right\}.$$

Following this approach, the entire data set is modeled by a Gaussian mixture model:

$$f(\mathbf{x}) = \sum_{k=1}^{G} \pi_k \phi_k(\mathbf{x}|\mu_k, \Sigma_k).$$

In presence of high-dimensional data and limited sample size the GMM tends to perform poorly. In fact, the number of parameters increases quadratically with p, thus making the maximum-likelihood estimation problem ill-posed very quickly. The earliest approaches, which have appeared in the literature to overcome this limit and to attain parsimony, propose alternative parameterizations of the component densities. For instance, Banfield and Raftery (1993) and Celeux and Govaert (1995) introduce a parsimonious parameterizations of the covariance matrix in terms of its eigenvalue decomposition so as to control the volume, shape and orientation of the Gaussian ellipsoids. Biernacki and Lourme (2014) define different parsimonious models based on a variance-correlation decomposition of the covariance matrices.

When performing variable selection for clustering, the aim is essentially to identify those features that bring relevant information about the underlying group structure. In the model-based context, the definition of 'relevance' should be expressed in terms of probabilistic dependence (or independence) with respect to $\mathbf{Z}$, i.e., the random vector which describes the latent class membership ($\mathbf{Z} = [Z_1, \ldots, Z_k, \ldots, Z_G]^\top$, such that $Z_k = 1$ if $\mathbf{x}$ arises from the kth subpopulation, 0 otherwise, $j = 1, \ldots, k-1, k+1, \ldots, G$).

4

Specifically, the distribution of relevant variables directly depends on $\mathbf{Z}$ as these features contain the key clustering information. Conversely, both redundant and uninformative variables do not provide any additional nor useful information and, thus, they can be assumed to be conditionally independent given the relevant variables or completely independent of the group structure, respectively. Following this approach, several authors have recast the variable selection problem for clustering in a model selection one. Namely, relevant variables are sought through a stepwise procedure that, at each step, compares models that differ in the role assigned to the variables in explaining the clustering structure.

Pioneers of this framework were Raftery and Dean (2006), who introduced a procedure in which the decision for inclusion or exclusion of a generic (set of) variable(s) $\mathbf{x}^P$ into the current set of clustering ones $\mathbf{x}^C$ is taken by comparing two competing models in terms of their BIC, defined as

$$BIC = 2\ell - q\log n,$$

where ℓ is the log-likelihood of the model, q is the number of estimated parameters and n the number of observations. In particular, Model I assumes that $\mathbf{x}^P$ carries relevant information about the cluster membership, whereas Model II states that $\mathbf{x}^P$ does not depend on $\mathbf{Z}$. The BIC associated to these models are:

$$\begin{aligned} \text{BIC}_I &= \text{BIC}_{\text{clust}}(\mathbf{x}^C, \mathbf{x}^P) \\ \text{BIC}_{II} &= \text{BIC}_{\text{clust}}(\mathbf{x}^C) + \text{BIC}_{\text{reg}}(\mathbf{x}^P|\mathbf{x}^C) \end{aligned} \tag{2}$$

Here, $\text{BIC}_{\text{clust}}(\mathbf{x}^C, \mathbf{x}^P)$ is the BIC of the GMM in which $\mathbf{x}^P$ adds useful information, $\text{BIC}_{\text{clust}}(\mathbf{x}^C)$ is the BIC of the GMM on the current set of clustering variables only and $\text{BIC}_{\text{reg}}(\mathbf{x}^P|\mathbf{x}^C)$ is the BIC of the regression of $\mathbf{x}^P$ on $\mathbf{x}^C$. If $\text{BIC}_I - \text{BIC}_{II} > 0$, then $\mathbf{x}^P$ is added to the set of clustering variables $\mathbf{x}^C$.

This method has been further improved by Maugis et al. (2009a) and Maugis et al. (2009b) under the assumption that the irrelevant variables can be independent of some relevant ones.

Recently, two further extensions of the above modeling appeared in the literature: Scrucca (2016) suggests to overcome the sub-optimality of a stepwise model search by employing genetic algorithms; Galimberti et al. (2018) take into account the possibility that different variable vectors provide information about different clustering structures.

Although effective in many applications, in the unsupervised classification context the variable selection problem is ill-posed: clusters indeed strongly depend on the selected features and the features are selected according to the clusters (see Ruiz et al., 2009). For this reason, feature extraction procedures would rather be preferred.

Dasgupta (Dasgupta, 1999, 2000) demonstrated that RPs can be successfully used to handle high-dimensional clustering issues with a model-based approach. Firstly, by introducing the definition of *c-separation* of two multivariate Gaussian populations, he showed that a mixture of G Gaussians can be embedded onto just $O(\log G)$ random coordinates while preserving such separation with probability $1 - \delta$, $\delta \in (0, 1)$. The idea is to guarantee that Gaussians which are well-separated in $\mathbb{R}^p$ remain so when they

are projected in $\mathbb{R}^d$, by preserving the $O(G^2)$ line segments joining pairs of Gaussian centers in the original space. In order to satisfy such definition of separation, the largest eigenvalue λ_{max} of the component covariance matrices Σ_k in the reduced space, $k = 1, \ldots, G$, has to be smaller than or equal to the largest eigenvalue λ_{max}^* of the corresponding component covariance matrix Σ_k^* in $\mathbb{R}^p$, $k = 1, \ldots, G$. Second, he proved that even if the original Gaussians exhibit eccentric elliptical contours, their projected counterparts are always more spherical. These two benefits are of major importance and they definitely facilitate the learning of a Gaussian mixture model. In particular, dimension reduction saves a lot of time and computational costs on one hand; on the other, clusters of low eccentricity reduce the EM algorithmic challenges ensuring that, at each step, the estimated covariance matrices are not singular.

3 Random projection ensemble clustering

As discussed in the previous section, high-dimensional data pose many challenges to model-based clustering. Methods in this class indeed become rapidly over-parameterized since the number of parameters to estimate increases quadratically with the number p of observed features.

Random projections have shown to provide promising results for the analysis of high-dimensional data. Boutsidis et al. (2010) propose to use Random Projections with K-means clustering; specifically, they prove that data can be projected into a t-dimensional space, with $t = \Omega(K/\varepsilon^2)$, for any $\varepsilon \in (0, 1/3)$, such that the optimal K-partition of the point set is preserved within a factor of $2 + \varepsilon$ with constant probability. However, the main inconvenience of RPs is that they are highly unstable: namely, different random projections of the original data may provide completely different classification results. That is the reason why most of the successful proposals on RPs resort to ensembles. For example, Fern and Brodley (2003) propose to aggregate the clustering results of a GMM on different random projections of the data into a similarity matrix containing the probability "estimates" that any two data points belong to the same cluster; then, they suggest to perform an agglomerative clustering procedure on such a matrix to produce the final groups.

In this paper, we also exploit the general idea of RP ensemble for high-dimensional clustering. In particular, our novel proposal consists of applying a Gaussian mixture model to *carefully chosen* random projections of the original data, but differently from Fern and Brodley, we use the GMM properties for both projection selection and consensus aggregation.

3.1 On the choice of random projections

Differently from other transformation techniques (such as, for example, principal components or projection pursuit), the random projection method does not exploit any 'interestingness' criterion to identify the 'optimal' projection. High-dimensional data are just embedded into a lower dimensional subspace by using a random projection matrix A with orthogonal and unit length columns (e.g. generated according to Haar measure, Haar, 1933). As a consequence of that, results from distinct configurations of

the same data can be even dramatically different: some projections indeed can highlight a clear group structure in the lowered data, whilst some others can derail any hope of learning by confusing all the groups together.

In this section, we propose a method for choosing a number B^* of 'good' random projections, that is, a criterion for identifying those projections showing a clear group structure.

Hennig (2019) provides a detailed review of the validation indexes proposed in the literature to evaluate the quality of a clustering procedure. Although effective, many of these indexes rely on a measure of distance/dissimilarity and, therefore, they may seem inconsistent with a model-based framework. Furthermore, since in the unsupervised context no *apriori* information about the structure being looked for is available, we believe it makes sense to consider the RP selection as a part of the clustering algorithm, i.e., as the choice of the model that best fits the data according to a specific criterion (e.g. the BIC).

The BICs of mixture models fitted to different random projections cannot in principle be compared, because they are referred to different variables generated by the different random projections. On the contrary, the BIC of different models defined in the original variable space can be compared. We search for the solution that maximizes the log-likelihood of the GMM fitted on the original data, penalized by the number of free parameters.

In practice, in order to avoid the drawbacks associated with the high-dimensional spaces, a feasible solution consists in considering the following variable partition

$$\mathbf{Y}^* = [\mathbf{Y}, \bar{\mathbf{Y}}] = [\mathbf{XA}, \mathbf{X}\bar{\mathbf{A}}],$$

where $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the original high-dimensional data matrix, $\mathbf{A} \in \mathbb{R}^{p \times d}$ is the random projection matrix and $\bar{\mathbf{A}} \in \mathbb{R}^{p \times (p-d)}$ is its orthogonal complement, that can be easily computed via QR decomposition (see, e.g., Golub and Van Loan, 1996). The basic idea is to perform model-based clustering on the reduced data $\mathbf{Y} = \mathbf{XA}$, assuming that the underlying group structure may be well approximated by the one in the d dimensions of the block matrix $\mathbf{Y}^*$, i.e.,

$$f(\mathbf{Y}|\mathbf{Z} = \mathbf{z}) = \sum_{k=1}^{G} \pi_k \phi_k(\mathbf{Y}; \mu_{Y_k}, \Sigma_{Y_k}). \tag{3}$$

This assumption does not imply that $\bar{\mathbf{Y}}$ is not useful for clustering, but only that it contains some information on the group membership $\mathbf{z}$ that is very similar to that already available in $\mathbf{Y}$. Therefore, in terms of distributional representation, it seems reasonable to think of $\bar{\mathbf{Y}}$ as conditionally independent of $\mathbf{Z}$ given $\mathbf{Y}$; it could be necessary for the clustering, but only if $\mathbf{Y}$ is not present (Fop and Murphy, 2018). Its density, by assuming a multidimensional multivariate regression model between $\bar{\mathbf{Y}}$ and $\mathbf{Y}$, can be expressed as:

$$f(\bar{\mathbf{Y}}|\mathbf{Y}) = \phi(\bar{\mathbf{Y}}; \mu_{\bar{\mathbf{Y}}|\mathbf{Y}}, \Sigma_{\bar{\mathbf{Y}}|\mathbf{Y}}) \tag{4}$$

where

$$\mu_{\bar{\mathbf{Y}}|\mathbf{Y}} = \mu_{\bar{\mathbf{Y}}} + \Sigma_{\bar{\mathbf{Y}}\mathbf{Y}}\Sigma_{\mathbf{Y}}^{-1}(\mathbf{Y} - \mu_{\mathbf{Y}})$$
$$\Sigma_{\bar{\mathbf{Y}}|\mathbf{Y}} = \Sigma_{\bar{\mathbf{Y}}} - \Sigma_{\bar{\mathbf{Y}}\mathbf{Y}}\Sigma_{\mathbf{Y}}^{-1}\Sigma_{\mathbf{Y}\bar{\mathbf{Y}}}. \tag{5}$$

(5) describes the Schur complemement of the block $\Sigma_{\mathbf{Y}}$ in the $p \times p$ block-matrix

$$\Sigma_{\mathbf{Y}^*} = \begin{bmatrix} \Sigma_{\mathbf{Y}} & \Sigma_{\mathbf{Y}\bar{\mathbf{Y}}} \\ \Sigma_{\bar{\mathbf{Y}}\mathbf{Y}} & \Sigma_{\bar{\mathbf{Y}}} \end{bmatrix}.$$

The distribution of $\mathbf{Y}^*$ is the product of the marginal density of $\mathbf{Y}$, $f(\mathbf{Y}|\mathbf{z})$, and the conditional density of $\bar{\mathbf{Y}}|\mathbf{Y}$, $f(\bar{\mathbf{Y}}|\mathbf{Y})$:

$$f(\mathbf{Y}^*|\mathbf{Z} = \mathbf{z}) = \left[\sum_{k=1}^{G} \pi_k \phi_k(\mathbf{Y}; \mu_{\mathbf{Y}_k}, \Sigma_{\mathbf{Y}_k})\right] \phi(\bar{\mathbf{Y}}; \mu_{\bar{\mathbf{Y}}|\mathbf{Y}}, \Sigma_{\bar{\mathbf{Y}}|\mathbf{Y}}). \tag{6}$$

(6) can be easily rewritten in terms of log-likelihood as:

$$\sum_{i=1}^{n} \log[f(\mathbf{y}_i^*|\mathbf{Z}_i = \mathbf{z}_i)] = \sum_{i=1}^{n} \log[f(\mathbf{y}_i|\mathbf{Z}_i = \mathbf{z}_i)] + \sum_{i=1}^{n} \log[\phi(\bar{\mathbf{y}}_i|\mathbf{y}_i)]. \tag{7}$$

The BIC corresponding to (7) is:

$$\mathrm{BIC} = \mathrm{BIC}_{\mathrm{GMM}}(\mathbf{Y}) + \mathrm{BIC}_{\mathrm{reg}}(\bar{\mathbf{Y}}|\mathbf{Y}), \tag{8}$$

where $\mathrm{BIC}_{\mathrm{GMM}}(\mathbf{Y}) = 2\log[f(\mathbf{Y})] - q_{\mathbf{Y}}\log(n)$ is the BIC associated to the Gaussian mixture fitted on the d-dimensional data and $\mathrm{BIC}_{\mathrm{reg}}(\bar{\mathbf{Y}}|\mathbf{Y}) = 2\log[f(\bar{\mathbf{Y}}|\mathbf{Y})] - q_{\bar{\mathbf{Y}}}\log(n)$ is the BIC for the linear regression of the $(p - d)$ last columns of $\mathbf{Y}^*$ on the first d ones. In high dimensional clustering the dimension of $\mathbf{Y}$ is generally much smaller than that of $\bar{\mathbf{Y}}$ and is the same for each projection; therefore, the size of $\mathbf{Y}$ only slightly affects the penalty term in the $\mathrm{BIC}_{\mathrm{reg}}$. This condition does not hold in the scenarios described in Maugis et al. (2009a) and Scrucca and Raftery (2018).

The number of free parameters of the GMM on $\mathbf{Y}$ and those of the linear regression are described by $q_{\mathbf{Y}}$ and $q_{\bar{\mathbf{Y}}}$, respectively. In order to allow for great flexibility, $\Sigma_{\bar{\mathbf{Y}}|\mathbf{Y}}$ is assumed to have a general form and, thus,

$$q_{\bar{\mathbf{Y}}} = (p - d)(d + 1) + \frac{(p - d)[(p - d) + 1]}{2}.$$

When the number of observed features p is particularly large with respect to d, a restricted form for $\Sigma_{\bar{\mathbf{Y}}|\mathbf{Y}}$ is suggested. Namely, $\Sigma_{\bar{\mathbf{Y}}|\mathbf{Y}} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{p-d}^2)$. In this case, the number of free parameters for the regression model reduces to $q_{\bar{\mathbf{Y}}} = (p - d)(d + 1) + (p - d)$.

As depicted in Figure 1, the criterion we propose provides a good ranking of the random projections according to the goodness of the partition they induce. Specifically, models with increasing BIC exhibit large values for the adjusted Rand index (ARI, Hubert and Arabie, 1985), i.e., a measure of the similarity between the classification yielded by the GMM on the reduced data and the true class membership.
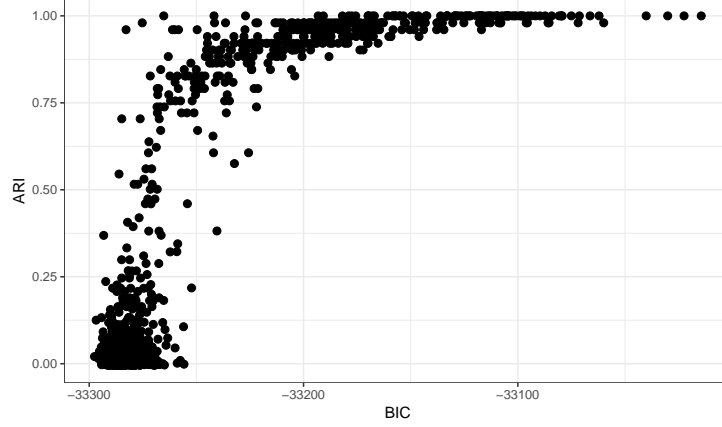
Figure 1: ARI of the classification yielded by the GMM on $B = 1000$ different 8-dimensional projections of a simulated dataset with $p = 100$, $G = 2$, $n_1 = n_2 = 100$ and the true class membership, ordered by increasing values of the BIC.

Such result does not imply that the projections associated with the largest BIC would yield redundant solutions. In fact, the random projection method naturally perturbs different configurations of the original data, thus, inducing diversity. Pairwise ARIs on all clustering vectors returned by the selected top 100 projections largely differ. For example, in the scenario depicted in Figure 1 the average ARI value ranges from 0.6170 to 0.9845, for 100 replications.

3.2 On the result aggregation

A possible solution to the inherent instability associated with random projections involves the use of cluster ensembles that combine multiple individual partitions into a single consensus one. This process was pioneered by Strehl and Ghosh (2002) who proved that ensembles can provide robust and stable solutions across different problem domains. A detailed review of the state-of-the-art cluster ensemble methods can be found in Boongoen and Iam-On (2018), where both theoretical aspects and empirical applications are widely discussed.

Consensus clustering algorithms generally derive the ultimate data partition by minimizing an objective function that measures how dissimilar each hard or soft consensus candidate is from the ensemble members. In this work, we suggest to derive the final unit allocation by using the greedy algorithm proposed by Dimitriadou et al. (2002) and developed in Hornik (2005); in the following, a description of their procedure is sketched.

The aim is to look for a partition $\mathbf{P}$ of the given dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ into G classes that optimally represents a given set of B^* partitions of the same set. Each of these B^* partitions is represented by an $n \times G$ membership matrix $\mathbf{U}^{(b)}$, $b = 1, \ldots, B^*$. The

element $u_{ik}^{(b)}$ of $\mathbf{U}^{(b)}$ is the membership of $\mathbf{x}_i$ to the kth class, $k = 1, \ldots, G$, of the bth partition. The final partition $\mathbf{P}$ is encoded as an $n \times G$ matrix with element p_{ik}. In order for the partition $\mathbf{P}$ to be optimal, it needs to be at the smallest distance from the considered B^* partitions. Thus, the task is to find $\mathbf{P}$ in such a way that

$$
\min_{\mathbf{P}} \left(\frac{1}{B^*} \sum_{b=1}^{B^*} h(\mathbf{U}^{(b)}, \mathbf{P}) \right),
$$

where $h(\mathbf{U}^{(b)}, \mathbf{P})$ is the dissimilarity function between $\mathbf{U}^{(b)}$ and $\mathbf{P}$:

$$
h\left(\mathbf{U}^{(b)}, \mathbf{P} \right) = \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{u}_i^{(b)} - \mathbf{p}_i \right\|^2.
$$

However, because of the label switching clustering issue, any relabeling of the classes is to be considered as the same partition. Thus, partitions $\mathbf{U}^{(b)}$ and $\Pi_b(\mathbf{U}^{(b)})$, which only differ by a permutation of the class labels are to be considered the same and the distances should remain the same too:

$$
h\left(\mathbf{U}^{(b)}, \mathbf{P} \right) = h\left(\Pi_b(\mathbf{U}^{(b)}), \mathbf{P} \right), \quad \forall \mathbf{P}.
$$

Therefore, the dissimilarity function $h(\mathbf{U}^{(b)}, \mathbf{P})$ between two clustering partition $\mathbf{U}^{(b)}$ and $\mathbf{P}$ should be rather defined as:

$$
h\left(\mathbf{U}^{(b)}, \mathbf{P} \right) = \min_{\Pi_b} \left(\frac{1}{n} \sum_{i=1}^{n} \left\| \Pi_b(\mathbf{u}_i^{(b)}) - \mathbf{p}_i \right\|^2 \right).
$$

where the minimum is taken over all possible column permutations Π_b.

The task of finding an optimal partition $\mathbf{P}$ is then given by the minimization problem

$$
\min_{\mathbf{p}_1, \ldots, \mathbf{p}_n} \min_{\Pi_1, \ldots, \Pi_{B^*}} \left(\frac{1}{B^*} \sum_{b=1}^{B^*} \frac{1}{n} \left\| \Pi_b(\mathbf{u}_i^{(b)}) - \mathbf{p}_i \right\|^2 \right). \tag{9}
$$

In order to find an optimal $\mathbf{P}$, (9) has to be minimized simultaneously for $\mathbf{p}_i$ and Π_b, because the choice of the permutations Π_b depends on $\mathbf{p}_i$. As a direct solution of the minimization problem is unfeasible, a greedy algorithm is employed.

The iterative procedure determines, at each step b $(b = 1, \ldots B^*)$, the locally optimal permutation matrix Π_b for relabeling by minimizing the Euclidean distance between the previously determined consensus candidate $\mathbf{P}^{(b-1)}$ (note that at the initial step $\mathbf{P}^{(0)} \equiv \mathbf{U}^{(1)}$) and all the possible permutations of the membership matrix $\mathbf{U}^{(b)}$, $\Pi_b(\mathbf{U}^{(b)})$. Then, it derives the updated consensus partition by:

$$
\mathbf{P}^{(b)} = \frac{1}{b} \sum_{l=1}^{b} \hat{\Pi}_l(\mathbf{U}^{(l)}) = \frac{b-1}{b} \mathbf{P}^{(b-1)} + \frac{1}{b} \hat{\Pi}_b(\mathbf{U}^{(b)}).
$$

In so doing, this sequential method helps to tackle the issue of simultaneous combination of all partitions, otherwise computationally unfeasible. For further details, see Dimitriadou et al. (2002), and Hornik (2005) for the corresponding R package.

3.3 Random projection ensemble clustering algorithm

In this paper, a new model-based clustering method for high-dimensional data based on random projections is introduced. The procedure is sketched in Algorithm 1.

Algorithm 1: RPEClu

Result: Partition of the original data $\mathbf{X}$ into G groups.

Set G, d, B, B^*;

1. **for** b **from** 1 **to** B **do**

     (i) Generate an independent d-dimensional random projection matrix $\mathbf{A}_b$;

     (ii) Fit a GMM with G components on the projected data $\mathbf{Y} = \mathbf{X}\mathbf{A}_b$;

     (iii) Retain the induced data partition $\mathbf{C}_b$;

     (iv) Fit linear regression of $\bar{\mathbf{Y}}$ on $\mathbf{Y}$, where $\bar{\mathbf{Y}} = \mathbf{X}\bar{\mathbf{A}}_b$ and $\bar{\mathbf{A}}_b$ is the
         orthogonal complement of $\mathbf{A}_b$;

     (v) Compute the BIC as described in (8) ;

end

2. Sort the BIC values of the B solutions;

3. Select the top B^* projections from the BIC list of point 2;

4. Aggregate the corresponding B^* cluster membership vectors via consensus;

5. Partition the original data $\mathbf{X}$ according to the consensus membership of point
     4;

---

4 Practical considerations

4.1 Computational complexity

The algorithm we propose derives the final partition by aggregating the results of Gaussian mixture model clustering performed on an 'optimal' subset of random projections.

The first step of this procedure involves the computation of B random projection matrices. The cost of this operation varies according to the method used: namely, generating a single RP from the Haar measure requires $O(pd^2)$ operations, whilst choosing each entry of this matrix uniformly and independently from $[-1,1]$ takes time only $O(dp)$ (see Achlioptas, 2003).

Once the projections have been generated, the original high-dimensional data should be embedded into the lowered spaces; each projection requires $O(npd)$ operations.

Then, for $b = 1, \ldots, B$, a GMM is fitted on the reduced set $\mathbf{Y} = \mathbf{X}\mathbf{A}_b$ with a total cost of $O(Gd^3) \cong O(d^3)$, while the orthogonal complement of $\mathbf{A}$ is computed via QR decomposition with a cost of $O(p^3)$. Simultaneously, a multiple linear regression of $\bar{\mathbf{Y}} = \mathbf{X}\bar{\mathbf{A}}_b$ on $\mathbf{Y}$ is computed. The cost of this step is $O((p-d)^3)$. Finally, the BIC values computed as in (8) are sorted and observations are clustered by using the best B^* projections (i.e., those yielding the highest values for the BIC). These steps involve $O(B^*)$[1] and $O(B^*nd)$ resources, respectively.

---

[1]See the R Documentation for the `sort` function with default settings.

4.2 Choice of d

Dasgupta (2000) proved that data from an arbitrary mixture of G Gaussian distributions can be randomly embedded into a subspace of just $O(\log G)$ dimensions, while preserving the group structure almost perfectly. Furthermore, if $d < \log G$, the worsening of the mapping performance is gradual. This result is particularly appealing as it proves that the dimension of the projection subspace is independent of the original dimensions of the data, that is, d does not depend upon n nor p. The numerical experiments conducted on both simulated and real data corroborate Dasgupta's result.

In particular, we generated data from two $p = 500$-dimensional Gaussian populations, with correlated features (0.90 for all of the features); the size of each sample was set to 100. RPEClu was applied for different values of d (namely, $\{2, 4, 8, 11, 15, 30\}$, corresponding to $\lfloor 1\log(G) \rceil + 1, \lfloor 5\log(G) \rceil + 1, \lfloor 10\log(G) \rceil + 1, \lfloor 15\log(G) \rceil + 1, \lfloor 20\log(G) \rceil + 1, \lfloor 42\log(G) \rceil + 1$, where $\lfloor x \rceil$ refers to the rounding of x to the next integer), with G considered as known (and equal to 2), $B = 1000$, $B^* = 100$ and no constraints were imposed on the covariance matrices; 50 replications were performed. Figures 2 reports the adjusted Rand index of the obtained partition and the true class membership for different d values.

The real data (fully described in Section 6.1) refer to $n = 231$ samples of homogenized raw meat, whose spectra are recorded; the total number of variables is $p = 1050$ and $G = 5$. RPEClu is run with $B = 1000$, $B^* = 100$ and different values of $d = \{3, 17, 33, 65, 98, 130\}$ corresponding to $\lfloor 1\log(G) \rceil + 1$, $\lfloor 10\log(G) \rceil + 1$, $\lfloor 20\log(G) \rceil + 1$, $\lfloor 40\log(G) \rceil + 1$, $\lfloor 60\log(G) \rceil + 1$, $\lfloor 80\log(G) \rceil + 1$; G was considered as known and equal to 5, no constraints were imposed on the covariance matrices. Figures 3 reports the adjusted Rand index of the obtained partition and the true class membership for different d values.

Figures 2 and 3 clearly show that a choice of $d = \lfloor 10\log(G) \rceil + 1$ works pretty well; higher values of d do dot noticeably improve the final performance.

4.3 Choice of B and B^*

The random projection ensemble clustering performances strongly depend on the possibility to identify those random projections that induce a very clear group structure in the reduced space.

The choice of B^*, i.e., the number of 'base' models to retain in the final ensemble, is insidious. Several studies have shown that ensembles of classifiers are generally more effective when they are constructed from members whose errors are dissimilar; see, for example, Kittler et al. (1998). In fact, aggregating the base results of models that agree on how a dataset should be partitioned does not provide any improvement. The random projection method itself represents a valid technique to introduce artificial instability (and thus *diversity*) to an ensemble as it allows to generate clustering results from different perturbed configurations of the original data. However, as Fern and Brodley (2003) point out, taking into account too many projections may degrade the final result, especially when the original features are highly correlated; furthermore, it surely increases the computational cost of the procedure. On the other side, considering a very small ensemble can be risky, too. In fact, since in clustering no *apriori* knowledge
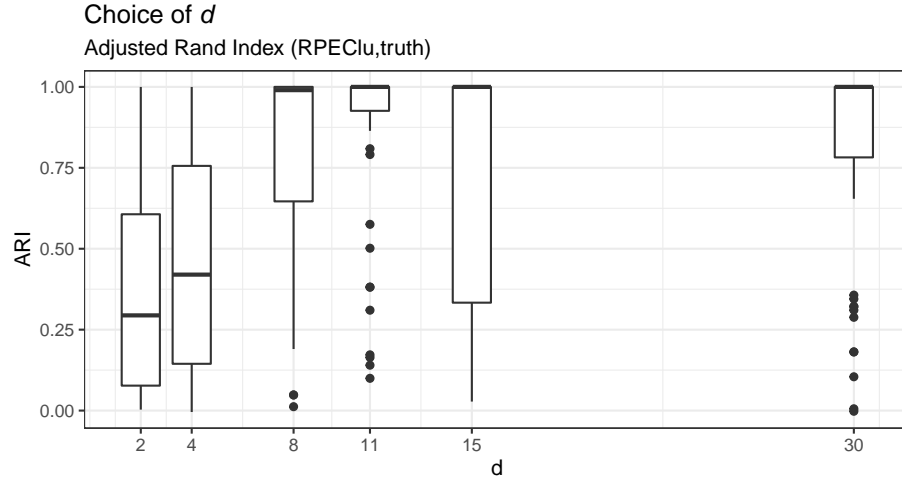
Figure 2: Study on the choice of d. The boxplots report the ARI of the random projection ensemble clustering algorithm for different dimensions of the projected space, d. The numbers of generated and selected projections are set equal to $B = 1000$ and $B^* = 100$, respectively.
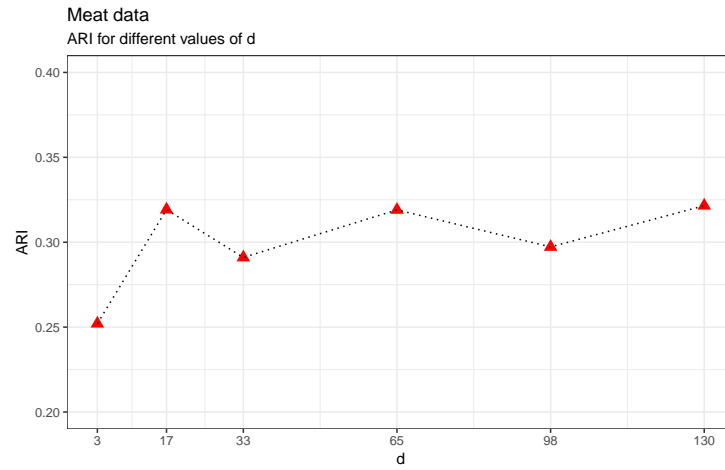

Figure 3: Meat data. ARI of the random projection ensemble clustering algorithm for different dimensions of the projected space, d. The numbers of generated and selected projections are set equal to $B = 1000$ and $B^* = 100$, respectively.
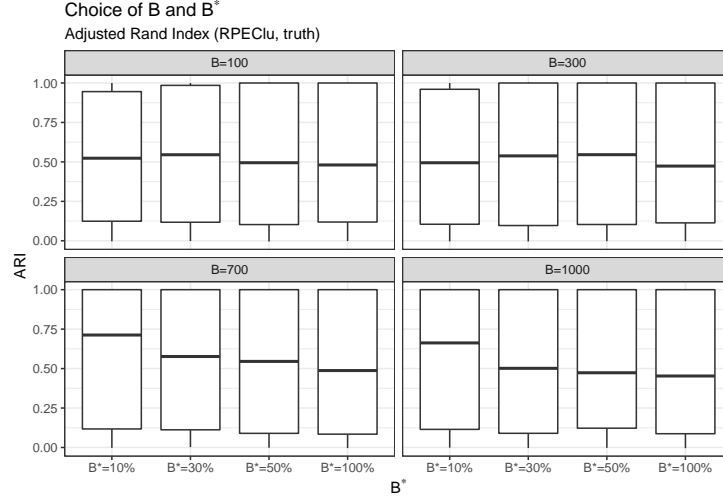
Figure 4: ARI of the partition yielded by the RPEClu algorithm and the true class membership on simulated data with $p = 500$, $G = 2$ and $n_1 = n_2 = 100$, over 100 replications. Results for different number of generated projections B (namely 100, 300, 700 and 1000) are displayed in panels; the B^* retained projections varies on the x-axis as a fraction of B (namely, 10%, 30%, 50% and 100%).

of the true data structure is available, identifying the best unsupervised classifier is not a trivial task and, therefore, any criterion (including the BIC we propose) could be confused. In order to avoid the selection of too similar or inaccurate base classifiers, a compromise solution for B^* is highly suggested.

Figure 4 displays the distribution of the ARI for the simulated scenario studied in section 4.2 for different values of B and B^*, over 100 replications. The resulting message says that the choice of B^* is indeed the most crucial and, provided that the number of generated projections is large enough, a fraction of 10% (or of an overall number of projections that is about 100) generally returns better results; accuracy decreases for larger ensembles.

Figure 5 reports the study results for the meat real data (employed in Section 4.2 and fully described in Section 6.1). The plot highlights how larger B settings are associated with better results; here, an ensemble of 210 classifier would return the best result in terms of accuracy, but good performances are also observable with a smaller ensemble chosen over a larger set of projections.

On the basis of the numerical evidences we suggest $B = 1000$ and $B^* = 100$ as generally good choices; however, computationally less demanding $B = 700$ with a fraction of retained projections of 10-30% can also return acceptable results.
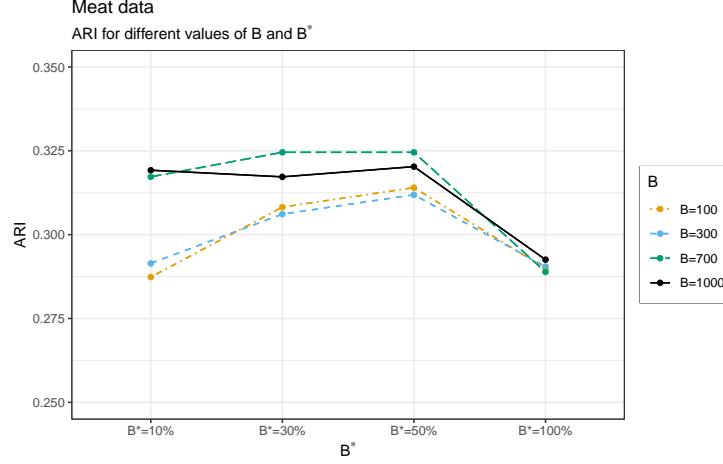
Figure 5: ARI of the partition yielded by RPEClu algorithm and the true class membership on the Meat dataset. The ensemble size is taken as fraction of the total number of generated projections and is reported on the *x*-axis. The dash-dot yellow line refers to $B = 100$, the dashed blue line to $B = 300$, the long-dashed green line to $B = 700$ and the black solid line to $B = 1000$.

5 Simulation study

The performance of the RPEClu algorithm is evaluated in a variety of scenarios through an extensive simulation study. In particular, $G = \{2, 4\}$ different Gaussian clusters of size 100 are generated in $p = \{100, 500, 1000\}$ dimensions by using the `sim_normal` function of the `clusteval` R package (Ramey, 2012). According to its parametrization, each population has a *p*-dimensional multivariate normal distribution, with mean vector

$$\mu_k = \frac{1}{2} \sum_{j=1}^{p/G} \mathbf{e}_{(p/G)(k-1)+j},$$

where $\mathbf{e}_k$ is the *k*-th canonical basis vector; therefore, the first p/G dimensions of μ_1 are set to 1 and all the remaining to 0, the second p/G dimensions of μ_2 are set to 1 and all the remaining to 0, and so on. The *k*-th population covariance matrix is

$$\Sigma_k = (1 - \tau_k)\mathbb{1}_p + \tau_k I_p,$$

where $\mathbb{1}_p$ and I_p denote the $p \times p$ matrix of ones and identity matrix, respectively. Here, $-(p-1)^{-1} < \tau_k < 1$ governs the intra-class correlation; throughout the study, we evaluate different levels of correlation between variables, i.e., we take $\tau_k = \{0.1, 0.3, 0.4, 0.6\}$, corresponding to correlation values of $\{0.9, 0.7, 0.6, 0.4\}$ so as to explore how the clustering algorithm behaves in different situations.

As an exemplification, consider the simple case of $p = 6$, $G = 3$ and $\tau_k = 0.1 \; \forall k$. The parameters of the three Gaussians are the following:

$$\mu_1^\top = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mu_2^\top = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \Sigma_k = \Sigma = \begin{pmatrix} 1 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 1 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 1 \end{pmatrix}$$

$$\mu_3^\top = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Furthermore, we consider scenarios characterized by both *homoscedastic* (settings 1–12) and *heteroscedastic* (settings 13–16) components. Scenarios with *heteroscedastic rotated* components are also investigated (settings 17–20). In this case, as depicted in the illustrative example of Figure 6, the first fifty odd variables of half of the groups are rotated with respect to the axis $x = 0$.

In addition, we studied the behaviour of our proposal in contexts where original data deviate from Normality. In particular, settings 21–23 consider the exponential, the logarithm and the square-root transformations of p-variate Gaussian distributions (for both the logarithm and the square root functions the absolute value was taken first), respectively (p=100, n_k=100, $k = 1, \dots, G$); the number of groups is set to two and only 50% of the variables are relevant for clustering. Scenarios 24–26 extend the study to the case of four groups. A brief description of the simulation settings considered for the analysis is given in Table 1 for Gaussian scenarios 1–20 and in Table 2 for non-Gaussian scenarios 21–26; more details are given in the Supplementary Material.

To validate our proposal, we apply other clustering algorithms on the same settings: the 'standard' Gaussian mixture model (McLachlan and Peel, 2000) (via `Mclust` function of the `mclust` package (Scrucca et al., 2016)), the K-means algorithm (MacQueen et al., 1967; Hartigan and Wong, 1979) (via `kmeans` function (R Core Team, 2020)), Ward's agglomerative hierarchical clustering (Ward, 1963) (via `hclust` function) and the Partition Around Medoids (Pam) (Kaufman and Rousseeuw, 2009) (via `pam` function of the `cluster` package (Maechler et al., 2019)). We also applied the high-dimensional data clustering procedure proposed by Bouveyron et al. (2007) and implemented in the R package `HDclassif` (Bergé et al., 2012), allowing for both Cattel's scree test (HDDC-C) and BIC (HDDC-B) methods to estimate the intrinsic dimensions of each subclass. Two other recent procedures that have shown good performances in the context of high-dimensional unsupervised classification are also included: namely, the spectral clustering approach (Ng et al., 2002) (`specc` function of the `kernlab` package (Karatzoglou et al., 2004)) and the affinity propagation algorithm (Frey and Dueck, 2007) (`apclusterK` function of the `apcluster` package (Bodenhofer et al., 2011)). A further comparison is with the variable selection methodology for Gaussian model-based clustering (Cl VarSel) presented in Section 2. This procedure is applied by using the `clustvarsel` function included in the namesake R package (Scrucca and Raftery, 2018). We also implemented the RP-KM algorithm proposed in Boutsidis et al. (2010), by applying the K-means algorithm with 5 random starts to the dimensionally reduced data with $d = \lfloor cG/\varepsilon^2 \rceil + 1$, with $c = 10$ and $\varepsilon = 1/3$.

The number of groups G is always taken as known. The default settings of each algorithm are considered, except for the K-means which run with 5 starts. As previously discussed, the RPEClu algorithm is performed with $B = 1000$, $B^* = 100$, $G = \{2, 4\}$ and $d = \{\lfloor 10\log(2)\rceil + 1 = 8, \lfloor 10\log(4)\rceil + 1 = 15\}$, respectively.

16

Table 1: Summary description of the simulation settings 1–20. When only one value for τ is given, it means that *homoscedastic* Gaussian components are considered. The $*$ indicates *rotated* components.

Setting	p	G	τ
1	100	2	0.1
2	500	2	0.1
3	1000	2	0.1
4	100	4	0.1
5	500	4	0.1
6	1000	4	0.1
7	100	2	0.4
8	500	2	0.4
9	1000	2	0.4
10	100	4	0.4
11	500	4	0.4
12	1000	4	0.4
13	100	2	0.1-0.6
14	100	2	0.1-0.3
15	500	2	0.1-0.6
16	500	2	0.1-0.3
17	100	2	0.1-0.6*
18	100	2	0.1-0.3*
19	500	2	0.1-0.6*
20	500	2	0.1-0.3*

Table 2: Summary description of the simulation settings 21–26. Non-Gaussian distributions.

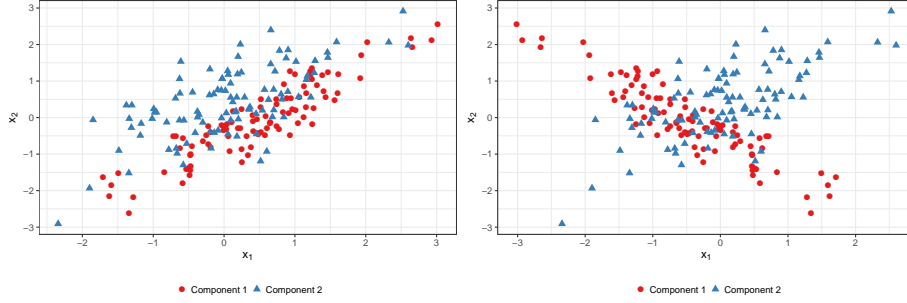| Setting	p	G	Transformation
21	100	2	Exp(Gaussian)
22	100	2	Log(abs(Gaussian))
23	100	2	Sqrt(abs(Gaussian))
24	100	4	Exp(Gaussian)
25	100	4	Log(abs(Gaussian))
26	100	4	Sqrt(abs(Gaussian))

Figure 6: An example of bivariate dataset with *heteroscedastic rotated* components: variable x_1 of Component 1 (red points) is rotated with respect to the axis $x = 0$.

Figure 7 contains the aggregated results for the considered scenarios: (a)-(b) *homoscedastic* Gaussian components with highly correlated features, with two and four groups respectively; (c)-(d) *homoscedastic* Gaussian components with mildly related features, with two and four groups respectively; (e) *heteroscedastic* Gaussian components and *heteroscedastic rotated* Gaussian components; (f)-(g) *non-Gaussian* components, with two and four groups respectively. The boxplots show the distribution of the ARI over 100 simulations of each setting; the horizontal dotted line helps the comparison with the other approaches, by highlighting the median ARI for the random projection ensemble clustering algorithm. Adjusted Rand indexes of each individual setting are reported in the Supplementary Material.

Results coming from this numerical study clearly show the general effectiveness of the algorithm we introduce. In fact, for almost all the situations considered in the boxplots of Figure 7, the RPEClu produces better solutions than those from the other state-of-the-art methods, including the two procedures that usually work well in high-dimensional contexts (i.e., spectral and affinity propagation clustering algorithms). Not surprisingly, this aspect is particularly evident in those scenarios where the original features are strongly related, as some approaches tend to discard this kind of information. With reference to the mixtures of Gaussians, for example, when p is very large compared to n, mclust is able to estimate only those models that have a small number of parameters, i.e., models with spherical, diagonal, or homoscedastic covariance matrix. Furthermore, the K-means algorithm can be viewed as a procedure which attempts to model the data as a mixture of Gaussian distributions with diagonal covariance matrices and thus it does not account for the variable correlation. Scenarios with mildly related features, i.e., 7–9 and 10–12, appear to be very hard tasks: basically all the considered methods, except for HDDC-C, perform poorly in terms of recovering the 'true' grouping structure. The HDDC methods (Bouveyron et al., 2007) perform generally well; however, the Cattel's scree test method to estimate the intrinsic dimensions of each subclass always outperforms the BIC-based one.

As expected, K-means algorithm, hierarchical agglomerative clustering with Ward's method and Pam often fail because the distance measures they rely on become increas-
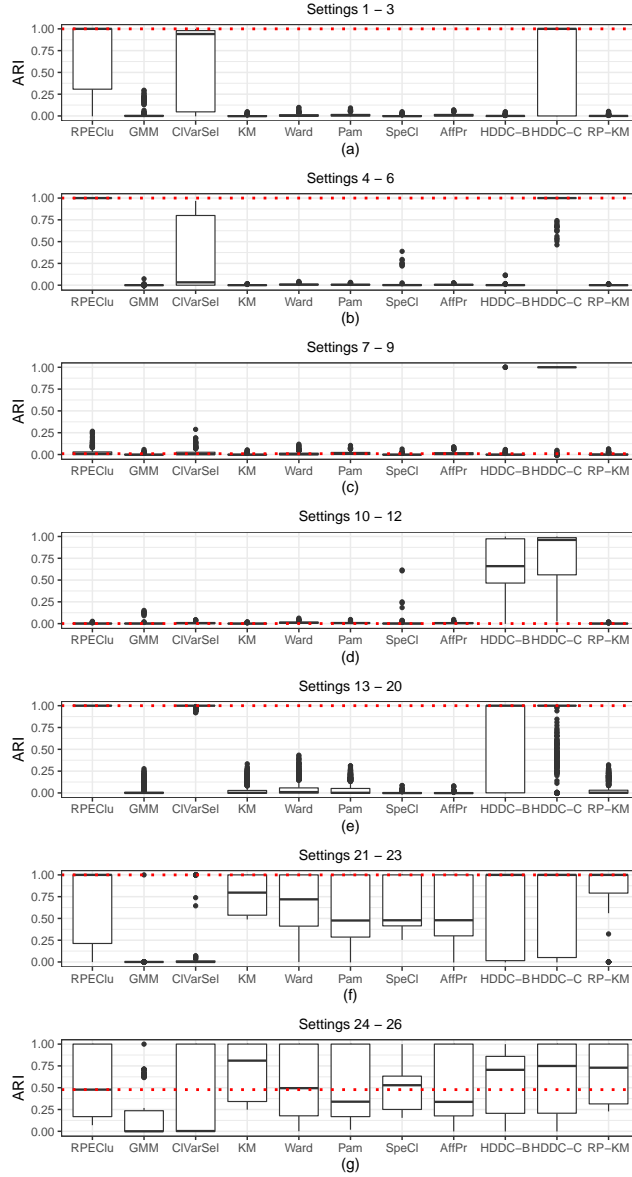
18

Figure 7: Performance of different clustering algorithms. The labels along the horizontal axis refer to the different methods: *RPEClu*, Random projections ensemble clustering; *GMM*, Gaussian mixture model; *ClVarSel*, Gaussian mixture model with variable selection; *KM*, *K*-means clustering; *Ward*, hierarchical clustering with Ward's method; *Pam*, partition around medoids algorithm; *SpeCl*, spectral clustering; *AffPr*, affinity propagation; *HDDC-B* high-dimensional data clustering with BIC method; *HDDC-C* high-dimensional data clustering with Cattel's scree test; *RP-KM* random-projection for *K*-means clustering. The seven panels show the distribution of the adjusted Rand index for (a) homoscedastic Gaussian clusters ($G = 2$) with highly correlated features, (b) homoscedastic Gaussian clusters ($G = 4$) with highly correlated features, (c) homoscedastic Gaussian clusters ($G = 2$) with mildly correlated features, (d) homoscedastic Gaussian clusters ($G = 4$) with mildly correlated features (e) heteroscedastic Gaussian clusters ($G = 2$), (f) non-Gaussian clusters ($G = 2$) with 50% of relevant features and (g) non-Gaussian clusters ($G = 4$) with 50% of relevant features.

Table 3: Distribution of the meat samples.

Species	Samples
Beef	32
Chicken	55
Lamb	34
Pork	55
Turkey	55

ingly meaningless in high-dimensions; however, with non-Gaussian data they exhibit an acceptable performance. The RP for K-means clustering approach does not seem to improve considerably over the standard KM.

A special mention should be made for the variable selection procedure (ClVarSel) that seems capable to correctly identify relevant clustering information in most of the settings. Nevertheless, it underperforms the RPEClu, especially in the case of homoscedastic components with highly correlated features or in case of non-Gaussian data. This outcome corroborates our initial idea that feature extraction techniques are generally more effective than feature selection ones.

Globally, the capability of the RPEClu in recovering the cluster membership does not change too much with p nor with the number of groups. In addition, it is quite robust to deviations from Gaussianity as shown in plots (f) and (g).

6 Real data examples

For illustration, we evaluate the performances of the clustering algorithms described in the previous section on two different real data experiments. Namely, we use the set of near infrared spectroscopic meat data originally described in the study of Downey et al. (2000), and subsequently analyzed in Dean et al. (2006) and Murphy et al. (2010), and the lymphoma gene expression dataset, from Alizadeh et al. (2000) and further studied in Dettling and Bühlmann (2002), Dettling (2004) and Chung and Keles (2010).

6.1 Meat data

This dataset contains $n = 231$ samples of homogenized raw meat coming from $G = 5$ different animal species. The distribution of the samples is described in Table 3. The spectra are recorded over the wavelength range 400–2498 nm, with measurements taken every 2 nm. The total number of variables is thus $p = 1050$. Figure 8 shows the spectrum of each sample, grouped by type of meat. The original data source information and conditions for the use of the data are outlined in the online supplementary material of Murphy et al. (2010).

The objective of the analysis is to partition the set of 231 samples so as to reflect the corresponding type of meat by employing the information coming from their spectra. The number of groups $G = 5$ is taken as known; we set $B = 1000$, $B^* = 100$ and
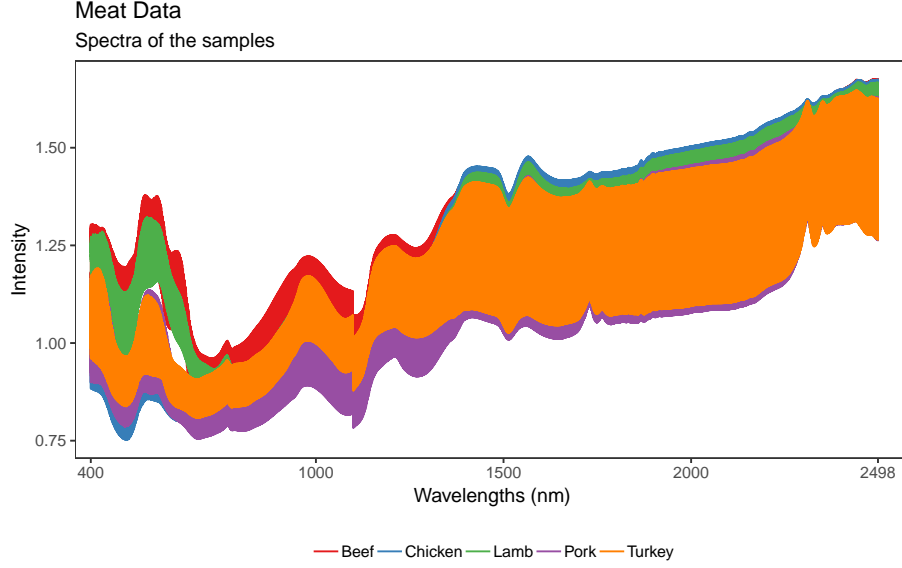
Figure 8: Meat data. Spectra of the samples, grouped by type of meat.

$d = \lfloor 10\log(5) \rfloor + 1 = 17$. Table 4 contains the adjusted Rand index yielded by each method.

Globally, none of the employed methods is able to perfectly recover the original cluster membership of the meat data. Nevertheless, the RPEClu algorithm, together with HDDC-C, provides an adjusted Rand index that is considerably superior to all the other solutions. The GMM performs poorly; this is probably due to the fact that, as p is very large, `Mclust` could only estimate mixtures of Gaussians with spherical or diagonal covariance matrices, while data require a model that accounts for the high correlation between the features. The Clust VarSel methodology could not run because the too much correlated variables induced a model perfectly correlated with the response. HDDC-C performs well and better than HDDC-B. The accuracy of RP-KM is similar to that of K-means and among the lowest ones.

6.2 Gene expression data

The lymphoma dataset (available in the R package `spls` (Chung et al., 2019)) contains the expression levels of $p = 4026$ genes for $n = 62$ patients. The study reports that 42 subjects have diffuse large B-cell lymphoma (DLBCL), 9 follicular lymphoma (FL), and 11 chronic lymphocytic leukemia (CLL). All gene expression profiles were base 10 log-transformed and, in order to prevent single arrays from dominating the analysis, standardized to zero mean and unit variance, as described in Dettling and Bühlmann (2002) and Dettling (2004).

The objective of the analysis is to group patients according to the corresponding

21

Table 4: ARI for the Meat data and the computational time (in seconds) required by each clustering method.

Method	ARI	Time (s)
RPEClu	0.32	6460.76
GMM	0.14	1.39
ClVarSel	-	-
KM	0.18	0.10
Ward	0.23	0.08
Pam	0.18	0.12
SpeCl	0.25	1.23
AffPr	0.18	1.11
HDDC-B	0.22	6.45
HDDC-C	0.31	15.50
RP-KM	0.17	0.11

lymphoma diagnosis, by using the information on their gene expression levels. RPEClu procedure run with $B = 1000$, $B^* = 100$ and $d = \lfloor 10\log(3) \rfloor + 1 = 12$; the number of groups is taken as known and set equal to 3 for all the methods. Clustering results in terms of ARI are reported in Table 5. As it can be seen, the random projection ensemble clustering algorithm is capable to perfectly detect the grouping structure identified by the diagnosis. Mixture of Gaussians, K-means, hierarchical agglomerative clustering with Ward's method, Spectral clustering, HDDC methods and RP-KM provide exactly the same (good) result, up to a label switching. This is due to the fact that, when $p \gg n$, Mclust only works on the restricted set of parsimonious models (e.g. spherical or diagonal models) and, therefore, its optimal solution often slightly improves the one yielded by the hierarchical algorithm. The mixture of Gaussians fitted to the set of 10 variables returned by Clust VarSel does not improve over the full set solution.

7 Some considerations on the choice of G

The presented results refer to the cases for which the number of clusters G is fixed and known. However, in real life applications it may happen that there is no insight about the "true" number of homogeneous groups and such information has to be inferred from the data.

The proposed clustering method is based on an ensemble of clustering vectors. A simple option would be to repeatedly apply RPEClu with different values for G and then choose a posteriori the best solution, according to some clustering quality criteria. However, such an approach would not exploit the information yielded by the model-based approach; furthermore, the adopted clustering quality criterion would largely impact the final choice. For example, the average silhouette width (ASW) may prefer solutions that emphasizes gaps between neighboring clusters; Calinski and Harabasz (CH) penalizes large within-cluster distances more strongly. Pearson version of Hubert's

Gene Expression Data
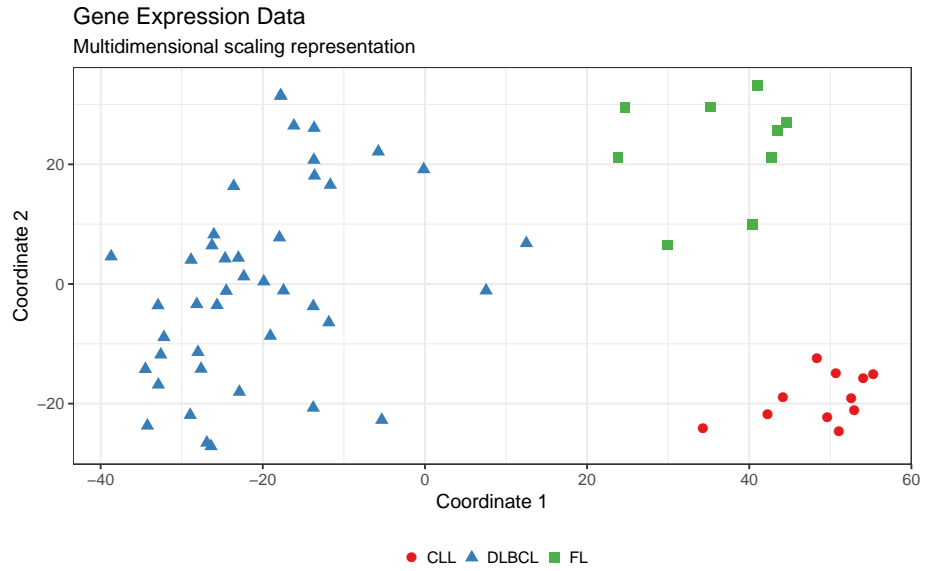
Multidimensional scaling representation

Figure 9: Gene expression data. Multidimensional scaling representation of the samples, grouped by type of diagnosis.

Table 5: ARI for the Gene Expression data and the computational time (in seconds) required by each clustering method.

Method	ARI	Time (s)
RPEClu	1.00	83596.81
GMM	0.95	5.69
ClVarSel	0.44	5290.78
KM	0.95	0.10
Ward	0.95	0.05
Pam	0.84	0.04
SpeCl	0.95	0.22
AffPr	0.84	0.35
HDDC-B	0.95	1.66
HDDC-C	0.95	1.70
RP-KM	0.95	0.06

Table 6: Choice of *G* within the ensemble procedure. Average (standard error) number of clusters returned by the ensemble (column *G*) and average adjusted Rand index (ARI) over 100 replications.

G	ARI
2.00 (0.00)	1.00 (0.00)

Table 7: Choice of *G* within the ensemble procedure for the real datasets.

Data	*G*	ARI
Meat	2	0.31
Gene expression	3	0.90

Γ (PG) emphasizes good approximation of the dissimilarity structure by the clustering in the sense that whether observations are in different clusters should strongly be correlated with large dissimilarity; as the Pearson correlation is based on squares, it penalizes large within-cluster dissimilarities. For a thorough discussion on the characteristics of different criteria see, e.g., Hennig and Liao (2013).

Alternatively, one may let each mixture model choose the optimal *G* via BIC and the ensemble is built on the resulting vectors, each potentially characterized by a different value for *G*. In other words, this approach relies on the ensemble of clustering vectors resulting from a model selection procedure via BIC; the aggregation of such B^* vectors, each potentially pointing to solutions with different number of clusters, can lead to a final group allocation whose size is unknown until the end of the procedure, as the employed ensemble method of Section 3.2 does not need its elements to be homogeneous in terms of cluster number and only follows the ordering of the BIC. Its behaviour is studied on simulated Gaussian data (described in both Sections 4.2 and 4.3) and on the real data sets.

Specifically, we consider 100 replications of the simulated data and the number of groups ranges from 2 to 5. Table 6 shows that the model selection procedure is able to recover the true *G* and to recover the true cluster membership.

Table 7 outlines the results with meat and gene expression data. The performance slightly deteriorates in terms of accuracy for the gene expression, as the corresponding ARI reduces to 0.90. The meat data ensemble yielded a two-cluster solution; this suggests that for such data even the best projections could not identify the correct grouping structure.

Overall, the proposed method for model selection appears to lead to fairly accurate results; however, additional measures can be considered and more evidence would be needed to conclude on that.

8 Discussion

In this work we propose a novel procedure for model-based clustering of high-dimensional data. This procedure is based on random projections and it has been firstly inspired by the original idea of Cannings and Samworth (2017) in the context of supervised classification.

More in detail, we suggest to apply a Gaussian mixture model to random projections of the high-dimensional data and to select a subset of solutions accordingly to the Bayesian information criterion, computed here as discussed in Raftery and Dean (2006); the multiple 'base' results are then aggregated via consensus to obtain the final partition.

Such proposal has been initially motivated by some benefits associated to RPs for learning mixture of Gaussians. Dasgupta (2000) proved that a mixture of G Gaussians can be embedded onto just $O(\log G)$ random coordinates while preserving the separation of the components with probability $1 - \delta$, $\delta \in (0, 1)$; furthermore, he demonstrated that even when the original mixing components exhibit elliptical contours, their projected counterparts are always more spherical.

Method performances, evaluated in terms of ARI with respect to the true class membership on both synthetic and real datasets, seem to confirm our motivating ideas. Overall results indeed show that RPs represent a key ingredient that decisively facilitates the learning of high-dimensional mixtures of Gaussians. Moreover, the advantage of their use in conjunction with GMM becomes even more evident as the correlation between the original variables increases. In fact, when dealing with high-dimensional sets, `Mclust` search is restricted to models with few parameters only (i.e., EII, VEI, VII, VVI, EEI and EVI) whereas data would require more complex parameterizations.

The RPEClu algorithm is a very general tool for model-based clustering of high-dimensional data. We explore in detail its behavior within the Gaussian mixture model framework only; however, many other distributions can in principle be used. Moreover, further options for combining the clustering results can be tested.

The number of clusters G is generally fixed here; however, some model selection procedures are suggested as a valid tool to choose the number of clusters when such information is not available.

Compliance with Ethical Standards

Conflict of Interest. The authors declare that they have no conflict of interest.

References

Achlioptas, D. (2003). Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences 66*(4), 671–687.

Ahfock, D. C., W. J. Astle, and S. Richardson (2020, 07). Statistical properties of sketching algorithms. *Biometrika*. asaa062.

Alizadeh, A. A., M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature 403*(6769), 503–511.

Banfield, J. D. and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*, 803–821.

Bellman, R. (1957). *Dynamic programming*. Princeton University Press.

Bergé, L., C. Bouveyron, and S. Girard (2012). HDclassif: An R package for model-based clustering and discriminant analysis of high-dimensional data. *Journal of Statistical Software 46*(6), 1–29.

Bhattacharya, A., P. Kar, and M. Pal (2009). On low distortion embeddings of statistical distance measures into low dimensional spaces. In *International Conference on Database and Expert Systems Applications*, pp. 164–172. Springer.

Biernacki, C. and A. Lourme (2014). Stable and visualizable Gaussian parsimonious clustering models. *Statistics and Computing 24*(6), 953–969.

Bingham, E. and H. Mannila (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 245–250.

Bodenhofer, U., A. Kothmeier, and S. Hochreiter (2011). Apcluster: an r package for affinity propagation clustering. *Bioinformatics 27*, 2463–2464.

Boongoen, T. and N. Iam-On (2018). Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review 28*, 1–25.

Boutsidis, C., A. Zouzias, and P. Drineas (2010). Random projections for k-means clustering. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (Eds.), *Advances in Neural Information Processing Systems*, Volume 23, pp. 298–306. Curran Associates, Inc.

Bouveyron, C. and C. Brunet-Saumard (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis 71*, 52–78.

Bouveyron, C., G. Celeux, T. B. Murphy, and A. E. Raftery (2019). *Model-based clustering and classification for data science: with applications in R*, Volume 50. Cambridge University Press.

Bouveyron, C., S. Girard, and C. Schmid (2007). High-dimensional data clustering. *Computational statistics & data analysis 52*(1), 502–519.

Cannings, T. I. (2021). Random projections: Data perturbation for classification problems. *WIREs Computational Statistics 13*(1), e1499.

Cannings, T. I. and R. J. Samworth (2017). Random-projection ensemble classification. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*(4), 959–1035.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate behavioral research 1*(2), 245–276.

Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition 28*(5), 781–793.

Chang, W.-C. (1983). On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 32*(3), 267–275.

Chung, D., H. Chun, and S. Keles (2019). *spls: Sparse Partial Least Squares (SPLS) Regression and Classification*. R package version 2.2-3.

Chung, D. and S. Keles (2010). Sparse partial least squares classification for high dimensional data. *Statistical applications in genetics and molecular biology 9*(1).

Dasgupta, S. (1999). Learning mixtures of gaussians. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pp. 634–644.

Dasgupta, S. (2000). Experiments with random projection. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, San Francisco, CA, USA, pp. 143–151. Morgan Kaufmann Publishers Inc.

Dean, N., T. B. Murphy, and G. Downey (2006). Using unlabelled data to update classification rules with applications in food authenticity studies. *Journal of the Royal Statistical Society, Series C: Applied Statistics 55*(1), 1–14.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological) 39*(1), 1–22.

Dettling, M. (2004). Bagboosting for tumor classification with gene expression data. *Bioinformatics 20*(18), 3583–3593.

Dettling, M. and P. Bühlmann (2002). Supervised clustering of genes. *Genome biology 3*(12), research0069–1.

Dimitriadou, E., A. Weingessel, and K. Hornik (2002). A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence 16*(07), 901–912.

Downey, G., J. McElhinney, and T. Fearn (2000). Species identification in selected raw homogenized meats by reflectance spectroscopy in the mid-infrared, near-infrared, and visible ranges. *Applied Spectroscopy 54*(6), 894–899.

Durrant, R. J. and A. Kabán (2015). Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. *Machine Learning 99*(2), 257–286.

Fern, X. Z. and C. E. Brodley (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 186–193.

Fop, M. and T. B. Murphy (2018). Variable selection methods for model-based clustering. *Statistics Surveys 12*, 18–65.

Frey, B. J. and D. Dueck (2007). Clustering by passing messages between data points. *Science 315*(5814), 972–976.

Galimberti, G., A. Manisi, and G. Soffritti (2018). Modelling the role of variables in model-based cluster analysis. *Statistics and Computing 28*(1), 145–169.

Gataric, M., T. Wang, and R. J. Samworth (2020). Sparse principal component analysis via axis-aligned random projections. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 82*(2), 329–359.

Ghahramani, Z. and G. Hinton (1997). The em algorithm for factor analyzers. In *Technical Report CRG-TR-96-1*. University of Toronto Toronto.

Golub, G. H. and C. F. Van Loan (1996). *Matrix computations* (3rd ed.). The Johns Hopkins University Press.

Haar, A. (1933). Der Massbegriff in der Theorie der kontinuierlichen Gruppen. *Annals of Mathematics 34*, 147–169.

Hartigan, J. A. and M. A. Wong (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 28*(1), 100–108.

Hennig, C. (2019). Cluster validation by measurement of clustering characteristics relevant to the user. *Data Analysis and Applications 1: Clustering and Regression, Modeling-estimating, Forecasting and Data Mining 2*, 1–24.

Hennig, C. and T. F. Liao (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 62*(3), 309–369.

Hennig, C., M. Meila, F. Murtagh, and R. Rocci (2015). *Handbook of cluster analysis*. CRC Press.

Hornik, K. (2005). A CLUE for CLUster ensembles. *Journal of Statistical Software 14*(12), 1–25.

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

Johnson, W. B. and J. Lindenstrauss (1984). Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics 26*(189-206), 1.

Karatzoglou, A., A. Smola, K. Hornik, and A. Zeileis (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software 11*(9), 1–20.

Kaufman, L. and P. J. Rousseeuw (2009). *Finding groups in data: an introduction to cluster analysis*, Volume 344. John Wiley & Sons.

Kittler, J., M. Hatef, R. P. Duin, and J. Matas (1998). On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence 20*(3), 226–239.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley symposium on Mathematical Statistics and Probability*, Volume 1, pp. 281–297. Oakland, CA, USA.

Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik (2019). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.0.

Maugis, C., G. Celeux, and M.-L. Martin-Magniette (2009a). Variable selection for clustering with Gaussian Mixture Models. *Biometrics 65*(3), 701–709.

Maugis, C., G. Celeux, and M.-L. Martin-Magniette (2009b). Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics & Data Analysis 53*(11), 3872–3882.

McLachlan, G. J., S. X. Lee, and S. I. Rathnayake (2019). Finite mixture models. *Annual review of statistics and its application 6*, 355–378.

McLachlan, G. J. and D. Peel (2000). *Finite Mixture Models*. Wiley.

McLachlan, G. J., D. Peel, and R. Bean (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis 41*(3-4), 379–388.

McNicholas, P. D. (2016). Model-based clustering. *Journal of Classification 33*(3), 331–373.

McNicholas, P. D. and T. B. Murphy (2008). Parsimonious gaussian mixture models. *Statistics and Computing 18*(3), 285–296.

Montanari, A. and C. Viroli (2011). Maximum likelihood estimation of mixtures of factor analyzers. *Computational statistics & data analysis 55*(9), 2712–2723.

Murphy, T. B., N. Dean, and A. E. Raftery (2010). Variable selection and updating in model-based discriminant analysis for high dimensional data with food authenticity applications. *The Annals of Applied Statistics 4*(1), 396 – 421.

Ng, A. Y., M. I. Jordan, and Y. Weiss (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pp. 849–856.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Raftery, A. E. and N. Dean (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association 101*(473), 168–178.

Ramey, J. A. (2012). *clusteval: Evaluation of Clustering Algorithms*. R package version 0.1.

Ruiz, F. E., P. S. Pérez, and B. I. Bonev (2009). *Information theory in computer vision and pattern recognition*. Springer Science & Business Media.

Scrucca, L. (2016). Genetic algorithms for subset selection in model-based clustering. In *Unsupervised Learning Algorithms*, pp. 55–70. Springer.

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal 8*(1), 289–317.

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.

Slawski, M. et al. (2018). On principal components regression, random projections, and column subsampling. *Electronic Journal of Statistics 12*(2), 3673–3712.

Strehl, A. and J. Ghosh (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research 3*(Dec), 583–617.

Thanei, G.-A., C. Heinze, and N. Meinshausen (2017). Random projections for large-scale regression. In *Big and complex data analysis*, pp. 51–68. Springer.

Vempala, S. S. (2004). *The random projection method*, Volume 65 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Soc.

Ward, J. H. J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association 58*(301), 236–244.

Xu, D. and Y. Tian (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science 2*(2), 165–193.