Journal of the Operational Research Society

ISSN: 0160-5682 (Print) 1476-9360 (Online) Journal homepage: http://www.tandfonline.com/loi/tjor20

Mixed integer linear programming and heuristic methods for feature selection in clustering

Stefano Benati, Sergio García & Justo Puerto

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY, 2017
https://doi.org/10.1080/01605682.2017.1398206

Mixed integer linear programming and heuristic methods for feature selection in clustering

Stefano Benati[a], Sergio García[b] and Justo Puerto[c]

[a]School of International Studies, University of Trento, Trento, Italy; [b]School of Mathematics, The University of Edinburgh, Edinburgh, UK; [c]Institute of Mathematics of the University of Seville (IMUS), Universidad de Sevilla, Sevilla, Spain

ABSTRACT

This paper studies the problem of selecting relevant features in clustering problems, out of a data-set in which many features are useless, or masking. The data-set comprises a set U of units, a set V of features, a set R of (tentative) cluster centres and distances d_{ijk} for every $i \in U$, $k \in R$, $j \in V$. The feature selection problem consists of finding a subset of features $Q \subseteq V$ such that the total sum of the distances from the units to the closest centre is minimised. This is a combinatorial optimisation problem that we show to be NP-complete, and we propose two mixed integer linear programming formulations to calculate the solution. Some computational experiments show that if clusters are well separated and the relevant features are easy to detect, then both formulations can solve problems with many integer variables. Conversely, if clusters overlap and relevant features are ambiguous, then even small problems are unsolved. To overcome this difficulty, we propose two heuristic methods to find that, most of the time, one of them, called q-vars, calculates the optimal solution quickly. Then, the q-vars heuristic is combined with the k-means algorithm to cluster some simulated data. We conclude that this approach outperforms other methods for clustering with variable selection that were proposed in the literature.

ARTICLE HISTORY
Received 4 July 2016
Accepted 20 October 2017

KEYWORDS
Integer linear programming; heuristics; q-vars; cluster analysis; p-median problem

1. Introduction

Clustering is a useful and important unsupervised learning technique widely studied in the literature. The goal of clustering is to group similar units (or objects) into one cluster, while partitioning dissimilar units into different clusters. Clustering becomes difficult if data contain features (also referred to as variables) with no relevant information. When those features are not detected, the calculation of the dissimilarity between units is biased by their presence, resulting in inconsistent clusters (Fowlkes, Gnanadesikan, & Kettering, 1988). The problem becomes more and more relevant as the number of features in a database increases, as frequently occurs nowadays with data containing hundreds and even thousands of covariates (see Guyon & Elisseef, 2003). Therefore, researchers from different disciplines need tools to discard the noising or masking features that are useless to recognise the true clusters. Previous literature has focused on three methods to select or reject variables. The first is the most simple and consists of calculating peculiar indices, one for each variable, to distinguish those features that contain recognisable patterns. The second approach is the most elaborate and consists of optimising the maximum likelihood function, assuming that data are generated by multivariate distributions. The third approach, whose difficulty is intermediate between the previous ones, is to solve an optimisation model whose structure is easier than the maximum likelihood.

The first approach, developing "clusterability" indices, has been suggested in Andrews and McNicholas (2014), Carmone, Kara, and Maxwell (1999), Morlini and Zani (2013), and Steinley and Brusco (2008b). In Carmone et al. (1999), an index called Total Pairwise Rand Index (TOPRI) has been used to select those features that form the best clusters. Features are selected using a constructive procedure that selects features one at a time until a stop criterion is met. In Steinley and Brusco (2008b) an index called clusterability index is defined and used within a multi-step procedure. This procedure is composed of variable preprocessing, data standardisation and variable selection through optimisation. In Andrews and McNicholas (2014), the role of the index is played by an inequality that balances correlation with variability reduction, used for preliminary data screening. In Morlini and Zani (2013), another index is used to compare the outcome of hierarchical clusterings applied to data subsets. The advantage of all these methods is that the calculation of an index is usually a fast task. However, the "greedy" constructive way in which features are selected is clearly sub-optimal.

The second approach, optimising the data likelihood function, assumes that data are described by multivariate distribution functions. This distribution function

CONTACT Stefano Benati stefano.benati@unitn.it

is a mixture of components, that is, other distributions, each one describing one cluster (see Fraley & Raftery, 2002). The parameters of the mixture distribution are estimated through the maximisation of the likelihood function, so that all means, variances and covariances have to be calculated (with cluster memberships expressed as probabilities). When feature selection is introduced to the model (see Raftery & Dean, 2006), maximum likelihood optimisation must be repeated several times, which increases sharply computational times. Similar approaches appear in Law, Figuereido, and Jain (2004) and Pan and Shen (2007), where a penalty term is added to the maximum likelihood estimation to enforce variables' rejection. The penalisation is justified in the framework of minimum message length (Law et al., 2004), or objective function regularisation (Pan & Shen, 2007). Another approach that simultaneously selects variables and estimates cluster parameters is proposed in Tadesse, Sha, & Vannucci (2005), where Bayesian a-priori distributions are introduced and the variable selection problem is solved through probabilistic search.

These two strategies suffer from opposite drawbacks: Resorting to just one index is too simple, whereas maximum likelihood estimation can be too difficult to solve. A third more convenient approach is to retain the optimisation structure of the variable selection problem by replacing the likelihood function with an easier function. For example, in Benati and García (2014), Brusco (2004), Friedman and Meulman (2004) and Witten and Tibshirani (2010), the easier objective function is the minimisation of distances (or sum of squares), as occurs in the k-means model (MacQueen, 1967). Particularly, in Friedman and Meulman (2004) a penalty term is added to the k-means objective function with the purpose of penalising redundant variables, while in Witten and Tibshirani (2010) a new constraint is introduced to the k-means optimisation model with the same goal. When solved, the two approaches weigh variables according to their importance, but this methodology avoids using the more natural 0–1 combinatorial decisions for selecting or rejecting features. Conversely, 0–1 decisions are used in Brusco (2004) and Benati and García (2014). In Brusco (2004) the k-means objective function depends on variable selection solved with a greedy heuristic algorithm. In Benati and García (2014), optimal clustering and variable selection are calculated with a mixed integer linear programming (MILP) problem using the p-median as the objective function. This approach is promising since very fast exact methods have been recently proposed to solve the p-median problem (see Avella, Boccia, Salerno, & Vasilyev, 2012; García, Labbé, & Marín, 2011).

In this paper, we show how to formulate the optimal feature selection for clustering as an integer programming problem. It will be proved that the problem is NP-complete. In addition, we experimented two different formulations that were solved using CPLEX to determine the practical problem complexity. The experiments showed that computational times depend much on the data: If the relevant variables are clearly recognisable and clusters are well separated, then computational times are negligible even for large size instances (more than 1000 features). Conversely, times increase fast if variables are hard to detect and clusters overlap, to the point that instances with just 40 features are not solved within a reasonable time. Therefore, we propose two heuristic algorithms, derived from the integer linear formulation, in order to apply our methodology to all kind of data, both easy and difficult. The algorithms take advantage from the fact that the optimal relevant feature selection and unit allocation to clusters can be solved separately, being both problems solvable in polynomial time. One of the two, called q-vars for its similarity with the k-means, is the one that performs better.

The second part of the paper considers the application of the q-vars algorithm to clustering, following the same procedure tested in Brusco (2004). We simulate some data-sets in which statistic units are divided into clusters and some variables are masking. Our purpose is to discover the hidden clusters after having discarded the masking variables. The clustering algorithm is composed of three steps. In the first step, we apply a clustering algorithm to the whole data-set to determine potential cluster centres, necessary as input of the q-vars algorithm. Next, in the second step, the q-vars algorithm selects the relevant features and it is compared to other variable selection algorithms proposed in the literature. Finally, in the third step, the clustering algorithm is applied again to the data-set now composed of the relevant variables only. As will be seen, clustering with the q-vars algorithm is the most accurate procedure, and, when paired with the k-means clustering, the computational times are very fast.

2. Problem formulation

Let a clustering problem be defined on the set $U = \{1, \dots, n\}$ of objects (or units), for which the variables (or features) $V = \{1, \dots, m\}$ are recorded. Some of the variables of V are relevant, in the sense that objects belonging to different groups take different values on these variables, but some other variables are masking, which means that their values are not relevant for group membership. Suppose that cluster centres $R = \{1, \dots, r\}$ have been given by some preliminary analysis, the search for the optimal clustering can be improved by discarding the masking variables: What is the set of variables $Q \subseteq V$ that best discriminates the units group membership for the given set R of cluster centres? When the problem is solved, Q is the set of the relevant variables and $V \setminus Q$ is the set of the masking variables that are discarded from the data-set.

We will use the following notation. For every $i \in U$, $k \in R$, $j \in V$, let d_{ijk} be the dissimilarity (or distance) between i and k, measured through variable j. The dissimilarity between i and k is $d_{ik}(V) = \sum_{j \in V} d_{ijk}$. If only a subset $Q \subseteq V$ of variables is selected, then the dissimilarity between unit i and centre k is $d_{ik}(Q) = \sum_{j \in Q} d_{ijk}$.

The allocation (or membership) of the objects of U to one of the centres of R is determined by the shortest distance: For a given $Q \subset V$, a unit i is assigned to the centre $k(i)$ such that $d_{i,k(i)}(Q) = \min\{d_{ik}(Q) \mid k = 1, \ldots, r\}$. Let $D(Q) = \sum_i d_{i,k(i)}(Q)$ be the sum of all distances between units and centres. To select the best variables for clustering, the researcher finds the set Q for which the index $D(Q)$ is minimised, with the additional constraint $|Q| = q$ (q is a parameter that is exogenously fixed). This new combinatorial optimisation problem will be called *the q-variable selection problem* and its properties are discussed in the following sections. To visualise the relevant decisions, observe that the data can be represented as a bipartite graph $G = (U, R, E)$ in which objects $i \in U$ and centres $k \in R$ are regarded as nodes, while arcs represent features. More formally, there are m (multiple) arcs between every node pair i and k, where each arc $e_{ijk} \in E$ corresponds to feature $j = 1, \ldots, m$, with cost d_{ijk}.

Minimising the objective function $D(Q)$ for variable selection has been proposed in Benati and García (2014) and Friedman and Meulman (2004). It is worth noting that the model we are discussing here assumes that centres R are given without any assumption about their quality, that is, it may happen that the data do not have any cluster structure at all, or that different clusters can be detected with different sets of variables. In this sense, the choice of R can be made using different approaches. In Benati and García (2014) a full-fledged model in which decisions are the relevant variables Q and the optimal centres R has been formulated, but it was reported to be very difficult to solve to optimality: For a fixed Q, one has to solve the p-median problem, which is itself an NP-complete problem. The q-variable selection problem developed here can be considered a simplification of that model because the centres R are now fixed. Unfortunately, the problem remains NP-complete even in this case, as we prove later in Theorem 1. First, the problem must be formulated as a decision problem:

[q-variable selection]: Given a distance matrix $D \in \mathbb{R}^{n \times m \times r}$ and a non-negative real number α, is there any variable selection Q, such that the resulting cluster assignment has a value $D(Q) = g^* \leq \alpha$?

The complexity proof uses the following problem: Consider a bipartite graph $G = (U, R, E)$ in which $i \in U$, $|U| = n$, and $k \in R$, $|R| = m$, are the sets of nodes, and there is an arc e_{ik} with cost c_{ik} for all $i \in U$, $k \in R$. For a set $P \subseteq R$, $|P| = p$, called p-median, the distance from $i \in U$ to P is $c_{i,P} = \min\{c_{ik} \mid k \in P\}$, the value of the objective function is $F(P) = \sum_{i \in U} c_{i,P}$. The p-median problem is: $\min_{\substack{P \subseteq R \\ |P| = p}} F(P)$. This problem is known to be NP-complete (Kariv & Hakimi, 1979):

[p-median]: Given a cost matrix $C \in \mathbb{R}^{n \times m}$, and a non-negative real number α, is there any p-median of value $F(P) = v^* \leq \alpha$?

Theorem 1: *The q-variable selection problem is NP -complete.*

Proof: Checking whether a given solution Q is such that the objective function has value $g^* \leq \alpha$ can be done in polynomial time. Therefore the problem is in NP. To see NP-completeness, we will show that the p-median problem can be reduced to q-variable selection with $p = q$.

Given a p-median problem with cost matrix $C \in \mathbb{R}^{n \times m}$, then the following q-variable selection problem with cost matrix $D \in \mathbb{R}^{n \times m \times m}$ is defined on the auxiliary bipartite graph $G = (U, R, E)$, in which $i \in U = \{1, \ldots, n\}$ stands for units and $k \in R = \{1, \ldots, m\}$ stands for cluster centres. For every $i \in U$, $k \in R$ there are m arcs $e_{ijk} \in E$ that are indexed by $j \in V = \{1, \ldots, m\}$ and whose weights are $d_{ijk} = c_{ik}$ if $j = k$ and $d_{ijk} = M$ otherwise (with M a suitable large number). The structure of the three-dimensional matrix D is as follows (where d^k is the distance matrix from $i \in U$, $k \in R$):

$$\begin{array}{c|cccc|cccc|c|cccc|}
\multicolumn{1}{c}{} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{d^1}} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{d^2}} & \multicolumn{1}{c}{\cdots} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{d^m}} \\
1 & c_{11} & M & \ldots & M & M & c_{12} & \ldots & M & \ldots & M & \ldots & M & c_{1m} \\
2 & c_{21} & M & \ldots & M & M & c_{22} & \ldots & M & \ldots & M & \ldots & M & c_{2m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
n & c_{n1} & M & \ldots & M & M & c_{n2} & \ldots & M & \ldots & M & \ldots & M & c_{nm}
\end{array}$$

Now consider the problem of q-variable selection in this graph with $q = p$. Whenever a set $Q \subseteq V$, $|Q| = p$, is selected, distances from $i \in U$ to $k \in R$ can be calculated and each unit must be assigned to the closest cluster centre. The distance between unit $i \in U$ and a cluster centre $k \in R$ according to the selected variables in Q is $d_{ik} = d_{ijj} + (p-1)M = c_{ik} + (p-1)M$ if $j = k \in Q$, $d_{ij} = pM$ otherwise. Therefore, when solving the q-variable selection problem, it cannot happen that a unit i is assigned to a cluster centre k such that the corresponding variable index $k \notin Q$ because $c_{ik} + (p-1)M < pM$. So, i is assigned to that k for which $d_{ik} = \min\{d_{iw} \mid w \in Q\} = \min\{c_{iw} + (p-1)M \mid w \in Q\} = \min\{c_{iw} \mid w \in Q\} + (p-1)M$. Thus, the objective function value for the variable selection problem is $\sum_i \min\{c_{iw} \mid w \in Q\} + n(p-1)M$. But the objective function of the p-median problem with cost matrix C and set Q of medians is $\sum_i \min\{c_{iw} \mid w \in Q\}$. Now, if $q = p$, then the p-median problem has a solution of value $F(Q) = v^* < \alpha$ if and only if the q-variable selection problem on the auxiliary graph has a solution of value $D(Q) = g^* < \alpha + n(p-1)M$. □

2.1. Variable selection and variability reduction

The index $D(Q)$ is closely related to the representation of the variability within clusters: If variables are standardised using the z-score, dissimilarities d_{ijk} are squared distances, and centroids R are calculated as cluster means, then minimising the objective function $D(Q)$ is equivalent to the minimisation of the within-group variability, which is the same than maximising variability between groups. The result is mentioned in Andrews and McNicholas (2014) and Steinley and Brusco (2008a), but no formal proof is provided.

Let s_{ij} be the value of feature j, $j = 1, \ldots, m$, recorded for unit $i, i = 1, \ldots, n$, and let $\mu_j = \frac{1}{n}\sum_{i=1}^{n} s_{ij}$ be the average of j. The total variability brought by feature j is expressed by the sum of squares:

$$\mathrm{TSS}_j = \sum_{i=1}^{n} (s_{ij} - \mu_j)^2.$$

Let $\mathrm{TSS}(Q) = \sum_{j\in Q} \mathrm{TSS}_j$ be the total variability that is covered by a feature subset $Q \subseteq V$.

Lemma 1: *$TSS(Q)$ is constant for all $Q \subseteq V$ such that $|Q| = q$ if, and only if, every feature j has the same variance σ^2.*

Proof: If all features have the same variance $\sigma_j^2 = \sigma^2$ for all $j \in V$, then $TSS_j = \sum_{i=1}^{n} (s_{ij} - \mu_j)^2 = n\sigma^2$. Therefore, $\sum_{j\in Q} TSS_j = q\, n\, \sigma^2$.

Consider now the case in which units are partitioned into clusters G_k, $k = 1, \ldots, r$, and feature j for each cluster centre is represented by its mean, that is, $r_{kj} = \frac{1}{|G_k|}\sum_{i\in G_k} s_{ij}$ for $j = 1, \ldots, m;\ \ k = 1, \ldots, r$. Let $k(i)$ be the cluster to which unit i is assigned. For the given partition, the total sum of squares is:

$$\sum_{i=1}^{n} (s_{ij} - \mu_j)^2 = \sum_{i=1}^{n} (s_{ij} - r_{k(i),j})^2 + \sum_{i=1}^{n} (r_{k(i),j} - \mu_j)^2 + 2\sum_{i=1}^{n} (s_{ij} - r_{k(i),j})(r_{k(i),j} - \mu_j).$$

After some arithmetic manipulation, it can be seen that the term $\sum_{i=1}^{n} (s_{ij} - r_{k(i),j})(r_{k(i),j} - \mu_j)$ is null. Therefore the total sum of squares can be decomposed into two terms:

$$\mathrm{WSS}_j = \sum_{i=1}^{n} (s_{ij} - r_{k(i),j})^2,$$

which is the variability within clusters, and a second term:

$$\mathrm{CSS}_j = \sum_{i=1}^{n} (r_{k(i),j} - \mu_j)^2,$$

which represents the variability between clusters. If the researcher is free to choose a set $Q \subseteq V$ of variables, then the variability decomposition depends on the set Q according to the formula:

$$\sum_{j\in Q} \mathrm{TSS}_j = \sum_{j\in Q} \mathrm{WSS}_j + \sum_{j\in Q} \mathrm{CSS}_j. \tag{1}$$

As can be seen, calculating $\min_{Q\subseteq V} \sum_{j\in Q} \mathrm{WSS}_j$ with Q a set of fixed size q does not correspond, in general, to calculating $\max_{Q\subseteq V} \sum_{j\in Q} \mathrm{CSS}_j$, as the term $\sum_{j\in Q} \mathrm{TSS}_j$ depends on Q. The two problems are equivalent only in the special case that variables V are standardised so that all have the same variance. □

Theorem 2: *Assume that variables V are measured for units U and that all the variables have the same variance $\sigma_j^2 = \sigma^2$ for all $j \in V$. Let the units be partitioned into clusters G_k, $k = 1, \ldots, r$, and let R be the set of cluster centres calculated as the means of the clusters. Solve the q-variable selection problem with $d_{ijk} = (s_{ij} - r_{k(i),j})^2$. If an optimal solution Q^* is obtained, such that $\sum_{j\in Q^*} WSS_j = \min_{\substack{Q\subseteq V\\ |Q|=q}} \sum_{j\in Q} WSS_j$, then $\sum_{j\in Q^*} CSS_j = \max_{\substack{Q\subseteq V\\ |Q|=q}} \sum_{j\in Q} CSS_j$.*

Proof: It has already been shown in the proof of Lemma 1 that $\sum_{j\in Q} \mathrm{TSS}_j = qn\sigma^2$. Since this expression is constant, from Equation (1) we have that $\sum_{j\in Q} \mathrm{CSS}_j = qn\sigma^2 - \sum_{j\in Q} \mathrm{WSS}_j$, therefore minimising the latter is the same than maximising the former. □

2.2. Linear programming formulations for the q-variable selection problem

In this section, we propose two integer linear programming formulations. The first model represents the natural implementation of the q-variable selection problem and it has decision variables for both unit-to-cluster assignments and variable selection. The formulation is flexible enough to allow for the insertion of additional statistical constraints like outlier detection, conflicting variables, etc. The model requires a quadratic number of binary variables, which correspond to the assignments. Then a second model is proposed in which assignment variables are replaced with radius variables, which reduces the number of binary variables from quadratic to linear. It is worth noting that this radius reformulation was already implicit in a selection/clustering model discussed in Benati and García (2014) where the decisions were both variable selection and centre location. It is included here for the sake of completeness to show the integer linear model with the best computational times (even for the restricted case with fixed centres).

The decision variables of the first model are:

- z_j, $j = 1, \ldots, m$, represents whether feature j is chosen or not, that is, $z_j = 1$ if, and only if, $j \in Q$, $z_j = 0$ otherwise;
- x_{ik}, $i \in U$, $k \in R$ are the (global) assignment variables of unit i to cluster centre k, that is, $x_{ik} = 1$ if, and only if, unit i is assigned to cluster k, $x_{ik} = 0$ otherwise;

- w_{ijk}, $i \in U$, $j \in V$, $k \in R$, are the auxiliary (local) assignment variables of unit i to cluster centre k using feature j, that is, $w_{ijk} = 1$ if, and only if, unit i is assigned to cluster centre k and feature j is chosen, $w_{ijk} = 0$ otherwise.

The problem formulation is:

$$P_1 : f(z, x, w) = \min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{r} d_{ijk} w_{ijk} \tag{2}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} w_{ijk} = q x_{ik} \quad \forall i,\ \forall k, \tag{3}$$

$$\sum_{k=1}^{r} x_{ik} = 1 \quad \forall\, i, \tag{4}$$

$$\sum_{k=1}^{r} w_{ijk} \le z_j \quad \forall\, i, \forall j, \tag{5}$$

$$\sum_{j=1}^{m} z_j = q, \tag{6}$$

$$w_{ijk} \in \{0, 1\} \quad \forall\, i, \forall j, \tag{7}$$

$$x_{ij} \in \{0, 1\} \quad \forall\, i,\ \forall j, \tag{8}$$

$$z_j \in \{0, 1\} \quad \forall j. \tag{9}$$

Constraint (3) states that no local assignment (i, k) that uses variable j is feasible unless a global assignment (i, k) is established. Moreover, the total number of local assignments (i, k) is exactly q. Constraint (4) establishes that every unit i must be assigned to exactly one cluster k. Constraint (5) imposes that a local assignment (i, k) that uses variable j is feasible only if variable j has been selected. Constraint (6) sets the number of variables to q. Constraint (8) imposes binary values to the x assignment variables, while constraints (7) and (9)can be weakened to require that the variables are continuous (see Theorem 3).

One of the advantages of formulating the problem as an MILP model is that it can handle additional side constraints that the researcher may need to impose. In the following some examples are given, regarding:

- Constraining total variability;
- Restricting covariances;
- Balancing clusters cardinality;
- Discarding outliers.

As shown in Theorem 2, the dual relation between variability within clusters and variability between clusters holds only under restricted conditions, that is, data must have equal variances. For the cases in which variances are not equal, minimising $\sum_{k\in Q} \text{WSS}_k$ may result in selecting variables for which TSS_k is small as well. Therefore researchers may require to balance two objectives: on the one hand minimising $\sum_{k\in Q} \text{WSS}_k$, on the other hand maximising $\sum_{k\in Q} \text{CSS}_k$. In Steinley and Brusco (2008a) the two objectives are combined through their ratio, but, as it is common in bi-objective decision making, the ratio can be simplified imposing a bound on $\sum_{k\in Q} \text{CSS}_k$. Let K be a parameter that is chosen by the researcher. Then the following constraint can be added to P_1:

$$\sum_{k=1}^{m} \text{CSS}_k z_k \ge K. \tag{10}$$

By varying K researchers can observe a whole range of solutions from which to single out the best one for their purposes.

Sometimes researchers want to select variables that are not correlated (see Andrews & McNicholas, 2014; Fraiman, Justel, & Svarc, 2008). For example, in Andrews and McNicholas (2014) variables are discarded using the following rule. If ρ_{ij} is the correlation between variables i and j, then check if:

$$|\rho_{ij}| \le 1 - (\min\{\text{WSS}_i, \text{WSS}_j\})^p \tag{11}$$

with $p = 1, 2, \ldots, 5$. If the inequality is false, then impose that only one of the variables i and j can be selected. This procedure can be modelled with the following constraints. First, inequalities (11) are checked and then, for every incompatible pair $(i\,, j)$, the following constraint is added to P_1:

$$z_i + z_j \le 1. \tag{12}$$

Some other applications require that clusters are balanced. For example, the objective function in Friedman and Meulman (2004) encourages clusters of similar or equal size. This request can be formulated by allowing cluster cardinality between a given range, let us say between l and u. Then, for all j, the following constraint is added to P_1:

$$l \le \sum_{i\in U} x_{ij} \le u. \tag{13}$$

Sometimes researchers want to circumvent the effects of outliers, for example, using the so-called trimmed k-means (García-Escudero, Gordaliza, & Matrán 2003). The trimmed k-means algorithm is like the standard k-means, but a percentage α of the farthest statistic units is discarded from the computation of the means. Even in this application, it can be required that variable selection is not affected by outliers, trimming the α percent of the farthest units. This can be done by adding the following constraint to P_1:

$$\sum_{i\in U} \sum_{j\in R} x_{ij} = \lfloor n(1-\alpha) \rfloor \tag{14}$$

and turning constraint (4) into an inequality. Alternatively, outliers can be discarded if their distance to R exceeds a given threshold D.

Other constraints that can be imposed to clustering are discussed in Caballero et al. (2011). All the constraints proposed here are linear and can be added to P_1 without increasing its theoretical difficulty. Therefore, it is likely that the algorithms that are proposed for P_1 can be straightforwardly extended to problems involving these additional constraints.

The formulation P_1 of the q-variable selection problem requires an MILP model with a quadratic number of binary variables, which are the (i, j) assignments. This seems somewhat unnecessary as the natural decisions of the problem are the variables to select. It is worth exploring the possibility of an alternative formulation that has fewer binary variables. We will show that this can be done using the so-called radius formulation. The radius formulation is a technique that writes the objective function as a telescopic sum of terms. Since many of these terms are redundant when calculating the objective function, radius formulations usually contain fewer variables than the original problem. As a consequence, the problem is (usually) solved faster. The methodology was suggested long ago in Cornuejols, Nemhauser, and Wolsey (1980), but only recently has been widely applied (see AlBdaiwi, Ghosh, & Goldengorin, 2011; Avella, Sassano, & Vasil'ev, 2007; Benati & García, 2014; Elloumi, 2010; Elloumi, Labbé, & Pochet, 2004; García et al., 2011; García, Landete, & Marín, 2012; Marín, Nickel, Puerto, & Velten, 2009; Puerto, Ramos, & Rodríguez-Chía, 2013). The methodology is connected to pseudo-Boolean representation and data aggregation for the p-median problem (see AlBdaiwi et al., 2011; Church, 2003; Church, 2008).

The radius formulation replaces assignment variables w_{ijk} with radius variables h_{ijt}. Their definition requires the following steps. Consider any unit i and any variable j:

- **Step 1:** Remove multiplicities from $\{d_{ij1}, d_{ij2}, \ldots, d_{ijr}\}$ and sort the values in increasing order:

$$D_{ij1} < D_{ij2} < \cdots < D_{ij,g(i,j)},$$

where $g(i, j)$ is the number of different values that d_{ijk} assumes. In addition, define $D_{i0} = 0$. Note that if there is some null d_{ijk}, then $D_{ij1} = D_{ij0} = 0$, but this notation allows us to write always the same model, no matter whether D_{ij1} is zero or not.

- **Step 2:** Define binary variables h_{ijt} as follows:

$$h_{ijt} = \begin{cases} 1, \text{ if variable } j \text{ is selected and if unit } i \\ \quad \text{is allocated to a centre } k \\ \quad \text{such that } d_{ijk} \geq D_{ijt}; \\ 0, \text{ otherwise.} \end{cases}$$

Consider the example shown in Figure 1: $i \in U$, $\{u, v, o, l\} \in R$, $j \in V$. Distances are $d_{iju} = d_{ijv} = D_{ij1}$, because u and v are on the same circumference, and $d_{ijl} = d_{ijo} = D_{ij2}$, for the same reason. Then binary variables are h_{ij1} and h_{ij2}. If unit i is assigned to cluster centre u, then $h_{ij1} = 1$ and $h_{ij2} = 0$. If unit i is assigned to cluster l, then $h_{ij1} = 1$ and $h_{ij2} = 1$.

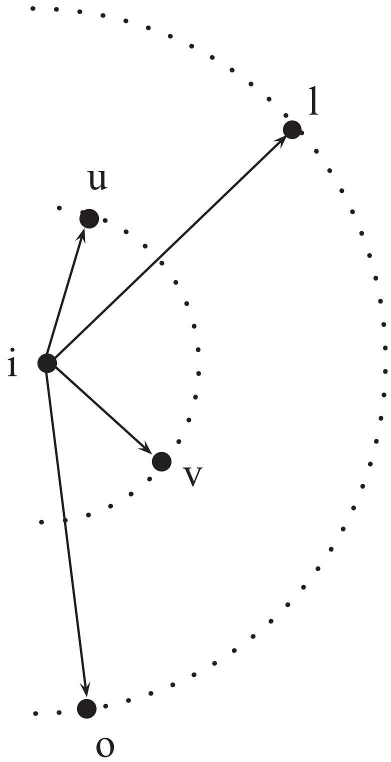

Figure 1. Radius description of equidistant points.

Radius variables can decrease the problem size: For given $i \in U$, $j \in V$ and constant $c \in \mathbb{R}$, tiers are defined as sets of archetypes at the same distance: $T_c^{ij} = \{k \in R \mid d_{ijk} = c\}$. In Step 1, if two archetypes $k, z \in T_c^{ij}$ for some c, then there is only one index u such that $D_{iju} = d_{ijk} = d_{ijz}$. Therefore, if there are many equal distances, as occurs when variables are ordinal, then the model reduction can be substantial.

For a pair $i \in U$, $j \in V$, the corresponding term of the objective function must be rewritten in telescopic form:

$$\sum_{t=1}^{g(i,j)} (D_{ijt} - D_{ij,t-1})h_{ijt} = \sum_{k=1}^{r} d_{ijk} w_{ijk}$$

and the overall problem is:

$$P_2 : \min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{t=1}^{g(i,j)} (D_{ijt} - D_{ij,t-1})h_{ijt} \tag{15}$$

$$\text{s.t.} \sum_{k=1}^{r} x_{ik} = 1 \quad \forall i, \tag{16}$$

$$\sum_{j=1}^{m} z_j = q, \tag{17}$$

$$h_{ijt} + \sum_{\{k \mid d_{ijk} < D_{ijt}\}} x_{ik} \geq z_j, \quad \forall\, i, \forall j, \forall t \geq 1, \tag{18}$$

$$h_{ijt} \geq 0 \quad \forall\, i, \forall j, \forall t, \tag{19}$$

$$x_{ik} \geq 0 \quad \forall\, i, \forall k, \tag{20}$$

$$z_j \in \{0, 1\} \quad \forall k. \tag{21}$$

Constraint (18) ensures that a radius variable h_{ijt} takes value one if $z_j = 1$ and $x_{ik} = 0$ for all k such that $d_{ijk} < D_{ijt}$. Regarding the continuous bound on x and h, that is, constraints (19) and (20), it can be seen that for the binary vector z, the problem decomposes into n independent problems, one for each j. If $z_j = 0$ for some j, then $h_{ijt} = 0$ for all i, t, as the problem is in minimisation form and the coefficients in the objective function are positive. If $z_j = 1$, then for any i there is some optimal solution x_{ik}, $k = 1, \ldots, r$, that takes values in $\{0, 1\}$. In order to see this, suppose that we have a fractional solution. Then there are at least two indices a and b such that x_{ia} and x_{ib} are fractional. For pair (i, j), if $d_{ija} = d_{ijb}$, then centres a and b are at the same distance from i. It follows that the fractional solution can be turned into an integer solution without affecting the value of the objective function (for example, by doing $x_{ia} = 1$ and $x_{ib} = 0$). If $d_{ija} < d_{ijb}$, then let $D_{ijw} = d_{ija} < D_{ijq} = d_{ijb}$ for some w and q. In order to simplify the writing of the proof, we assume without loss of generality that for unit i and variable j there is no allocation at a distance smaller than D_{ijw} and that a and b are the only cluster centres at exactly distances d_{ija} and d_{ijb}, respectively. From constraints (16) and (18), it follows that $h_{ijt} = 1$ if $t \leq w$, $h_{ijt} = 1 - x_{ia}$ if $w < t \leq q$, and that $h_{ijt} = 1 - x_{ia} - x_{ib}$ if $t > q$. If we substitute these h values in the objective function, then we obtain the value $D_{ijw} + (D_{ij,q+1} - D_{ijw})(1 - x_{ia}) - (D_{ij,q+1} - D_{ijq})x_{ib}$. This value can be reduced by reducing x_{ib} to zero and by increasing x_{ia} by the same amount.

2.3. Heuristic methods for variable selection

There are applications in which the q-variable selection problem needs to be solved several times (for example, when different values of q are tested). However, the NP-completeness theorem implies that computational time increases exponentially with data size. This means that the problem can only be solved within a reasonable time for instances of a limited size. The largest instances must be solved with heuristic algorithms. To develop these algorithms, it can be observed that the complexity of the problem derives from the fact that optimal variables z and optimal assignments x need to be calculated simultaneously, but that, if solved separately, then both of them are polynomially solvable problems.

Theorem 3: *If x^* is a feasible assignment solution for problem P_1, then the optimal solution for $\min_{z,w} f(z, x^*, w)$ can be calculated in polynomial time. Moreover, there is an optimal solution for which z and w have binary values.*

Proof: For a feasible assignment solution x^*, let $k(i)$ be the cluster centre k for which $x^*_{i,k(i)} = 1$. Then problem P_1 reduces to the following problem:

$$\min \sum_{i=1}^{n} \left(\sum_{j=1}^{m} d_{i,j,k(i)} w_{i,j,k(i)} \right) \tag{22}$$

$$\text{s.t.} \sum_{j=1}^{m} w_{i,j,k(i)} = q \quad \forall i / x_{i,k(i)} = 1, \tag{23}$$

$$w_{i,j,k(i)} \leq z_j \quad \forall\, i, \forall j, \tag{24}$$

$$\sum_{j=1}^{m} z_j = q, \tag{25}$$

$$0 \leq w_{i,j,k(i)} \leq 1 \quad \forall\, i, \forall j, \tag{26}$$

$$0 \leq z_j \leq 1 \quad \forall j. \tag{27}$$

Due to (23) and (25), every constraint (24) is satisfied as an equality as can be seen by summing both sides over j: $q = \sum_{j=1}^{m} w_{i,j,k(i)} \leq \sum_{j=1}^{m} z_j = q$. Using this property, the objective function can be written as $\sum_{j=1}^{m} \left(\sum_{i=1}^{n} d_{i,j,k(i)} \right) z_j$, which depends only on the cardinality constraint (25) and vector z. It can be shown that there is always a solution for which z takes integer values. Let $b_j = \sum_{i=1}^{n} d_{i,j,k(i)}$ be the total distance from units to centres using variables j. Then rank values b_j in increasing order: $b_{j(1)} \leq b_{j(2)} \leq \cdots \leq b_{j(m)}$. If we now choose $z_{j(t)} = 1$ for $t = 1, \ldots, q$, then we have an optimal solution. Finally, for integer values of z and x, there is a solution w with integer values because (24) holds as an equality. □

If the vector of variables z is fixed, then distances (i, k) are easily calculated to find the optimal assignments of unit i to the closest cluster centre k. If the vector of assignments x is fixed, then the optimal variables z are calculated using the distance ranking, as shown in the proof of Theorem 3. This observation suggests that a heuristic procedure can alternate between the two subroutines: Start with some fixed z; find the corresponding optimal assignment x. Then, calculate the optimal z for that given x, and repeat until the solution does not improve any more. We use the notation $D_Q(X)$ to denote the value of the objective function when Q is fixed (subroutine input) and X is the decision variable (subroutine output); the notation $D_X(Q)$ is defined similarly. Here are the details of these two subroutines:

Subroutine Best-Assignment:

- Input: The set $Q \subseteq V$.
- Output: The assignment matrix X and the value of the objective function $D_Q(X)$.
- Step 1: For all $i \in U$, $k \in R$, let $c_{ik} = \sum_{j \in Q} d_{ijk}$,

- Step 2: For all $i \in U$, let $x_{iw} = 1$ if $c_{iw} = \min\{c_{ik} | 1 \leq k \leq r\}$; $x_{iw} = 0$ otherwise.
- Step 3: Let $D_Q(X) = \sum_{i \in U} \sum_{k \in R} c_{ik} x_{ik}$.

The next subroutine calculates an optimal variable set Q for a given allocation to clusters X.

Subroutine Best-Variables:

- Input: The assignments matrix X.
- Output: The variables $Q \subseteq V$, objective function $C_X(Q)$.
- Step 1: For all $j \in V$, let $b_j = \sum_{i,k} d_{ijk} x_{ik}$.
- Step 2: Rank b_j in increasing order: $b_{j(1)} \leq \cdots \leq b_{j(m)}$.
- Step 3: Let $j(i) \in Q$ if, and only if, $i \leq q$.
- Step 4: Let $C_X(Q) = \sum_{i=1}^{q} b_{j(i)}$.

The two subroutines (and Theorem 3) show that there is a decomposition principle at work here. The variable selection problem is NP-complete because one has to decide concurrently which variables to select and to which cluster assign the units to. But if we separate the two decisions, then we obtain two polynomially solvable problems. Using subroutines Best-Assignments and Best-Variables one can start with a tentative variable set Q^0 and calculate the corresponding best assignment X^0 using subroutine Best-Variable. Then, for the assignment X^0, one calculates the optimal variables Q^1 using subroutine Best-Variable. If $Q^1 \neq Q^0$, then a new assignment X^1 is calculated until, for some t, one has that $Q^t = Q^{t-1}$. In this case, we say that the algorithm converged. There is always convergence because, as shown in Theorem 3, all subroutines calculate optimal values, so that a non-increasing sequence of objective values is obtained: $D_X(Q^0) \geq D_Q(X^0) \geq D_X(Q^1) \geq \cdots \geq D_X(Q^{t-1}) = D_{X^t}(Q)$. Since Q and X are discrete sets, then the sequence converges in a finite, although potentially exponential, number of steps. It is worth noting that this decomposition principle is very similar to the one that is used by the k-means method for clustering: k-means alternates assignments and cluster centres until a local optimum is reached (see Chen et al., 2004; Hartigan & Wong, 1979; MacQueen, 1967).

We take the advantage of the similarity with the k-means and we name our variable selection algorithm q-vars. It starts with a random selection of variables and then optimal assignments and variables are calculated alternately. When a local optimum is found, the procedure is repeated with a new random selection of variables, Random Restart, as it is common practice in the standard implementations of the k-means algorithm found in the literature.

The q-vars Algorithm:

- Initialisation: Objective function $C^{\text{best}} = +\infty$, variables $Q^{\text{best}} = \emptyset$, random start counter $s = 1$, maximum number of random start: $s^{\max} = M$.
- Step 1: Random Start: Select randomly a set of variables Q^0 and let $t := 0$.
- Repeat until a local optimum is found, that is, $C_{Q^t}(X^t) = C_{X^t}(Q^{t+1})$.
  - Step 2: (Unit Allocation) For given Q^t, call Best-Assignments to calculate optimal X^t and $C_{Q^t}(X^t)$.
  - Step 3: (Variable Selection) For given X^t, call Best-Variables to calculate optimal Q^{t+1} and $C_{X^t}(Q^{t+1})$ and update $t := t + 1$.
- Step 4: $C^{\text{best}} = \min\{C^{\text{best}}, C_{Q^t}(X^t)\}$, update Q^{best} accordingly.
- Step 5: $s = s + 1$. If $s \leq s^{\max}$, then return to Step 1.

The next method is called Add-and-Drop, as it seeks the optimal solution by adding and removing variables from an incumbent set. It starts with a solution set $Q \subseteq V$. Then, if the objective function decreases, a new variable from $V - Q$ is added to Q and one variable is removed from Q. The process is repeated until no further improvement is found, that is, until a local optimum, say Q^t, is reached. Add-and-Drop, has been successfully applied to the p-median problem (see Mladenovic, Brimberg, Hansen, & Moreno-Pérez, 2007) and it is described next:

The Add-and-Drop Algorithm

- Initialisation: $D^{\text{best}} = D(Q^0) = +\infty$, $Q^{\text{best}} = Q^0 = \emptyset$, random start counter $s = 1$, maximum number of random start: $s^{\max} = M$.
- Step 1, (Random Start): Select randomly a set of variables Q^1 and let $t := 1$.
- Repeat until $D(Q^t) = D(Q^{t-1})$:
  - Step 2: (Add) Calculate $D(Q^t \cup \{i^*\}) = \min_{i \notin Q^t} D(Q^t \cup \{i\})$.
  - Step 3: (Drop) Calculate $D(Q^t \cup \{i^*\} - \{j^*\}) = \min_{j \notin Q^t; j \neq i^*} D(Q^t \cup \{i^*\} - \{j\}))$. Update $Q^{t+1} = Q^t \cup \{i^*\} - \{j^*\}$ and $t := t + 1$.
- Step 4: $D^{\text{best}} = \min\{D^{\text{best}}, D_{Q^t}(X^t)\}$, update Q^{best} accordingly.
- Step 5: $s = s + 1$; if $s \leq s^{\max}$ return to Step 1.

2.4. Computational tests

Here, we compare the two MILP formulations and the two heuristic methods. The best of the four methods will be used in Section 3 in a large-scale simulation in which variable selection is used in conjunction with clustering. The experiments of this subsection are inspired by the ones reported in Tadesse et al. (2005), Fraiman et al. (2008) and Law et al. (2004).

In the tables that report the experiments, the notation $n \times r \times m$ stands for the size of the input matrix. Tests regarding the comparison of MILP formulations are coded using CPLEX 12.6 (Concert Library) and run on an Intel Core i5-3470, double core (3.20 GHz each) with 8 GB RAM and Windows 64 bits. Tests regarding the comparison of heuristic methods are coded in Visual C++ 2010 and run on an Intel Pentium Dual CPU T3400 (2.16 GHz), 3 GB RAM.

The first experiment replicates the test carried out in Tadesse et al. (2005) and Fraiman et al. (2008) to validate algorithms for masking variables detection. Random data consist of 15 statistic units partitioned into 4 groups. Groups are described by multivariate normal densities with 20 true variables. Their theoretical distribution is:

$$s_{ij} = I_{1\le i\le 4}N(\mu_1, \sigma_1^2) + I_{5\le i\le 7}N(\mu_2, \sigma_2^2) \\ + I_{8\le i\le 13}N(\mu_3, \sigma_3^2) + I_{14\le i\le 15}N(\mu_4, \sigma_4^2),$$

where $I_.$ is the indicator function that takes value 1 if the condition is met and takes value 0 otherwise. Thus, the first four samples arise from the same first distribution, the next three from the second distribution, and so on. The distribution parameters are $\mu_1 = 5, \sigma_1^2 = 1.5$, $\mu_2 = 2, \sigma_2^2 = 0.1, \mu_3 = -3, \sigma_3^2 = 0.5, \mu_4 = -6, \sigma_4^2 = 2$. Additionally, for $m = 50, 100, 500, 1000$, which is also the number of binary variables in our models, $m - 20$ noisy variables are added to the data, generated with a $(0 - 1)$-uniform distribution. The experiment assumes that the cluster membership is known for each $i = 1, \ldots, n$, therefore, the arithmetic mean is used to establish the coordinates of the groups archetypes.

The results are reported in Table 1. A time limit of 7200 seconds is established for solving the MILP problems whereas the heuristic uses $s^{\max} = 100$. If the time limit is reached without having obtained optimality, then the best solution found so far is retained. For all algorithms, column "fo" reports the value of the objective function. For the heuristics, we report the iteration in which the best solution is found in column " it-best" and the computational time is given in column "time-best". For the MILP models, we report the value of the continuous relaxation at the root node of the branch-and-bound tree (" root LP") and the total needed time in column "time".

The first conclusion is that the q-vars method is able to calculate always the optimal objective value in very few iterations, both when q is fixed to the true value of 20 and when q is erroneously fixed to 10 or 40. In comparison, the other heuristic algorithm Add-and-Drop is much slower and it finds the optimal value only for q is 20. Regarding the LP models, we see that computational times are negligible, except when $m \geq 500$ and $q = 40$, with only one problem unsolved within the time limit of two hours by the radius formulation. In many cases the value of the linear relaxation calculated at the root node of the radius formulation P_2 is the optimal value, while for P_1 it is around 8% less than the optimum. Moreover, the computational times of P_2 are the best for low values of m, but they are more sensitive to the problem scale than formulation P_1: As can be seen, when $q = 40$ and $m > 500$, the linear relaxation at the root node is better in formulation P_2 than in formulation P_1, but the computational times are lower for P_1. Regarding the statistical ability to recover the true masking variables and the correct (i, k) assignments, the optimal solution is such that the 20 true variables are always selected, or a subset of them if $q = 10$. Empiric assignments are the same as the true.

The following test is analogous to the Trunk data test mentioned in Law et al. (2004) and it is an example of dimensionality reduction. The goal is to test the ability of the model to reduce the dimension of the sample when all variables are important, but some variables are more discriminating than others. Statistic units belong to two groups G_k, $k = 1, 2$. For $j = 1, \ldots, m$, the measure of unit i belonging to group k is the outcome of a normal distribution $N(\mu_{kj}, \sigma_{kj})$, with $\sigma_{kj} = 1$ for all k, j, so that units are well separated if the difference between μ_{1j} and μ_{2j} is high. In order to control this feature, averages are $\mu_{1j} = 0$ for all j, $\mu_{2j} = 6 - 6\left(\frac{j}{m}\right)$, so that the most important variables are the ones with the lowest values of j. Therefore, the groups are well separated and become less and less distinguishable as j increases. The rest of the parameters are $n = 100$, $|G_1| = |G_2| = 50$, and $m = 40, 80, 120$.

Data reporting computational tests are provided in Table 2. Again, the q-vars is faster and more accurate than the Add-Drop heuristic. The radius formulation P_2 is better than P_1. With regard to the statistic quality of the solution, for fixed q the theoretical best solution is $z_k = 1$ if $k \leq q$ and $z_k = 0$ otherwise. Columns "inaccuracy" report the cardinality of the difference between the theoretical optimal variable set and the one calculated by the models: As can be seen, most of the time the experimental variable set is equal to the theoretical best set, with differences that are never more than a few units. Since random effects in data generation are unavoidable, it can be concluded that the model is very accurate in selecting the most relevant variables.

The previous experiments are encouraging, but it is anomalous that an NP-complete problem is solved so easily. Next, we carry out tests where the groups are not so well separated by generating difficult computational instances of the problem. The following experiment assumes surveys in which every variable j follows the (0–1)-uniform distribution. In this unstructured data, wrong clusters are superimposed in the form of random group membership $k(i)$ for units i. The other parameters are $n = 96$, $r = 8$ or 12, and $m = 40$ or 80. In Table 3, computational results are reported. The difficulty of this class of problems emerges because the MILP models are never able to calculate the optimal value within the time limit of 2 h. The reader should observe that Table 3 does not report "time" for the

Table 1. Computational results: variable selection with 20 true variables.

Problem name		*q*-vars			Add-Drop			ILP-P1			ILP-P2		
	q*	fo	it-best	time-best	fo	it-best	time-best	fo	root LP	time	fo or best UB	root LP	time
MSA-A-15x4x50	20	43.89	1	0	43.89	9	0	43.89	40.95	0	43.89	43.89	0
MSA-B-15x4x50	20	41.44	1	0	41.44	9	0	41.44	38.62	0	41.44	41.44	0
MSA-C-15x4x50	20	46.14	1	0	46.14	9	0	46.14	42.22	0	46.14	46.14	0
MSA-D-15x4x50	20	45.47	1	0	45.47	9	0	45.47	42.03	0	45.47	45.47	0
MSA-E-15x4x50	20	39.61	1	0	39.61	9	0	39.61	36.59	0	39.61	39.61	0
MSA-A-15x4x100	20	43.63	2	0	43.63	17	0	43.63	39.96	0	43.63	43.63	0
MSA-B-15x4x100	20	43.61	1	0	43.61	17	0	43.61	39.72	0	43.61	43.61	0
MSA-C-15x4x100	20	40.77	1	0	40.77	17	0	40.77	38.53	0	40.77	40.77	0
MSA-D-15x4x100	20	39.57	1	0	39.57	17	0	39.57	37.46	0	39.57	39.57	0
MSA-E-15x4x100	20	46.74	2	0	46.74	17	0	46.74	43.51	0	46.74	46.74	0
MSA-A-15x4x500	20	43.38	1	0	43.38	19	0	43.38	39.38	0	43.38	43.38	0
MSA-B-15x4x500	20	47.17	2	0	47.17	19	0	47.17	44.17	0	47.17	47.17	0
MSA-C-15x4x500	20	45.05	2	0	45.05	19	0	45.05	40.91	0	45.05	45.05	0
MSA-D-15x4x500	20	43.46	2	0	43.46	19	0	43.46	39.89	0	43.46	43.46	0
MSA-E-15x4x500	20	46.60	1	0	46.60	19	0	46.60	42.14	0	46.60	46.60	0
MSA-A-15x4x1000	20	44.34	2	0	44.34	20	0	44.34	40.21	2	44.34	44.34	0
MSA-B-15x4x1000	20	42.01	2	0	42.01	20	0	42.01	38.44	1	42.01	42.01	0
MSA-C-15x4x1000	20	41.25	2	0	41.25	20	0	41.25	38.76	1	41.25	41.25	0
MSA-D-15x4x1000	20	39.41	2	0	39.41	20	0	39.41	36.82	1	39.41	39.41	0
MSA-E-15x4x1000	20	38.59	2	0	38.59	20	0	38.59	36.58	1	38.59	38.59	0
Average	20	43.11			43.11			43.11	39.85		43.11	43.11	
MSA-A-15x4x50	10	18.12	1	0	21.23	547	0	18.12	17.44	0	18.12	18.12	0
MSA-B-15x4x50	10	17.18	1	0	18.05	580	0	17.18	15.77	0	17.18	17.18	0
MSA-C-15x4x50	10	18.94	1	0	20.23	1876	0	18.94	17.39	0	18.94	18.94	0
MSA-D-15x4x50	10	18.89	1	0	20.36	13	0	18.89	18.08	0	18.89	18.89	0
MSA-E-15x4x50	10	15.55	1	0	17.49	34	0	15.55	14.97	0	15.55	15.55	0
MSA-A-15x4x100	10	16.87	2	0	18.10	378	0	16.87	15.68	0	16.87	16.87	0
MSA-B-15x4x100	10	17.48	2	0	19.17	892	0	17.48	17.28	0	17.48	17.48	0
MSA-C-15x4x100	10	16.75	2	0	18.42	333	0	16.75	15.68	0	16.75	16.75	0
MSA-D-15x4x100	10	16.33	2	0	17.82	201	0	16.33	16.21	0	16.33	16.33	0
MSA-E-15x4x100	10	19.53	2	0	21.18	686	0	19.53	18.80	0	19.53	19.53	0
MSA-A-15x4x500	10	16.53	2	0	21.39	1154	0	16.53	15.85	0	16.53	16.53	0
MSA-B-15x4x500	10	19.15	2	0	21.87	994	0	19.15	18.70	0	19.15	19.15	0
MSA-C-15x4x500	10	18.16	2	0	19.94	530	0	18.16	17.12	0	18.16	18.16	0
MSA-D-15x4x500	10	17.96	2	0	20.42	226	0	17.96	16.74	0	17.96	17.96	0
MSA-E-15x4x500	10	19.69	2	0	23.35	540	0	19.69	18.25	0	19.69	19.69	0
MSA-A-15x4x1000	10	17.26	2	0	19.22	550	0	17.26	16.90	0	17.26	17.26	0
MSA-B-15x4x1000	10	18.01	2	0	19.24	765	0	18.01	16.46	1	18.01	18.01	0
MSA-C-15x4x1000	10	16.32	2	0	20.42	1010	0	16.32	15.89	0	16.32	16.32	0
MSA-D-15x4x1000	10	16.16	2	0	18.36	11	0	16.16	15.36	0	16.16	16.16	0
MSA-E-15x4x1000	10	17.08	2	0	17.98	600	0	17.08	16.29	1	17.08	17.08	0
Average		17.60			19.71			17.60	16.74		17.60	17.60	
MSA-A-15x4x50	40	247.86	1	0	247.86	634	1	247.86	154.60	0	247.86	247.86	0
MSA-B-15x4x50	40	235.34	1	0	235.34	180	0	235.34	146.05	0	235.34	235.34	0
MSA-C-15x4x50	40	247.30	1	0	247.30	520	1	247.30	159.87	0	247.30	247.30	0
MSA-D-15x4x50	40	241.20	1	0	241.20	980	1	241.20	149.30	0	241.20	241.20	0
MSA-E-15x4x50	40	239.69	1	0	239.69	308	0	239.69	149.89	0	239.69	239.69	0
MSA-A-15x4x100	40	230.18	1	0	238.29	384	1	230.18	139.79	2	230.18	230.18	0
MSA-B-15x4x100	40	226.71	1	0	236.00	102	0	226.71	140.97	1	226.71	226.71	0
MSA-C-15x4x100	40	221.81	1	0	227.80	273	0	221.81	138.54	1	221.81	221.81	0
MSA-D-15x4x100	40	217.69	1	0	225.45	2773	4	217.69	135.03	1	217.69	217.69	0
MSA-E-15x4x100	40	236.60	1	0	245.25	553	1	236.60	145.58	2	236.60	236.60	0
MSA-A-15x4x500	40	185.61	1	0	226.71	164	0	185.61	118.34	7	185.61	170.83	167
MSA-B-15x4x500	40	204.59	1	0	234.22	801	1	204.59	126.77	15	204.59	176.45	964
MSA-C-15x4x500	40	189.23	1	0	228.33	706	1	189.23	124.20	9	189.23	173.99	107
MSA-D-15x4x500	40	201.97	1	0	227.99	2069	4	201.97	116.75	15	201.97	171.78	1158
MSA-E-15x4x500	40	205.43	1	0	230.80	343	1	205.43	123.63	16	205.43	177.57	772
MSA-A-15x4x1000	40	187.79	2	0	229.06	198	0	187.79	115.45	38	187.79	159.05	1904
MSA-B-15x4x1000	40	183.91	1	0	222.83	914	2	183.91	109.07	30	184.03	156.41	7200
MSA-C-15x4x1000	40	180.67	1	0	220.27	77	0	180.67	108.39	49	180.67	154.69	2867
MSA-D-15x4x1000	40	181.34	1	0	222.50	847	2	181.34	111.17	48	181.34	153.37	4797
MSA-E-15x4x1000	40	173.76	1	0	219.92	186	0	173.76	11.82	21	173.76	153.49	988
Average		211.93			232.34			211.93	126.26		211.94	199.60	

MILP models (P_1 and P_2) since in all instances they required computational times larger that 7200 seconds and, even so, they were unable to find the optimal solution. Regarding heuristics, computational times and number of iterations increase considerably. The deduction from these findings is that the data of the problem affect the reliability of the algorithms. It is very likely that there are real applications, e.g. applications with overlapping groups or with high values of m, r, n, in which one cannot use the MILP models because they are not efficient (i.e. they cannot solve the problem within a reasonable time). For those cases, the heuristic q-vars should be used.

Table 2. Computational results: application to dimension reduction.

		q-vars				Add-Drop				ILP-P1			ILP-P2		
Problem name	q	fo	opt-it	time-opt	accuracy	fo	opt-it	time-opt	accuracy	fo	root LP	time	fo	root LP	time
PRV-A-100x2x40	10	119.14	1	0	3	119.78	133	0	2	119.14	117.79	0	119.14	119.14	0
PRV-B-100x2x40	10	130.29	1	0	2	130.29	569	0	2	130.29	126.00	0	130.29	130.29	0
PRV-C-100x2x40	10	128.01	1	0	0	128.01	6	0	0	128.01	124.37	0	128.01	128.01	0
PRV-D-100x2x40	10	121.92	1	0	1	121.92	7	0	1	121.92	118.84	0	121.92	121.92	0
PRV-E-100x2x40	10	120.67	1	0	0	120.67	6	0	0	120.67	120.48	0	120.67	120.67	0
PRV-A-100x2x40	20	357.23	1	0	1	357.23	11	0	1	357.23	324.51	0	357.23	357.23	0
PRV-B-100x2x40	20	345.93	1	0	1	345.93	11	0	1	345.93	325.88	0	345.93	345.93	0
PRV-C-100x2x40	20	352.89	1	0	1	352.89	88	0	1	352.89	322.01	0	352.89	352.89	0
PRV-D-100x2x40	20	334.82	1	0	1	334.82	11	0	1	334.82	308.16	0	334.82	334.82	0
PRV-E-100x2x40	20	357.06	1	0	0	357.06	11	0	0	357.06	331.69	0	357.06	357.06	0
PRV-A-100x2x40	30	827.52	1	0	2	827.52	25	0	2	827.52	641.14	1	827.52	827.52	0
PRV-B-100x2x40	30	782.85	1	0	0	782.85	7	0	0	782.85	637.04	1	782.85	782.85	0
PRV-C-100x2x40	30	799.33	1	0	0	799.33	7	0	0	799.33	628.64	1	799.33	799.33	0
PRV-D-100x2x40	30	787.40	1	0	0	787.40	7	0	0	787.40	624.59	1	787.40	787.40	0
PRV-E-100x2x40	30	818.74	1	0	0	818.74	7	0	0	818.74	643.29	1	818.74	818.74	0
Average		425.59				425.63				425.59	359.63		425.59	425.59	
PRV-A-100x2x80	20	263.31	1	0	2	263.71	80	0	1	263.31	257.02	0	263.31	263.31	0
PRV-B-100x2x80	20	266.87	1	0	1	266.87	168	0	1	266.87	260.95	0	266.87	266.87	0
PRV-C-100x2x80	20	255.95	1	0	2	255.95	14	0	2	255.95	252.64	0	255.95	255.95	0
PRV-D-100x2x80	20	263.48	1	0	2	263.48	216	0	2	263.48	256.38	0	263.48	263.48	0
PRV-E-100x2x80	20	245.06	1	0	2	246.80	34	0	1	245.06	239.28	0	245.06	245.06	0
PRV-A-100x2x80	40	727.85	1	0	1	727.85	24	0	1	727.85	665.12	1	727.85	727.85	0
PRV-B-100x2x80	40	719.68	1	0	2	719.68	92	1	2	719.68	666.79	1	719.68	719.68	0
PRV-C-100x2x80	40	729.66	1	0	1	729.66	23	0	1	729.66	676.67	1	729.66	729.66	0
PRV-D-100x2x80	40	726.86	1	0	1	726.86	129	1	1	726.86	652.20	1	726.86	726.86	0
PRV-E-100x2x80	40	690.03	1	0	1	690.03	128	1	1	690.03	634.69	0	690.03	690.03	0
PRV-A-100x2x80	60	1694.27	1	0	1	1694.27	279	4	1	1694.27	1311.62	5	1694.27	1694.27	0
PRV-B-100x2x80	60	1627.80	1	0	1	1627.80	15	0	1	1627.80	1297.12	5	1627.80	1627.80	0
PRV-C-100x2x80	60	1654.34	1	0	0	1654.34	15	0	0	1654.34	1319.34	4	1654.34	1654.34	0
PRV-D-100x2x80	60	1635.18	1	0	0	1635.18	15	0	0	1635.18	1289.34	4	1635.18	1635.18	0
PRV-E-100x2x80	60	1621.49	1	0	2	1621.49	31	0	2	1621.49	1279.15	9	1621.49	1621.49	0
Average		874.79				874.93				874.79	737.22		874.79	874.79	
PRV-A-100x2x120	30	390.01	1	0	4	391.07	471	2	3	390.01	381.819	0	390.01	390.01	0
PRV-B-100x2x120	30	386.92	1	0	5	388.78	1556	5	4	386.92	377.796	1	386.92	386.92	0
PRV-C-100x2x120	30	386.30	1	0	3	386.30	371	2	3	386.30	380.574	1	386.30	386.30	0
PRV-D-100x2x120	30	376.49	1	0	2	376.49	375	1	2	376.49	369.607	0	376.49	376.49	0
PRV-E-100x2x120	30	370.08	1	0	4	374.20	841	3	2	370.08	362.519	0	370.08	370.08	0
PRV-A-100x2x120	60	1040.36	1	0	3	1040.36	215	3	3	1040.36	970.97	1	1040.36	1040.36	0
PRV-B-100x2x120	60	1046.18	1	0	0	1046.18	26	0	0	1046.18	970.81	1	1046.18	1046.18	0
PRV-C-100x2x120	60	1085.35	1	0	2	1087.07	357	5	2	1085.35	990.176	1	1085.35	1085.35	0
PRV-D-100x2x120	60	1067.89	1	0	1	1070.13	206	3	2	1067.89	989.424	2	1067.89	1067.89	0
PRV-E-100x2x120	60	1025.97	1	0	1	1025.97	26	0	1	1025.97	964.7	1	1025.97	1025.97	0
PRV-A-100x2x120	90	2391.07	1	0	2	2391.07	19	1	2	2391.07	1904.8	11	2391.07	2391.07	0
PRV-B-100x2x120	90	2366.39	1	0	1	2366.39	44	1	1	2366.39	1888.17	6	2366.39	2366.39	0
PRV-C-100x2x120	90	2469.22	1	0	3	2469.22	19	1	3	2469.22	1941.87	12	2469.22	2469.22	0
PRV-D-100x2x120	90	2451.00	1	0	2	2451.00	240	8	2	2451.00	1929.39	11	2451.00	2451.00	0
PRV-E-100x2x120	90	2340.39	1	0	1	2340.39	20	6	1	2340.39	1886.16	10	2340.39	2340.39	0
Average		1279.57				1280.31				1279.57	1087.25		1279.57	1279.57	

Table 3. Computational results: variable selection to data with no structure.

		q-vars			Add-Drop			ILP-P1		ILP-P2	
Problem name	q	fo	opt-it	time-opt	fo	opt-it	time-opt	obj/best UB	root LP	obj/best UB	root LP
WRN-A-96x8x80	10	691.39	8821	2	730.58	2163	3	880.17	296.30	750.05	422.52
WRN-A-96x8x80	20	1548.05	24855	8	1613.55	551	2	1864.00	667.76	1669.25	1010.08
WRN-A-96x8x80	30	2435.64	10049	4	2513.95	318	3	2833.99	1078.13	2530.08	1740.70
WRN-A-96x8x80	40	3339.65	300	0	3414.16	54	1	3711.39	1522.24	3428.77	2599.22
WRN-B-96x12x40	10	669.73	16155	3	682.28	895	2	887.27	201.66	709.95	389.37
WRN-B-96x12x40	20	1532.12	3918	1	1558.96	259	2	1894.40	472.27	1545.83	1112.21
WRN-B-96x12x40	30	2435.38	4197	2	2443.20	1374	19	2963.43	788.62	2417.24	2144.23
Average		1807.42			1850.95			2147.81	718.14	1864.45	1345.48

As a conclusion, these tests suggest that *q*-vars is the most appropriate algorithm for the next simulations because:

- It is fast and performs better than the other heuristic Add-and-Drop.
- Most of the time, it calculates the optimal solution in just a few iterations (if compared with MILP solutions).
- There are problem instances in which MILP models are unable to find the optimal solution.

3. Clustering with variable selection

The main application of variable selection is clustering. In a broad view, the methodology to select variables and clusters is composed of three steps (see, for example, Steinley & Brusco, 2008a):

- Step 1: Standardise the data and run a clustering algorithm to estimate the cluster centres R.
- Step 2: Solve the variable selection problem to select the optimal set of variables Q.
- Step 3: Apply the clustering algorithm to the dataset in which only the selected variables Q are retained.

In Step 1, the data standardisation can be worked out in at least two ways. The most common approach is to apply the z-score so that all variables have variance equal to one. The second approach, proposed in Steinley and Brusco (2008a), is to modify the z-score using a correction factor, specific to each variable, with the purpose of preserving the so-called “clusterability index” of that variable. Here the method will be called c-score standardisation (to distinguish it from z-score). Details about its implementations are available in Steinley and Brusco (2008b) and Steinley and Brusco (2008a).

The variable selection problem (Step 2) is solved with the q-vars procedure (with parameter reduced to $s^{\max} = 40$). The choice of q-vars heuristic is dictated by the fact that in real applications q is an unknown parameter. The subroutine must run for different values of q before selecting the best one and therefore it is important to reduce its computational time, even at the cost of sacrificing the accuracy of the results.

The k-means and the Expectation–Maximisation (EM) algorithms are the most popular methods for clustering and they were also used in previous experiments in variable selection (see Andrews & McNicholas, 2014; Raftery & Dean, 2006; Steinley & Brusco, 2008a). Therefore, Steps 1 and 3 are experimented using these two alternatives. The k-means algorithm is the most popular clustering method in the literature by far, available in all packages and used in the experiments in Steinley and Brusco (2008a). Moreover, it does not assume that data should conform to any hypothesis. Conversely, the EM algorithm assumes that data are outcomes from a multivariate probability distribution which is the mixture of k components, each component describing one of the clusters (this is the reason why the method is called model-based clustering). The EM algorithm calculates the maximum of the likelihood function to estimate both cluster centres and cluster memberships (see McLachlan & Krishnan, 1997). The method can include the variable selection problem in the form of binary decisions (see Raftery & Dean, 2006), and it is used as the benchmark model in Andrews and McNicholas (2014).

Now we describe how to guess the exact number q^* of the relevant variables. The first method, proposed in Steinley and Brusco (2008a), decomposes the total variability of a data partition P into two terms, the within-groups and the between-groups variability and calculates an index called $\mathrm{VAF}(P, q)$. This index depends on the input value of q and on the partition P obtained by the clustering algorithm, but the specific number q^* is determined by comparing the slope of the function $\mathrm{VAF}(P, q)$ for different q. The second approach, proposed in Andrews and McNicholas (2014), takes advantage of the property that the EM algorithm outputs cluster memberships as probabilities, which are used to calculate an index of clustering uncertainty, called $\mathrm{UNC}(P, q)$. Then, the value of q^* is guessed as the minimum of $\mathrm{UNC}(P, q)$. In our experiments, the choice of which index to use is solved by considering the structure of the clustering algorithm. The mathematics of the k-means algorithm, with its sum of squares minimisation, is based on variability decomposition and thus $\mathrm{VAF}(P, q)$ seems the appropriate choice. Conversely, if one is using the EM-algorithm, then its outcomes are membership probabilities, so that the choice of $\mathrm{UNC}(P, q)$ is reasonable.

Following the discussion above, tests have been carried out with the following combinations of data standardisation and clustering algorithms:

- *qv-1*: z-score standardisation, k-means algorithm, $\mathrm{VAF}(P, q)$ used to guess q^*.
- *qv-2*: z-score standardisation, EM algorithm, $\mathrm{UNC}(P, q)$ used to guess q^*.
- *qv-3*: c-score standardisation, k-means algorithm, $\mathrm{VAF}(P, q)$ used to guess q^*.
- *qv-4*: c-score standardisation, EM algorithm, $\mathrm{UNC}(P, q)$ used to guess q^*.

3.1. The benchmark algorithms

The *qv* methods are compared to three benchmark algorithms: The method described in Steinley and Brusco (2008a), which is a constructive algorithm using the clusterability index (denoted here *sb-red*), the polynomial reduction described in Andrews and McNicholas (2014) (called here *am-pol*) and the model-based variable selection proposed in Raftery and Dean (2006) (called here *cvs*).

sb-red is a constructive heuristic composed of various steps. First, the pre-processing step screens variables using the clusterability index and retains only a subset of them. Then the remaining variables are standardised using the c-score. To determine the relevant variables, the method uses a combination of complete enumeration and greedy search. It enumerates all variable subsets with cardinality less than or equal to l (a parameter fixed by the user) and calculates optimal clustering with the k-means algorithm. This is the way in which $VAF(P, q)$ is calculated for $q \leq l$. For $q > l$, variables are inserted to subsets one at a time in a greedy fashion. The clusterability index is computed fast, but complete enumeration is practical only when l is a small number. When the method switches to the constructive greedy, it looses precision, so that it is likely that the algorithm works well only when q^* is small. In our simulations, the algorithm is coded as an R script and runs with $l = 3$.

The *am-pol* method is a greedy constructive procedure that selects the relevant features one at time using an index of within-cluster variability. The index is then modified using a non-linear factor to discard correlated features. Optimal clusters and parameter q^* are determined by the EM algorithm and UNC(P, q). The method is fast, but it depends on contingent rules to determine parameters, to select variables and to fit models; the dependency of these rules on the data at hand is an unexplored issue. The algorithm that we used is the one coded in the R package (Andrews & McNicholas 2013).

The *cvs* method is a model-based clustering technique that implements the EM algorithm on data that are assumed to be multivariate normal. Different variable sets are tested by inserting or deleting one variable from an incumbent set. The advantage of the model is that it estimates a full range of parameters: means, variances, covariances and memberships. The drawback is that the EM algorithm must run in every insertion/deletion step, which means that computational times are prohibitive for large data-sets. The algorithm that we used is available in the R package (Scrucca, Adrian, & Raftery, 2013).

3.2. Description of the test problems

The most accurate comparison of variable selection methods to date has been carried out in Steinley and Brusco (2008a). In that paper, data are simulated assuming clusters with multivariate Gaussian distributions and different shapes of masking variables. The experiments controlled for eight factors, including the size and the number of clusters and the probability of clusters overlapping. It was found that clustering results are mainly affected by three factors: The probability distribution of the masking variables, the probability of clusters overlap and the ratio between relevant and masking variables. Therefore, our tests controlled for just these three factors.

The relevant variables are simulated using the procedure described in Qiu and Joe (2006) and available as the R subroutine genRandCluster in the R package (Qiu & Joe 2013). All simulations assume four clusters of equal size, each with 62 units for a total of 248 units (250 units were used in Steinley & Brusco, 2008a). All experiments assume that there is only one way to cluster the data, as they do not consider the problematic case in which data can be clustered in more than one way, depending on which subset of features is selected. To deal with this problem, one has to be careful when selecting the cluster centres R that are used as input of the q-var heuristic. Moreover, one should design consistent methods for determining which alternative clusters are best: one solution is to resort to the objective function value, as it is done in the selection and clustering model proposed in Benati and García (2014). But the presence of alternative clustering structures is a peculiar and difficult problem that is not addressed in the experiments done so far.

Masking variables are simulated using four scenarios:

- The masking variables are all independent normal.
- The masking variables are normal, with means 0 and covariance matrix with diagonal terms equal to 1 and off-diagonal terms equal to 0.5.
- The masking variables are (0, 1)-uniform distributions.
- The masking variables are gamma distributions with location and scale parameters both equal to 1.

All scenarios except the third one were previously considered in Steinley and Brusco (2008a). The first and second scenarios take the clustering problems to the Gaussian setting, so that data fulfil the assumptions for using the EM algorithm. The third scenario, new to this simulation, has been considered to control the effect of symmetric but non-Gaussian random variables. The fourth scenario considers asymmetric masking variables. Regarding the second factor affecting the scenarios, the probability of clusters overlapping has been controlled by the *sep* parameter of the subroutine genRandCluster. The *sep* parameter has been set to 0.20, 0.01, and −0.10 corresponding to increasing probabilities of overlap. The third factor is the ratio between relevant and masking variables. We define n_R as the number of relevant variables and n_M as the number of masking variables. Simulations are run with $n_R = 6, 12$, and $n_M = n_R, 2n_R$.

There is a total of $4 \times 3 \times 4 = 48$ parameters combinations. For each combination, 10 random data-sets are generated. The largest problem is an application to data-sets with 248 rows and 36 columns, which cannot be considered a large data-set in the actual statistics literature. Still, it is the largest problem that the EM algorithm can solve. Finally, all clustering algorithms are run with $k = 4$, that is, the true number of clusters is assumed to be known, as the experiments are mainly focused on variable selection.

3.3. Performance measures

In Steinley and Brusco (2008a), the performance of the selection/clustering algorithms is measured in three different ways:

Cluster Recovery: The true recovery is measured by the ARI, an index which is 1 when there is a perfect recovery of the cluster structure, but which is close to 0 when the cluster recovery is equal to the random choice of cluster assignment (ARI can indeed be negative). The formula of the ARI can be found in Hubert and Arabie (1985), and is available in the R package (Fraley, Raftery, Brendan Murphy, & Scrucca, 2012). ARI is claimed in Steinley and Brusco (2008a) to be the most important index to assess the quality of an algorithm.

Precision: The precision is measured by the number of relevant variables that are contained in the selected subset divided by the cardinality of the selected subset. A precision value equal to 1 means that all the selected variables are relevant, 0 means that the selected variables are all masking.

Recall: The recall is measured by the number of relevant variables of the selected subset divided by the total number of the relevant variables. A value of 1 means that all the relevant variables are selected, 0 means that no relevant variable was selected.

While a high value of both precision and recall is always preferred, it is also true that the two measures must be considered together. For example, suppose that there are 6 relevant variables and 6 masking variables, a method that trivially selects all 12 variables would result in a precision of 1 but a recall of 0.5. It can hardly be considered better than a method with precision and recall both equal to 0.75.

3.4. Computational results

The test results are reported in Table 4. Due to numerical instability, the *cvs* algorithm could not calculate the outcome of 33 problems, so these problems were excluded from the computation of the means for that method. Regarding the clustering ability of the algorithms, the best ARIs are obtained by the two q-vars algorithms using the z-score standardisation. First comes *qv-2*, which uses the EM clustering algorithm, and a close second is *qv-1*, which uses the k-means. Since clusters are simulated as multivariate normal distributions, it may happen that EM clustering is more efficient than k-means and its ARI is better. Regarding the variables selection, it has been found that usually high precision comes with low recall and vice versa. This means that if relevant variables are selected with high probability (the property of high precision), then it is also likely that some of the relevant variables are discarded (the property of low recall). In fact, the method with the best precision is *sb-red*, for which on average 70% of the selected variables are relevant. However, *sb-red* is the method with the worst recall, as on average only 20% of the relevant variables are retrieved. Conversely, the method with the best recall is *am-pol*, however it is also the method with the worst precision. As discussed previously, there is a trade-off between precision and recall and in our tests all the q-vars methods obtain intermediate values of precision and recall. Remarkably, both values are above the threshold of 0.5, which means that at least half of the relevant variables are selected (recall > 0.5) and that at least half of the selected variables are relevant (precision > 0.5).

Tables 5 and 6 report data on ARI, precision and recall for the different structures of the masking variables. Regarding the ARI, when data are normal the *cvs* algorithm is the best, even in the difficult case in which the masking variables come from an elliptic distribution with correlation 0.5. Here all other methods fail to discover the true features, mistaking the elliptic masking multivariate distribution for regular clusters. But *cvs* does not work well when masking features are not normal. For example, it fails if the masking variables follow a gamma distribution. The explanation of this behaviour is that the *cvs* method assumes multivariate Gaussian data and it is indeed very good when data are normal. But the algorithm performs much worse when data are not normal, so it is highly dependent on the application. The methods based on the k-means are less sensitive to the normality assumption, as can be seen in the data for the gamma distribution. Moreover, the *cvs* method selects all relevant variables, but many true variables are discarded, as shown from the precision of 1 and the recall of 0.48. The precision and the recall of *qv-1* are good too, as both are greater than 0.75, thus showing that only one out of four relevant variables is left out of the selected set and only one out of four selected variables is masking.

Tables 7 and 8 report data conditional to the degree of separation between clusters. When clusters are well separated (see the results for high separation), clustering is easier and all algorithms achieve satisfactory ARI results. But when the separation is medium or low, the *qv-1* algorithm provides the best value of ARI. The reason is that Gaussian clusters are easy to recognise when well separated but hard to recognise when the clusters overlap. That is, the maximum likelihood estimation becomes more difficult. Conversely, methods based on k-means are more effective. Regarding precision and recall, the results of the *qv-1* method are always greater than 0.5, as opposed to the results of the *cvs*, which provides poor numbers when clusters are not well separated.

Tables 9 and 10 report data conditional to the ratio between masking and true variables. When the ratio is 1, the best method is *qv-2*, but when the ratio is 3 the best method is *qv-1*. The explanation is that too many

Table 4. Global results of the selection/clustering algorithms.

	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
ARI	0.541	0.568	0.349	0.517	0.386	0.480	0.463
precision	0.607	0.523	0.557	0.498	0.705	0.551	0.573
recall	0.596	0.737	0.627	0.774	0.228	0.868	0.280

Table 5. Average ARI on different contaminating distributions.

	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
N(0, 1)	0.730	0.774	0.586	0.651	0.481	0.630	0.783
Ncor0.5	0.055	0.330	0.190	0.321	0.454	0.328	0.783
Unif01	0.689	0.705	0.175	0.768	0.166	0.585	0.346
Gamma11	0.688	0.462	0.447	0.328	0.442	0.374	0.008

Table 6. Average precision and recall on different contaminating distributions.

	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
N(0, 1)-Prec	0.799	0.619	0.663	0.515	0.832	0.628	1.000
Recall	0.752	0.859	0.685	0.831	0.233	0.921	0.477
Ncor0.5-Prec	0.072	0.304	0.302	0.367	0.797	0.419	1.000
Recall	0.092	0.490	0.347	0.608	0.218	0.799	0.477
Unif01-Prec	0.791	0.621	0.814	0.710	0.419	0.649	0.371
Recall	0.760	0.803	0.906	0.960	0.240	0.889	0.203
Gamma11-Prec	0.767	0.547	0.449	0.400	0.771	0.508	0.008
Recall	0.782	0.795	0.572	0.698	0.220	0.865	0.006

Table 7. Average ARI on degrees of separation between clusters.

	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
High	0.735	0.835	0.499	0.731	0.702	0.705	0.719
Medium	0.535	0.521	0.281	0.483	0.253	0.420	0.397
Low	0.352	0.348	0.269	0.338	0.202	0.314	0.233

Table 8. Average precision and recall on degrees of separation between clusters.

	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
HighSep-Prec	0.645	0.584	0.579	0.516	0.903	0.642	0.742
Recall	0.615	0.812	0.645	0.814	0.322	0.888	0.413
MediumSep-Prec	0.619	0.508	0.541	0.480	0.665	0.525	0.551
Recall	0.573	0.727	0.598	0.780	0.195	0.880	0.246
LowSep-Prec	0.559	0.476	0.552	0.498	0.546	0.486	0.394
Recall	0.601	0.672	0.639	0.730	0.166	0.838	0.162

Table 9. Average ARI on ratio of relevant, masking variables.

	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
#mask = #true	0.571	0.696	0.443	0.669	0.427	0.663	0.486
#mask = 2(#true)	0.510	0.439	0.256	0.365	0.345	0.296	0.438

Table 10. Average precision and recall on ratio of relevant,masking variables.

	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
#mask = #true-prec	0.736	0.669	0.665	0.615	0.783	0.664	0.580
recall	0.531	0.859	0.681	0.928	0.238	0.842	0.286
#mask = 2(#true)-prec	0.479	0.377	0.449	0.381	0.627	0.439	0.565
recall	0.662	0.615	0.574	0.620	0.217	0.895	0.274

noising data hinder the EM algorithm, something that does not affect *k*-means based methods.

As a conclusion, the tests show that no method is the best overall scenarios, but that successful application depends on the data at hand. However, it is important to note that when data are more difficult to analyse, that is, when they are not normal, overlap, and contain many masking variables, then the *qv-1* method performs better. This conclusion is strengthened when looking at the computational times reported in

Table 11. Computation time in seconds (averages) of the seven algorithms.

#relevant	#masking	qv1	qv2	qv3	qv4	sb-red	am-pol	cvs
6	6	0.63	3.98	0.62	4.25	0.04	0.73	7.93
6	12	1.18	9.03	1.10	7.58	0.19	1.08	10.87
12	12	1.96	14.72	1.81	13.44	0.18	1.68	15.61
12	24	3.73	28.46	3.62	29.98	0.19	2.98	23.69

Table 11. The *qv-1* algorithm is one of the fastest methods. It is able to handle the large size data that the EM algorithm is unable to cope with. For these reasons, the combination q-vars/k-means is the tool that we suggest for data analysis. Some algorithms tested in this paper is available in the R/CRAN package “qVarSel”.

4. Conclusions

The problem of selecting relevant variables for clustering has been formulated as a combinatorial optimisation model in this paper. The model is solved with integer linear programming or heuristic methods to determine the best variable selection subroutine for a clustering application. Extensive tests on simulated data provided evidence that the approach can determine the relevant features and improve the clustering quality. Future research can be devoted to improve some computational issues of the problem. For example, the radius formulation is a methodology that has a strong connection with the pseudo-boolean representation of the objective function (see AlBdaiwi et al., 2011; Church, 2003; Church, 2008). In this way, one can refine the MILP formulation using even less coefficients and constraints, as proved and experimented in a similar problem in Goldengorin and Krushinsky (2011). Moreover, the problem of selecting relevant features is not only important in clustering, but also in other statistical techniques such as classification or supervised learning (see Guyon & Elisseef, 2003; Yang & Olafsson, 2009), support vector machines (see Maldonado, Pérez, Weber, & Labbé 2014), and linear regression (see Hoking, 1976). It is likely that the methods developed here can be modified to fit these relevant applications. Another application would be to consider the geographical interpretation of the model and insert distance selection into the p-median problem (see Mladenovic et al., 2007).

ORCID

Stefano Benati http://orcid.org/0000-0002-1928-5224
Serigo García http://orcid.org/0000-0003-4281-6916
Justo Puerto http://orcid.org/0000-0003-4079-8419

Funding

Justo Puerto has been funded by the Spanish Ministry of Economy [grant number MTM2013-46962-C02-01], [grant number MTM2016-74983-C02-01]; Sergio Garcia by the Fundacion Seneca Stefano Benati by Ministrero dell’Istuzione; Fundación Séneca [grant number 19320/PI/14]; Ministero dell’Istruzione, dell’Universitá e della Ricerca [grant number MTM2013-46962-C02-01].

References

AlBdaiwi, B., Ghosh, D., & Goldengorin, B. (2011). Data aggregation for p-median problems. *Journal of Combinatorial Optimization, 21*, 348–363.

Andrews, J. L., & McNicholas, P. D. (2013). *vscc: Variable selection for clustering and classification. R package version 1*. Retrieved from http://CRAN.R-project.org/package=vscc

Andrews, J. L., & McNicholas, P. M. (2014). Variable selection for clustering and classification. *Journal of Classification, 31*, 136–153.

Avella, P., Boccia, M., Salerno, S., & Vasilyev, I. (2012). An aggregation heuristic for large scale p-median problems. *Computers and Operations Research, 39*, 1625–1632.

Avella, P., Sassano, A., & Vasil’ev, I. (2007). Computational study of large-scale p-median problems. *Mathematical Programming, 109*, 89–114.

Benati, S., & García, S. (2014). A mixed integer linear model for clustering with variable selection. *Computers and Operations Research, 43*, 280–285.

Brusco, B. J. (2004). Clustering binary data in the presence of masking variables. *Psychological Methods, 9*, 510–523.

Caballero, R., Laguna, M., Martí, R., & Molina, J. (2011). Scatter tabu search for multiobjective clustering problems. *The Journal of the Operational Research Society, 62*, 2034–2046.

Carmone, F. J., Kara, A., & Maxwell, S. (1999). HINoV: A new model to improve market segmentation by identifying noisy variables. *Journal of Marketing Research, 36*, 501–509.

Chen, J. S., Ching, R. K. H., & Lin, Y. S. (2004). An extended study of the k-means algorithm for data clustering and its applications. *The Journal of the Operational Research Society, 55*, 976–987.

Church, R. L. (2003). COBRA: A new formulation of the classic p-median location problem. *Annals of Operations Research, 122*, 103–120.

Church, R. L. (2008). BEAMR: An exact and approximate model for the p-median problem. *Computers and Operations Research, 35*, 417–426.

Cornuejols, G., Nemhauser, G., & Wolsey, L. (1980). A canonical representation of simple plant location-problems and its applications. *SIAM Journal on Algebraic And Discrete Methods, 1*, 261–272.

Elloumi, S. (2010). A tighter formulation of the p-median problem. *Journal of Combinatorial Optimization, 19*, 69–83.

Elloumi, S., Labbé, M., & Pochet, Y. (2004). A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing, 16*, 84–94.

Fraiman, R., Justel, A., & Svarc, M. (2008). Selection of variables for cluster analysis and classification rules. *Journal of the American Statistical Association, 103*, 1294–1303.

Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association, 97*, 611–631.

Fraley, C., Raftery, A .E., Brendan Murphy, T., & Scrucca, L. (2012). *mclust version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation* (Technical Report No. 597). Department of Statistics, University of Washington.

Fowlkes, E. B., Gnanadesikan, R., & Kettering, J. R. (1988). Variable selection in clustering. *Journal of Classification, 5*, 205–228.

Friedman, J., & Meulman, J. (2004). Clustering objects on subsets of attributes. Journal of the Royal Statistical Society. *Ser. B, 66*, 815–849.

García, S., Labbé, M., & Marín, A. (2011). Solving large p-median problems with a radius formulation. *INFORMS Journal on Computing, 23*, 46–556.

García, S., Landete, M., & Marín, A. (2012). New formulation and a branch-and-cut algorithm for the multiple allocation p-hub median problem. *European Journal Of Operational Research, 220*, 48–57.

García-Escudero, L. A., Gordaliza, A., & Matrán, C. (2003). Trimming tools in exploratory data analysis. *Journal of Computational and Graphical Statistics, 12*, 434–449.

Goldengorin, B., & Krushinsky, D. (2011). Complexity evaluation of benchmark instances for the p-median problem. *Mathematical and Computer Modeling, 53*, 1719–1736.

Guyon, I., & Elisseef, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

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

Hoking, R. R. (1976). The analysis and selection of variables in linear regression. *Biometrics, 32*, 1–49.

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

Kariv, O., & Hakimi, S. L. (1979). An algorithmic approach to network location problems, part II. The p-medians. *SIAM Journal on Applied Mathematics, 37*, 539–560.

Law, M. H. C., Figuereido, M. A. T., & Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*, 1154–1166.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). Berkeley, CA: University of California Press.

Marín, A., Nickel, S., Puerto, J., & Velten, S. (2009). A flexible model and efficient solution strategies for discrete location problems. *Discrete Applied Mathematics, 157*, 1128–1145.

McLachlan, G. J., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York, NY: Wiley.

Maldonado, S., Pérez, J., Weber, R., & Labbé, M. (2014). Feature selection for support vector machines via mixed integer linear programming. *Information Science, 279*, 163–175.

Mladenovic, N., Brimberg, J., Hansen, P., & Moreno-Pérez, J. A. (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research, 179*, 927–939.

Morlini, I., & Zani, S. (2013). *Variable selection in cluster analysis: An approach based on a new index*. (in Giusti A., Ritter G. and Vichi M. - Classification and Data Mining - Springer, Berlin DEU, Studies in Classification, Data Analysis, and Knowledge Organization: 71–79).

Qiu, W., & Joe, H. (2006). Generation of random cluster with specified degree of separation. *Journal of Classification, 23*, 315–334.

Qiu, W., & Joe, H. (2013). *clusterGeneration: Random Cluster Generation (with specified degree of separation)*. Retrieved from http://CRAN.R-project.org/package=clusterGeneration

Pan, W., & Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research, 8*, 1154–1164.

Puerto, J., Ramos, A. B., & Rodríguez-Chía, A. M. (2013). A specialized branch & bound & cut for single-allocation ordered median hub location problems. *Discrete Applied Mathematics, 161*, 2624–2646.

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

Scrucca, L., Adrian, E., & Raftery, N. D. (2013). *clustvarsel: A package implementing variable selection for model-based clustering in R, version 2.0*. Retrieved from http://CRAN.R-project.org/package=clustvarsel

Steinley, D., & Brusco, M. J. (2008a). A new variable weighting and selection procedure for k-means cluster analysis. *Multivariate Behavioral Research, 43*, 77–108.

Steinley, D., & Brusco, M. J. (2008b). Selection of variables in cluster analysis: an empirical comparison of eight procedures. *Psychometrika, 73*, 125–144.

Tadesse, M. G., Sha, N., & Vannucci, M. (2005). Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association, 100*, 602–617.

Yang, J., & Olafsson, S. (2009). Near-optimal feature selection for large databases. *The Journal of the Operational Research Society, 60*, 1045–1055.

Witten, D. M., & Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association, 105*, 713–726.