Modelos de optimización entera mixta en el análisis clúster con selección de variables

Irene Urdin Bravo

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Directoras del trabajo:
Herminia I. Calvete Fernández y
Carmen Galé Pola

28 de junio de 2021

Prólogo

El análisis clúster, o análisis de conglomerados, es una técnica estadística multivariante que engloba una serie de métodos que tienen por objetivo establecer grupos de individuos dentro de un conjunto de datos, de tal forma que cada grupo sea lo más homogéneo posible y exista máxima heterogeneidad entre grupos. Estos métodos empezaron a desarrollarse en el campo de la biología, para solventar problemas de clasificación de especies, y a día de hoy están ampliamente extendidos en campos tan diversos como la lingüística, la ingeniería o la sociología.

Para determinar qué individuos componen cada grupo, se atiende a la información disponible sobre ellos que ha sido recogida en una colección de variables. Si el número de variables es muy alto, es útil aplicar procedimientos de reducción de la dimensión para reducir el coste computacional. También es posible que se haya recogido información sobre más variables de las necesarias para identificar los grupos. Estas variables pueden llegar a entorpecer el procedimiento, enmascarando las estructuras de agrupación inherentes en los datos. Este trabajo se centra en aquellos modelos de optimización que permitan seleccionar cuáles de entre todas las variables contribuyen a caracterizar los grupos existentes en la muestra. Los modelos estudiados están diseñados para colecciones de datos binarios, que han sido menos estudiados en la literatura, pero que aparecen con mucha frecuencia en muchos ámbitos de estudio, dada su versatilidad. Por ejemplo, estas variables aparecen en el ámbito biosanitario para describir la presencia o no de síntomas en pacientes.

El capítulo 1 de la memoria incluye una visión general de dónde se sitúan las técnicas de análisis clúster dentro de los procesos de aprendizaje estadístico y se concluye con un apartado en el que se especifica la contribución de este TFG en dicho contexto.

El capítulo 2 comienza describiendo la importancia de seleccionar variables o reducir la dimensión como paso previo a la definición de los grupos. A continuación, se introduce el primer modelo de programación entera que determina simultáneamente la selección de variables y los grupos de individuos atendiendo a dichas variables. Como es un modelo no lineal se han revisado las dos linealizaciones alternativas, una de ellas clásica y otra que reduce la simetría de la formulación original y es computacionalmente más eficiente.

En el capítulo 3 se describe el procedimiento estadístico elegido, tras revisar la literatura, que permite simultáneamente reducir la dimensión de un conjunto de variables categóricas y determinar los grupos. El enfoque no coincide exactamente con el del modelo de programación matemática y esto se ha tenido en cuenta en el diseño de la experiencia computacional.

Finalmente, en el capítulo 4 se presentan dos colecciones de datos generadas adhoc para cada uno de los procedimientos y se comparan los resultados propuestos por el modelo de programación entera y el procedimiento estadístico.

Summary

Cluster analysis comprises a series of statistical and optimization methods that aim to determine whether a data set can be divided into a number of groups or clusters, and if so, how this partition can be performed so that each cluster that is obtained is as homogeneous as possible. This technique allows to identify 'natural' structures in a data set, so that it can be later studied more thoroughly according to these. Cluster analysis is a so called *unsupervised learning* technique, since it is the method itself that discovers the patterns and determines the groups on its own. It was first developed in the field of biology, to solve problems of categorization of species, and nowadays it is broadly used in many other fields such as sociology and engineering.

When performing cluster analysis, there exists a number of prior decisions that must be made. These decisions concern the number of groups in the data set, the choice of a distance measure and the selection of the variables that are more relevant to identify the cluster structures as well as deciding on the specific method to be used. This work addresses, up to a certain point, all of them but focuses mostly on the last two.

There exists a vast number of clustering techniques, that can be divided into two main categories: hierarchical methods and non-hierarchical or partitioning methods. This work focuses on non-hierarchical methods, which seek to find a grouping of objects such that maximises or minimises a certain evaluating criteria, since these are better suited to be applied to large data sets. To determine whether an individual belongs to a certain group, we focus on the information available registered in the variables. When the set of variables is large, it is useful to apply some dimension reduction technique to the data, so that it can be better managed. Sometimes, it is not only useful but necessary too. It can happen that, out of all the variables that are collected, only a small amount of them are relevant to identify the cluster structures within the data set. We call these variables the 'real' ones, in opposition to the remaining ones, that are called 'masking' variables. The presence of a large amount of masking variables in a data set may hinder the ability of the partitioning method to retrieve the true cluster structure.

In the clustering literature, there is a very limited number of models that have been developed to work with binary variables. Most of the developed methods focus on numerical and categorical variables. However, binary variables are becoming more and more common due to their versatility. They arise, for example, in medical and psychological tests, to determine the presence or absence of symptoms in patients.

In chapter 1, we include a general overview of procedures of statistical learning and establish where cluster analysis stands among them. We include as well a paragraph where it is specified the contribution of this work in that context.

In chapter 2, we have studied a mixed integer programming model that has been developed for binary data sets, and that performs the clustering of the data at the same time that reduces the dimension, by selecting a given number of variables out of the whole set. The model, which we have called ACSV, seeks to select the variables that are most significant to define the clusters and also the observations that are the most representative of each group, according to the values that they take on the selected

variables. It is formulated initially as a non-linear optimization problem for which two different linearizations have been proposed in the literature. We have studied and compared both of them using the optimization software CPLEX.

In chapter 3, we describe a statistical procedure that we have chosen after revising the literature, in order to compare its results with those provided by the ACSV model. The method is called MCA *k*-means and it also performs a partitioning clustering method while reducing the dimension of the set of variables. However, the approach followed is different from the ACSV model. On the one hand, MCA stands for multiple correspondence analysis, which is a dimension reduction technique. This technique does not select a number of variables, but instead computes a smaller new set of variables, that are a combination of them all. Each new variable is built so that, when proyecting the data in the reduced dimension, the least possible loss of information is achieved. On the other hand, *k*-means is a famous partitioning method, that allocates observations into a given number of clusters so that the distance between each observation and the centroid of the cluster to which it belongs to is minimum. The objective function of MCA *k*-means is formulated as a convex combination of both the objective functions of *k*-means and MCA. This method is already implemented in the statistical software R, in the *clustrd* library, using the *clusmca()* function.

Since ACSV and MCA *k*-means follow different approaches, we have designed two different data sets, each one better suited for each method, and we have compared their performance. The first set of instances are binary data sets where there exists a big number of masking variables. The second sets of instances are sets of categorical variables available in the R library where MCA *k*-means is implemented.

In the first part of the computational experience we have studied more in depth the multiobjective approach of the method MCA *k*-means, by comparing the results obtained when moving the value of α in the convex formulation of the objective function from 0 to 1. We have seen that, since it is an heuristic method, the optimal solutions provided are not real optimals but approximations to them.

Secondly, we have studied the two linearized formulations of the ACSV model, in order to compare their efficiency in terms of the computational time required. It has been shown that the adhoc linearization requires a much smaller amount of time to compute the optimal solution.

Lastly, we have applied both models to the data sets and compared the results obtained. We have concluded that, in presence of a large number of masking variables, MCA *k*-means is not fully able to correctly retrieve the existing clustering structures. Nonetheless, this method is useful when there is not a large number of masking variables. In fact, in a set where most of the variables provide useful information to identify the clusters, the ACSV model, which selects only a few, might not be the most suitable option, unless further restrictions are set to the model.

Índice general

Capítulo 1

Introducción al aprendizaje estadístico

1.1. Introducción

Diariamente, se generan y almacenan grandes volumenes de datos para ser utilizados con distintos propósitos. Esos datos necesitan ser tratados adecuadamente para poder extraer conocimiento y contribuir a una buena toma de decisiones. En este capítulo se van a exponer técnicas matemáticas, basadas en conceptos estadísticos y de álgebra matricial, que se utilizan a día de hoy para el tratamiento multivariante de los datos [5] .

Para poder describir situaciones reales se define la población bajo estudio y se recoge información bien de una muestra o, si es posible, de toda la población con los procedimientos automáticos que los avances tecnológicos facilitan. En general, la información se registra en variables y se construyen ficheros de datos con los valores de las p variables observadas. El tipo de análisis que se aplica para extraer información depende de la naturaleza de las variables. Se distingue entre variables *cuantitativas*, cuyo valor se expresa numéricamente, y variables *cualitativas* cuando la naturaleza de la variable no es numérica. Estas últimas, para facilitar su gestión, en ocasiones son codificadas como numéricas.

El análisis estadístico multivariante abarca el conjunto de métodos estadísticos que contribuyen a analizar simultáneamente varias variables registradas para cada elemento estudiado de una población. En primer lugar incluye los *métodos de exploración de datos*, que permiten realizar un análisis descriptivo de los datos. El objetivo principal de estos métodos es facilitar la comprensión de la estructura de la muestra para poder determinar la técnica estadística más avanzada a utilizar en su análisis. Son métodos que presentan los datos de forma sintetizada con la menor pérdida de información posible. El resumen de información puede ser gráfico, con el uso de histogramas, diagramas de caja o gráficos de dispersión, o numérico, como son las medidas de centralización y variabilidad. Estos métodos permiten también detectar errores o datos atípicos, que aparecen con frecuencia en los conjuntos de datos multivariantes. La presencia de datos atípicos puede modificar los modelos, creando relaciones inexistentes entre las variables o llegando a ocultar las que sí existen.

Los métodos de análisis multivariante de datos se pueden organizar en tres grandes clases. Las *técnicas de reducción de la dimensión* tienen como principal objetivo representar los datos originales en un espacio de dimensión menor de manera que se conserve en la medida de lo posible la estructura inicial de los datos. Las *técnicas de aprendizaje supervisado* son aquellas para las que el proceso de aprendizaje a partir de los datos está guiado por una variable respuesta y pretenden definir criterios o modelos para predecir el comportamiento de un individuo a partir de la información disponible. Finalmente, las *técnicas de aprendizaje no supervisado* estudian las relaciones y estructuras inherentes a los datos y permiten extraer conocimiento de estos conjuntos de datos sin disponer de la guía de una variable respuesta para realizar el estudio.

1.2. Técnicas de reducción de la dimensión

La reducción de la dimensionalidad es muy importante al trabajar con datos multivariantes porque facilita la comprensión de la estructura subyacente de los datos. Atendiendo a los objetivos del análisis se distinguen tres grupos de técnicas. [5]

1.2.1. Análisis de componentes principales

Es la técnica más utilizada con variables numéricas continuas. La idea es determinar un conjunto de nuevas variables, definidas como combinación lineal de las variables originales, tales que al proyectar los datos sobre el subespacio generado por las nuevas variables se conserve la estructura del espacio original. El procedimiento de cálculo de los coeficientes de la combinación lineal está basado en los valores y vectores propios de la matriz de correlación. La primera componente principal tiene en cuenta el valor propio dominante y la segunda componente, el segundo valor propio. Ambas componentes definen el plano que mejor representa a los datos. En este procedimiento es importante la interpretación de las componentes principales.

Un método análogo cuando en vez de disponer de una matriz de datos por variables se dispone de una matriz cuadrada de distancias o disimilaridades es el escalado multidimensional.

1.2.2. Análisis factorial

El objetivo de esta técnica es reemplazar un conjunto amplio de variables por unas pocas nuevas no observables, denominadas $factores$ que nos permitan prever las originales. La idea es que si las variables son muy dependientes entre sí, la densidad de probabilidad se va a concentrar en unas zonas determinadas del espacio, definidas por la relación de dependencia entre las variables originales. En este caso, las variables originales se definen como combinación lineal de los factores. El análisis factorial permite comprender mejor el mecanismo generador de los datos.

1.2.3. Análisis de correspondencias

Esta técnica de análisis multivariante se aplica con variables cualitativas. En este caso, la matriz considera las frecuencias y se obtiene a partir de la tabla de contingencia. Al igual que en el análisis de componentes principales, la idea es encontrar nuevas variables que permitan sintetizar la información de los datos en un subespacio de dimensión menor con la menor pérdida de información posible. Este procedimiento permite también obtener variables numéricas a partir de la información dada por las categóricas.

1.3. Técnicas de aprendizaje supervisado

Para poder extrapolar a toda la población las conclusiones obtenidas a partir de una muestra, es necesario construir un modelo estadístico que caracterice cómo se generan los datos, es decir, qué distribución de probabilidad sigue la variable aleatoria vectorial en la población. Algunas de las distribuciones de variables vectoriales más comunes son la distribución *normal p-dimensional*, la *multinomial* y la distribución de *Dirichlet*. La primera es especialmente importante ya que los métodos de inferencia más utilizados asumen la normalidad conjunta de las variables.

1.3.1. Métodos de análisis discriminante

Estos métodos se aplican cuando se conoce que los datos provienen de dos o más poblaciones conocidas, con el objetivo de clasificar nuevas observaciones en ellas.

El análisis discriminante clásico supone que los datos se han generado siguiendo una distribución normal multivariante conocida en cada una de las poblaciones. El análisis busca las relaciones lineales

entre las variables disponibles que mejor discriminen en los grupos dados a los individuos. En este supuesto, la regla de clasificación a seguir consiste en asignar la observación a la población de cuya media esté más próxima. Cuando la distribución conjunta de las observaciones no es normal multivariante o no conocemos los parámetros de la distribución y debemos estimarlos a través de los datos, se utilizan otros métodos de discriminación, como los modelos de respuesta cualitativa o los *árboles de clasificación* y las *redes neuronales*, que requieren un uso intensivo del ordenador.

1.3.2. Análisis de correlación canónica

El análisis de correlación canónica es el método más utilizado para estudiar globalmente la relación entre dos conjuntos de variables, **X** e **Y**. La idea es encontrar un par de variables que se denominan *variables canónicas*, tal que una sea combinación lineal de las del primer grupo, **X**, y otra resuma las variables de **Y**, con la condición de que estas nuevas variables tengan correlación máxima.

En ocasiones, se da preferencia a uno de los dos conjuntos de variables para explicar el otro, por ejemplo a **X** sobre **Y**. El *análisis canónico asimétrico* estudia cuántas combinaciones lineales de las variables del primer grupo explican las del segundo. En este contexto se puede construir un *modelo de regresión multivariante*, que incluye el estudio conjunto de las regresiones entre cada variable de **Y** con todas las variables de **X**. En todoso los casos, el objetivo es buscar las relaciones que hay entre dos grupos de variables, de forma que se pueda predecir múltiples variables dependientes a partir de múltiples independientes.

1.4. Técnicas de análisis no supervisado

Estos métodos analizan directamente el conjunto de datos de entrada sin disponer de una 'etiqueta' previa de los datos. Entre los procedimientos más habituales están los métodos de análisis clúster o de conglomerados y las técnicas de clasificación mediante mezclas.

1.4.1. Análisis clúster

El objetivo consiste en, sin disponer de conocimiento previo, poder determinar si los datos recogidos corresponden a una muestra homogénea o no, y en este último caso, determinar el número de grupos distintos existentes y clasificar a los individuos en grupos homogéneos. Aunque normalmente se agrupan las observaciones, también se puede aplicar para agrupar variables, o incluso existen métodos que permiten agrupar simultáneamente variables e individuos.

Los métodos de análisis clúster requieren tomar una serie de decisiones previas sobre el número de grupos, la medida de proximidad entre individuos y las variables relevantes para identificar los grupos, así como sobre el método de agrupamiento a utilizar. Se distingue entre *métodos de partición* y *métodos jerárquicos*. El objetivo de los métodos de partición es que todas las observaciones queden clasificadas, cada una en un único grupo, atendiendo a algún criterio de optimización para realizar la asignación a los grupos. Por otra parte, los métodos jerárquicos estructuran los datos por niveles de manera que los niveles superiores contienen a los inferiores. Estrictamente no definen grupos, sino la estructura de asociación en cadena que pueda existir entre los elementos, pero la jerarquía construida también permite obtener una partición de los datos en grupos.

1.4.2. Métodos de clasificación mediante mezclas

Estos métodos son una extensión del problema de análisis clúster, y se aplican para determinar si los datos proceden o no de una misma población. Destacan los mecanismos de *estimación de mezclas normales* y el *método de proyección*. El primero supone que los datos se han generado como una mezcla de distribuciones normales multivariantes desconocidas, por lo que se precisa estimar los parámetros de los componentes de la mezcla para luego clasificar las observaciones en la población a la que tengan mayor probabilidad de pertenencia. El segundo método intenta encontrar una dirección de proyección

de los datos que revele, si existen, los distintos grupos de manera que haya la mayor distancia posible entre ellos.

1.5. Contribución del trabajo

En este capítulo se han introducido de forma breve las diferentes aproximaciones para el tratamiento de datos multivariantes que permiten extraer conocimiento a partir de las grandes cantidades de datos disponibles en la actualidad. Al ser necesario manejar gran cantidad de datos, los recursos de la estadística se combinan con los de otras áreas, como la computación y la investigación operativa, en particular, con los modelos de programación matemática. En este trabajo se va a explorar esta conexión. Entre las técnicas de aprendizaje no supervisado, el análisis clúster no requiere la construcción de un modelo probabilístico subyacente a las observaciones. Por ello, resulta de interés analizar el problema desde la perspectiva de la programación matemática.

Por otro lado, la mayor parte de los procedimiento estadísticos se han centrado en variables numéricas y categóricas, siendo pocos los métodos centrados en el tratamiento de variables binarias. En la actualidad, este tipo de datos es frecuente por su versatilidad en la codificación de información en muchos ámbitos. Por ejemplo, las variables binarias aparecen en test médicos y psicológicos para determinar la presencia o no de ciertos síntomas en un grupo de pacientes, o de color en procesos de procesamiento de imágenes.

El modelo de programación matemática entera en el que se centra este trabajo combina un método de selección de variables con la clasificación del conjunto de individuos en grupos. Así, de forma simultánea, elige el subconjunto de variables del total de variables definidas sobre los individuos en cuyo valor se va a fijar para determinar el grupo de pertenencia del individuo. En general, no todas las variables son igualmente importantes en la definición de la estructura de grupos de observaciones porque su inclusión deteriora la efectividad en los procedimientos de agrupamiento. Las variables que no definen la estructura de clúster se denomina *masking* variables o variables ruido frente a las variables verdaderas. El problema de programación entera se describe en el capítulo 2.

Con el propósito de comparar los resultados dados por el modelo de programación entera, se ha buscado un procedimiento estadístico conocido en la literatura que, en cierto modo, proporcionara de forma integrada la reducción de la dimensión y el análisis clúster y permitiera el tratamiento de datos binarios. Este procedimiento estadístico se describe en el capítulo 3. El enfoque es distinto al del modelo de programación matemática tanto en el proceso de reducción de la dimensión como en la aplicación de los métodos de agrupamiento. En cuanto a la reducción de la dimensión, el método no selecciona variables, sino que determina un nuevo conjunto reducido de variables que se definen como combinación de las variables originales. Por tanto, aunque se reduzca la dimensión, se tienen en cuenta todas las variables en el proceso. En lo que concierne al método de agrupamiento, la diferencia más notable es que el algoritmo utilizado define los centroides de los grupos, que son los elementos representantes de los mismos, como la media de las observaciones asignadas al grupo. En el caso del método de programación entera, los centroides son individuos del conjunto de datos.

Finalmente, en el capítulo 4 se incluyen los resultados de las experiencias computacionales desarrolladas. Para este estudio ha sido necesario generar dos colecciones de datos. Dado que los enfoques de los procedimientos que se comparan son diferentes, cada colección es apropiada para uno de ellos, y se evalúa la efectividad del otro. Se han comparado también las dos formulaciones matemáticas del problema de optimización propuesto y se ha incluido una aproximación multiobjetivo al procedimiento estadístico de la literatura.

Capítulo 2

Modelos de programación entera en análisis clúster con selección de variables

2.1. Importancia de reducir la dimensión y seleccionar las variables

El estudio de grandes conjuntos de datos conlleva una dificultad añadida evidente. En particular puede resultar problemática la ejecución de los algoritmos de agrupamiento habituales en estadística para determinar los grupos existentes en la muestra, y no sólo por el coste computacional elevado. Según aumenta el número de variables definidas sobre cada individuo, calcular la proximidad entre elementos y la separación entre grupos es cada vez más complicado. Por ello, en muchas ocasiones interesa reducir el número de variables que intervienen. En otras ocasiones, no se trata solo de trabajar en un espacio de menor dimensión sino que, de todas las variables, existen muchas que no aportan información para definir los grupos e incluso pueden enmascarar las estructuras reales de los mismos. Dicho de otro modo, pueden estar ocultando las variables verdaderas que definen los grupos y, de esta manera, obstaculizar que el algoritmo identifique correctamente los grupos.

Los métodos de selección de variables resuelven este doble objetivo de reducir la dimensión y elegir un subconjunto de las variables originales que resultan más relevantes en la definición de los grupos, descartando el resto. De esta manera, se diferencian de las técnicas de reducción de la dimensión, como el análisis de componentes principales y el análisis de correspondencias, descritos en el capítulo 1, dado que la reducción de la dimensión se lleva a cabo sin necesidad de transformar las variables.

2.2. Modelo de optimización entera mixta en el análisis clúster con selección de variables (ACSV)

En el trabajo de Benati y García [11] se formula un problema de programación entera que determina el subconjunto de variables de un tamaño fijado y la clasificación de todas las observaciones en un número dado de grupos, de forma que los grupos definidos son homogéneos en relación con las variables seleccionadas.

El modelo se basa en el problema de la p-mediana y está diseñado para trabajar con conjuntos de datos binarios, o conjuntos de datos cuyas variables se expresan en una escala de Likert [1]. En este contexto, la mediana se corresponde con el centroide de un clúster.

Escoger una medida de distancia entre los datos que sea apropiada es un paso fundamental. Para variables binarias, una distancia muy utilizada es la *distancia de hamming*, que cuenta el número de

[1] La Escala de Likert es una escala de calificación que se utiliza en las encuestas para preguntar a una persona sobre su nivel de acuerdo o desacuerdo con una declaración

variables en las que los individuos i, j toman valores distintos.

En este modelo se define la distancia entre dos observaciones, independientemente de si las variables son binarias o están expresadas en la escala de Likert, como

$$d_{ij} = \sum_{k=1}^{p} d_{ijk} = \sum_{k=1}^{p} |v_{ik} - v_{jk}|$$

donde v_{ik} es el valor de la observación i en la variable k. Para el caso de variables binarias, esta función de distancia coincide con la distancia de hamming.

Se introduce la siguiente notación:

- n es el número de elementos en la muestra,
- p es el número de variables,
- d es la dimensión final reducida que queramos obtener para los datos,
- K es el número de grupos,
- y_j toma valor 1 si la observación j es centroide de un cluster, y 0 en caso contrario,
- x_{ij} toma valor 1 si la observación i está en el grupo caracterizado por j, y 0 en caso contrario,
- z_k toma valor 1 si la variable k es seleccionada y 0 en caso contrario.

El modelo se formula como sigue:

ACSV

$$\min \quad \sum_{j=1}^{n} \sum_{i=1}^{n} \left(\sum_{k=1}^{p} d_{ijk} z_k \right) x_{ij} \tag{2.1a}$$

sujeto a

$$x_{ij} \leqslant y_j, \qquad i, j = 1, \ldots, n \tag{2.1b}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, \ldots, n \tag{2.1c}$$

$$\sum_{j=1}^{n} y_j = K \tag{2.1d}$$

$$\sum_{k=1}^{p} z_k = d \tag{2.1e}$$

$$x_{ij} \geqslant 0, \qquad i, j = 1, \ldots, n \tag{2.1f}$$

$$y_j \in \{0, 1\}, \qquad j = 1, \ldots, n \tag{2.1g}$$

$$z_k \in \{0, 1\}, \qquad k = 1, \ldots, p \tag{2.1h}$$

- La función objetivo (2.1a) minimiza la suma en todos los grupos de las distancias entre las observaciones y el centroide del grupo al que son asignadas, atendiendo sólo a un subconjunto de variables para determinar dicha distancia.
- Las restricciones (2.1b) garantizan que si una observación no es mediana de un grupo entonces no se puede asignar ninguna otra observación.
- Las restricciones (2.1c) garantizan que una observación se asigna a un único grupo.

- Las restricción (2.1d) garantizan que el número de centroides sea igual al número de grupos.
- La restricción (2.1e) garantizan que se seleccionan d variables para identificar los grupos.
- Las restricciones (2.1f), (2.1g) y (2.1h) determinan el rango de las variables de decisión.

El problema de la p-mediana corresponde al problema en el que no se reduce la dimensión y únicamente se determinan los K centroides y la asignación de observaciones a cada centroide que minimice la suma de todas las distancias desde la observación al centroide correspondiente.

2.3. Formulaciones lineales del modelo ACSV

El problema (2.1) es no lineal ya que la función objetivo contiene el producto de las variables $z_k x_{ij}$. Benati y García [5] en su trabajo proponen dos procedimientos para linealizar la función objetivo.

2.3.1. Primera formulación: Linealización directa (ACSV-w)

Se definen las nuevas variables: $w_{ijk} = x_{ij} z_k, \quad i,j = 1,\dots,n, \quad k = 1,\dots,p.$
Esta variable toma el valor 1 si la observación i se asigna al clúster cuyo centroide es j y la variable k ha sido seleccionada. La formulación lineal del problema es la siguiente:

ACSV-w

$$\min \quad \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{k=1}^{p} d_{ijk} w_{ijk} \tag{2.2a}$$

sujeto a

$$x_{ij} \leqslant y_j, \qquad i,j = 1,\dots,n \tag{2.2b}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\dots,n \tag{2.2c}$$

$$\sum_{j=1}^{n} y_j = K \tag{2.2d}$$

$$\sum_{k=1}^{p} z_k = d \tag{2.2e}$$

$$w_{ijk} \geqslant x_{ij} + z_k - 1, \qquad i,j = 1,\dots,n, \ \ k = 1,\dots,p \tag{2.2f}$$

$$w_{ijk} \leqslant x_{ij}, \qquad i,j = 1,\dots,n, \ \ k = 1,\dots,p \tag{2.2g}$$

$$w_{ijk} \leqslant z_k, \qquad i,j = 1,\dots,n, \ \ k = 1,\dots,p \tag{2.2h}$$

$$w_{ijk} \geqslant 0, \qquad i,j = 1,\dots,n, \ \ k = 1,\dots,p \tag{2.2i}$$

$$x_{ij} \geqslant 0, \qquad i,j = 1,\dots,n \tag{2.2j}$$

$$y_j \in \{0,1\}, \qquad j = 1,\dots,n \tag{2.2k}$$

$$z_k \in \{0,1\}, \qquad k = 1,\dots,p \tag{2.2l}$$

En la función objetivo (2.2a) se ha sustituido el producto de las variables por las nuevas variables introducidas w_{ijk} y la distancia entre las dos observaciones en relación con cada variable. Las restricciones (2.2b)-(2.2e) y (2.2j)-(2.2l) coinciden con las del modelo ACSV. Ha sido preciso además definir nuevas restricciones (2.2f), (2.2g), (2.2h) y (2.2l) para garantizar que las variables w_{ijk} queden bien definidas.

- Las restricciones (2.2f) aseguran que si ambas x_{ij} y z_k toman valor 1, también w_{ijk} vale 1.

- Las restricciones (2.2g), (2.2h), conjuntamente, garantizan que, si al menos una de las dos variables x_{ij} o z_k son 0, entonces w_{ijk} también vale 0.

- Las restricciones (2.2l) son restricciones de signo de las variables w_{ijk}.

Notemos además que, aunque las variables x_{ij} son binarias, en el modelo basta con imponer que sean no negativas, porque al ser un problema de minimización en el que las distancias son positivas y el resto de variables son binarias, la solución óptima se va a obtener cuando las variables x_{ij} sean también binarias.

2.3.2. Segunda formulación: Formulación del radio (ACSV-r)

Las dos formulaciones previas, ACSV y ACSV-w tienen un alto coste computacional debido a la presencia de los mismos valores de las distancias en múltiples términos. Notemos que las distancias entre observaciones para cada variable, al tratarse de variables categóricas y más aún en el caso de binarias, toman un conjunto reducido de valores, es decir, se sitúan en un mismo radio. Por tanto se puede reducir en gran medida el coste computacional si se definen las variables adecuadas que permitan evitar estas multiplicidades.

Dada una observación i y una variable k, se ordenan la distancias $\{d_{i1k}, d_{i2k}, \ldots, d_{ink}\}$ de menor a mayor, y se eliminan todos los valores repetidos, de forma que, para cada observación y variable, se obtienen G_{ik} valores distintos. Sea D_{ik1} el menor valor, D_{ik2} el segundo menor, etc. Se obtiene así $0 = D_{ik1} < D_{ik2} < \ldots < D_{ikG_{ik}}$. Se introducen las variables binarias r_{ikt} que toman valor 1 si la variable k se selecciona y la observación i está a una distancia del centroide de al menos D_{ikt}, y 0 en caso contrario. La nueva formulación del modelo es la siguiente:

ACSV-r

$$\min \quad \sum_{i=1}^{n}\sum_{k=1}^{p}\sum_{t=2}^{G_{ik}} (D_{ikt} - D_{ik,t-1})\, r_{ikt} \tag{2.3a}$$

sujeto a

$$x_{ij} \leqslant y_j, \qquad i,j = 1,\ldots,n \tag{2.3b}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\ldots,n \tag{2.3c}$$

$$\sum_{j=1}^{n} y_j = K \tag{2.3d}$$

$$\sum_{k=1}^{p} z_k = d \tag{2.3e}$$

$$r_{ikt} + \sum_{(j|\ d_{ijk} < D_{ikt})} x_{ij} \geqslant z_k, \qquad i = 1,\ldots,n,\ k = 1,\ldots,p,\ t = 2,\ldots,G_{ik} \tag{2.3f}$$

$$r_{ikt} \geqslant 0, \qquad i = 1,\ldots,n,\ k = 1,\ldots,p,\ t = 2,\ldots,G_{ik} \tag{2.3g}$$

$$x_{ij} \geqslant 0, \qquad i,j = 1,\ldots,n \tag{2.3h}$$

$$y_j \in \{0,1\}, \qquad j = 1,\ldots,n \tag{2.3i}$$

$$z_k \in \{0,1\}, \qquad k = 1,\ldots,p \tag{2.3j}$$

Notemos que la función objetivo (2.3a) coincide con (2.2a) si se tiene en cuenta la definición de las variables D_{ikt} y r_{ikt}. Las restricciones (2.3b)-(2.3e) y (2.3h)-(2.3j) coinciden con las del modelo original (2.1). En el nuevo modelo se añaden las restricciones (2.3f) que imponen que, si se selecciona la variable k, entonces la observación i necesariamente o se asigna a un grupo designado por una mediana j que se encuentre a distancia $d_{ijk} < D_{ikt}$ o, si esto no es posible, se asigna a algún otro grupo que esté por lo menos a distancia D_{ikt}. Se añaden también las restricciones (2.3g) que son restricciones de signo de las

variables r_{ikt}.

El modelo anterior se ha formulado de forma general para variables categóricas, en particular, para variables que expresan una escala de Likert. A continuación, se propone el modelo correspondiente al caso en el que todas las variables categóricas son binarias. En este caso se simplifica notablemente. Al ser todas las variables binarias, las distancias entre dos observaciones para cada variable, d_{ijk}, toman sólo dos valores posibles, 0 o 1. Luego se tiene que $0 = D_{ik1} < D_{ik2} = 1 \quad$ y $G_{ik} = 2 \quad \forall i,k, \quad i \in 1,\ldots,n \quad k \in 1,\ldots,p$. En este caso solo es preciso definir como nuevas variables r_{ik}, prescindiendo del índice t. Las nuevas variables r_{ik} toman valor 1 si la variable k se selecciona y la observación i toma un valor distinto al centroide en esa variable, y 0 en caso contrario. El modelo simplificado queda así:

ACSV-rb

$$\min \quad \sum_{i=1}^{n}\sum_{k=1}^{p} r_{ik} \tag{2.4a}$$

sujeto a

$$x_{ij} \leqslant y_j, \qquad i,j = 1,\ldots,n \tag{2.4b}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\ldots,n \tag{2.4c}$$

$$\sum_{j=1}^{n} y_j = K \tag{2.4d}$$

$$\sum_{k=1}^{p} z_k = d \tag{2.4e}$$

$$r_{ik} + \sum_{(j|\ d_{ijk}<1)} x_{ij} \geqslant z_k, \qquad i = 1,\ldots,n,\ \ k = 1,\ldots,p \tag{2.4f}$$

$$r_{ik} \geqslant 0, \qquad i = 1,\ldots,n,\ \ k = 1,\ldots,p \tag{2.4g}$$

$$x_{ij} \geqslant 0, \qquad i,j = 1,\ldots,n \tag{2.4h}$$

$$y_j \in \{0,1\}, \qquad j = 1,\ldots,n \tag{2.4i}$$

$$z_k \in \{0,1\}, \qquad k = 1,\ldots,p \tag{2.4j}$$

Las restricciones (2.4f), que son análogas a las restricciones (2.3i), establecen que si la variable k es seleccionada, la observación i será asignada, si es posible, a un grupo cuyo centroide tenga el mismo valor en la variable k.

2.4. Restricciones adicionales en modelos de programación matemática

Una de las virtudes de los modelos de programación matemática es que permiten introducir con relativa facilidad restricciones adicionales. Estas restricciones pueden servir para garantizar propiedades deseables de las soluciones. Es posible, por ejemplo, acotar el número de variables que se puede seleccionar, exigir grupos con un número de observaciones equilibrado, descartar valores atípicos, restringir la variabilidad total, ...

En el capítulo 4 se introducen algunas restricciones que se han añadido en el caso de los datos analizados para el que era conveniente limitar qué variables de entre todo el conjunto podía seleccionar el modelo y se han comparado los resultados con los obtenidos con el modelo sin estas restricciones añadidas.

Capítulo 3

Métodos estadísticos de análisis clúster y reducción de la dimensión

3.1. Introducción

Con el propósito de comparar el procedimiento basado en programación matemática con un procedimiento estadístico se revisaron varias librerías del software estadístico R. Markos et al. [2] presentan el paquete de R *clustrd*, en el que se implementan métodos que combinan técnicas de reducción de la dimensión y análisis clúster tanto para datos continuos como categóricos. Estos autores señalan que la opción más habitual, conocida como *tandem approach*, consiste en aplicar la técnica de reducción de la dimensión y después el análisis clúster sobre los datos transformados. Aunque es una opción intuitiva y directa, la definición de los grupos puede no ser óptima dado que ambos métodos optimizan criterios diferentes. En el caso de datos categóricos, el paquete proporciona tres técnicas: MCA k-medias, i-FCB y el análisis de correspondencias y clúster.

En este capítulo se considera una colección de datos con n observaciones y p variables definidas sobre cada observación. El interés se centra, por un lado, en reducir la dimensión a d variables ($d < p$) y, por otro, en determinar la asignación de cada observación a uno de los K grupos, donde K es un número prefijado. Para ello, el procedimiento MCA k-medias combina el análisis de correspondencias múltiple y el método de k-medias, usando una combinación convexa de los dos criterios.

3.2. Análisis de correspondencias múltiples

Es una técnica de reducción de la dimensión diseñada para conjuntos de datos con variables categóricas, cuyo objetivo es resumir una gran cantidad de datos en un número reducido de variables, que se calculan como combinación lineal de las originales, con la menor pérdida de información [13].

En la aplicación del análisis de correspondencias simple se utiliza la matriz de frecuencias relativas, que se obtiene al dividir la frecuencia absoluta en cada celda de una tabla de contingencia para dos variables por el total de elementos observados. Como no todas las filas contienen el mismo número de datos, el método tiene en cuenta estas diferencias y asigna un mayor peso a las filas con más datos, para que queden bien representadas, aunque pueda ser a costa de una peor representación de aquellas con menos elementos.

El análisis de correspondencias múltiples es una extensión al análisis de correspondencias simples. Se aplica sobre una matriz $Z = [Z_1, Z_2, \ldots, Z_p]$, siendo Z_j la matriz $n \times q_j$, donde q_j es el número de

categorías o niveles de la variable j, de manera que un elemento representante de esta matriz es

$$z_{ij} = \begin{cases} 1, \text{ si la observación } i \text{ tiene la característica } j \\ 0, \text{ en caso contrario} \end{cases}$$

A partir de Z se construye la matriz $B = ZZ^t$ que recibe el nombre de matriz de Burt. Esta matriz se construye por superposición de cajas. En los bloques diagonales aparecen las frecuencias marginales de cada una de las variables analizadas. Fuera de la diagonal, aparecen tablas de frecuencias cruzadas para todas las posibles combinaciones 2 a 2 de las variables.

Sobre esta matriz se realiza el análisis de correspondencias simple, que consiste en encontrar la descomposición en valores propios singulares de una matriz de distancias obtenida a partir de la matriz B para construir un sistema de coordenadas asociado a las filas y columnas de la matriz que refleje las relaciones existentes entre dichas filas y columnas. Como resultado, el análisis de correspondencias asigna valores numéricos a las categorías de las variables. En otras palabras, convierte una variable categórica en una numérica. Es por ello que, aunque trabaja con variables categóricas, se puede utilizar en combinación con el algoritmo de k-medias que se explica a continuación, el cuál está diseñado para trabajar con variables numéricas.

3.3. Algoritmo de k-medias

Este algoritmo se diseñó para trabajar con variables numéricas continuas, aunque en la literatura se ha extendido para considerar variables categóricas [5]. Se consideran las observaciones en el espacio de dimensión p. Cada grupo $k \in \{1, \ldots, K\}$ se representa por su centroide en este espacio:

$$\bar{\mathbf{x}}_k = (\bar{x}_{1k}, \ldots, \bar{x}_{pk})$$

cuya coordenada j, sea $\bar{x}_{jk}$, se obtiene como la media de la variable j en todas las observaciones pertenecientes al grupo.

El algoritmo consta de 4 etapas ([8]):

1. Seleccionar K observaciones como centroides de los grupos iniciales.
2. Asignar cada observación al grupo de cuyo centroide está más próxima.
3. Para cada grupo k, calcular el nuevo centroide a partir de todas las observaciones asignadas al grupo.
4. Comprobar si es necesario asignar las observaciones a un grupo diferente si se mejora el criterio de optimalidad, esto es, está más próxima a otro centroide.

Los pasos 2, 3 y 4 se repiten hasta que no es posible mejorar el criterio de optimalidad y finaliza el algoritmo. Por tanto, este procedimiento determina una partición en K grupos tal que se minimiza la *suma de cuadrados dentro de los grupos (SCDG)* dada por:

$$SCDG = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \sum_{j=1}^{p} (x_{ijk} - \bar{x}_{jk})^2$$

donde x_{ijk} es el valor de la variable j en la observación i asignada al grupo k, $\bar{x}_{jk}$ es la media de la variable j en el grupo k y n_k es el número de elementos en el grupo.

El resultado del algoritmo depende de la asignación inicial y del orden de las observaciones. Por ello, es aconsejable repetir el algoritmo para varios valores iniciales y permutar los elementos de la

muestra. También es esencial seleccionar un número óptimo de grupos K. En la práctica, para determinar el valor de K, se aplica el algoritmo con varios valores de K y se comparan los resultados para determinar con qué valor se obtiene la mejor solución.

Un criterio equivalente, conocido como *criterio de la traza*, fue propuesto por Ward en 1963 [3]. Se define la matriz W como:

$$\mathbf{W} = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\mathbf{x}_{ik} - \bar{\mathbf{x}}_k)(\mathbf{x}_{ig} - \bar{\mathbf{x}}_g)^t$$

entonces:

$$\min\ SCDG = \min\ tr(\mathbf{W})$$

El criterio de la traza no es invariante ante cambios de escala, por ello si los datos están medidos en distintas unidades es recomendable estandarizarlos. Sin embargo, cuando las unidades coinciden, el algoritmo proporciona mejores resultados si no se estandariza. Si existe una variable con una varianza mucho mayor que el resto, puede ser consecuencia precisamente de que existen dos o más grupos de observaciones en esa variable y conviene por tanto que la variable tenga un peso importante en la construcción de los grupos, información que se perdería en caso de estandarizar.

3.4. Método MCA k-medias

Este procedimiento trata de forma conjunta la reducción de la dimensión y la determinación de los grupos de las observaciones en dicha dimensión [2, 7] en el caso de datos categóricos.

Dada una variable categórica j ($j = 1, \ldots, p$) con q_j categorías o valores posibles, la matriz $\mathbf{Z}_j$ es una matriz indicadora de dimensión $n \times q_j$ de forma que $z_{ic} = 1$ si en la observación i está presente la categoría c y 0, en caso contrario.

Sea $P = \sum_{j=1}^{p} q_j$ y $\mathbf{Z}$ una matriz binaria superindicadora de dimensión $n \times P$:

$$\mathbf{Z} = [\mathbf{Z}_1, \ldots, \mathbf{Z}_p]$$

Se introduce para cada variable categórica j ($j = 1, \ldots, p$) una matriz $\mathbf{B}_j$ de dimensión $q_j \times d$ que proporciona los pesos que cuantifican cada categoría de la variable en el espacio de dimensión reducida. A partir de estas matrices se define una matriz $\mathbf{B}$ de dimensión $P \times d$:

$$\mathbf{B} = [\mathbf{B}_1^t, \ldots, \mathbf{B}_p^t]^t$$

Sea $\mathbf{Y}$ la matriz de dimensión $n \times d$ que proporciona los valores de las observaciones en el espacio reducido de dimensión d. La matriz indicadora, $\mathbf{U}_K$, de dimensión $n \times K$ determina la pertenencia de cada observación a uno de los grupos. Finalmente, se introduce la matriz $\mathbf{G}$ de dimensión $K \times d$ que proporciona los centroides o valores medios de las observaciones de cada grupo en el espacio de dimensión reducida d.

El procedimiento combina los objetivos del análisis de correspondencias múltiple y el algoritmo k-medias proponiendo una función objetivo como combinación lineal convexa de ambos , con $\alpha \in [0, 1]$:

$$\min_{(\mathbf{Y}, \mathbf{B}_j, \mathbf{G}, \mathbf{U}_K)} \phi_{mcak} = \alpha \frac{1}{p} \sum_{j=1}^{p} \|\mathbf{Y} - \mathbf{Z}_j \mathbf{B}_j\|^2 + (1 - \alpha) \|\mathbf{Y} - \mathbf{U}_K \mathbf{G}\|^2 \tag{3.1}$$

sujeta a la condición $\mathbf{Y}^t \mathbf{Y} = \mathbf{I}_d$.

El peso α permite priorizar uno u otro procedimiento, esto es, la reducción de la dimensión o la determinación de los grupos:

- Con $\alpha = 0$, la función objetivo minimiza $\|\mathbf{Y} - \mathbf{U}_K\mathbf{G}\|^2$, donde $\mathbf{Y}$ guarda la proyección de las observaciones en el espacio de dimensión reducida y $\mathbf{U}_K\mathbf{G}$ guarda las coordenadas de los centroides en el espacio de menor dimensión. Es evidente que el objetivo es minimizar la distancia entre cada observación y su centroide, que coincide con el objetivo del algoritmo k-medias.

- Con $\alpha = 1$, la función objetivo consiste en minimizar $\frac{1}{p}\sum_{j=1}^{p}\|\mathbf{Y} - \mathbf{Z}_j\mathbf{B}_j\|^2$. En este caso, como $\mathbf{Z}_j\mathbf{B}_j$ es la proyección de cada individuo en el espacio de dimensión reducida atendiendo únicamente a la variable j, la función objetivo minimiza la suma de las distancias entre la proyección de los datos respecto a todas las variables y respecto a sólo la variable j. Es decir, determina cuál es la mejor forma de proyectar los datos en un espacio de menor dimensión de manera que estén lo mejor representados posible respecto a todas las variables, perdiendo la mínima información, que coincide con el objetivo del MCA.

- Si $\alpha = 0{,}5$, el objetivo de clasificar correctamente los datos tiene el mismo peso en el estudio que la reducción de la dimensionalidad.

Las etapas del algoritmo son:

1. Generar una partición inicial de los datos, recogida en la matriz $\mathbf{U}_K$ y utilizar MCA para obtener una solución inicial $\mathbf{Y}$.

2. Calcular las matrices de pesos por categorías, $\mathbf{B}_j$ con $j = 1, \ldots, p$ y los $\mathbf{K}$ centroides.

3. Actualizar $\mathbf{Y}$.

4. Actualizar $\mathbf{U}_K$ aplicando el algoritmo k-medias a la matriz $\mathbf{Y}$.

Las etapas 2 a 4 se repiten hasta que el procedimiento converge y $\mathbf{U}_K$ es constante.

Capítulo 4

Experiencia computacional

4.1. Introducción

En este capítulo se van a comparar los resultados obtenidos al aplicar los modelos propuestos en los capítulos 2 y 3, tanto desde la perspectiva de los métodos clúster y selección de variables como del modelo de programación matemática, que ha sido implementado en el software de optimización CPLEX, cuyo código se ha incluido en el anexo A. La comparación se realiza a partir de dos colecciones de datos que se han diseñado adhoc ajustándose cada una de ellas a uno de los dos procedimientos.

4.2. Conjuntos de datos

Se describen en este apartado las características de los conjuntos de datos generados.

4.2.1. Conjuntos de datos Diamond

La primera colección de datos generada utiliza un conjunto de datos ilustrativo disponible dentro de la librería de R *clustrd*, cuyo nombre es *diamond*, y que recoge información sobre 308 diamantes en 4 variables categóricas y una numérica.

Se ha descartado la variable numérica, que proporciona el precio pagado por los diamantes, puesto que los algoritmos utilizados trabajan exclusivamente con variables categóricas. El método MCA k-medias se aplica a conjuntos de variables categóricas con varios niveles. Sin embargo, el modelo ACSV está diseñado para variables binarias. Por ello, para analizar los datos con el modelo ACSV se van a convertir en binarias todas las variables categóricas, definiendo cada categoría de cada variable como una nueva variable binaria, que tome valor 1 en caso que la observación tenga esa categoría, y cero en caso contrario.

Las variables recogidas en la colección de datos y sus categorías son las siguientes:

- Peso: los diamantes se dividen en pequeños (si su peso es menor de 0.5 kilates), medianos (pesan entre 0.5 y 1 kilate) y grandes (pesan más de 1 kilate). Estas variables se han denominado con el nombre de las categorías: Peq, Med, Gr.

- Color: la gama de colores va desde D (completamente incoloros), E, F, ... hasta I (casi incoloros). En este caso, las categorías se denominan D, E, ... I.

- Claridad: recoge las imperfecciones internas y externas de los diamantes. Se divide en IF (sin imperfecciones internas), VVS1 y VVS2 (muy muy levemente imperfectos) y VS1 y VS2 (muy levemente imperfectos). Las categorías se denominan IF, VVS1, ... VS2.

- Certificación: indica qué organismo ha certificado los diamantes. Los organismos que se consideran son el Instituto Gemológico Americano (GIA), el Instituto Internacional de Gemología (IGI) y los diamantes Hoge Raad Voor (HRD) , que se denominan GIA, IGI y HRD.

Debido al elevado coste computacional que requerían los modelos para encontrar la solución óptima del modelo ACSV para este conjunto de datos, se decidió crear 3 subconjuntos de 100 observaciones cada uno, escogidas aleatoriamente de entre las 308 observaciones del conjunto *diamond*, y que hemos denotado D1, D2 y D3.

4.2.2. Conjuntos de datos Brusco

En la literatura, el procedimiento de programación entera se ha probado sobre conjuntos de datos simulados. Esto permite validar la bondad del procedimiento, al conocer a priori cuáles son las variables verdaderas y los grupos definidos. En este trabajo se ha considerado la generación de datos propuesta por Brusco [10] y que también se utiliza en el trabajo de Benati y García [11]. A esta colección la hemos denominado Brusco.

Al igual que en el conjunto Diamond, se han generado conjuntos de 100 observaciones y 17 variables binarias. En cuanto al número de variables verdaderas que caracterizan los elementos de cada grupo se ha considerado los valores 3, 6 y 9. En todos los casos, el número de grupos es 3 y se han generado 46, 31 y 23 observaciones, en cada uno de los grupos. En la Tabla 4.1 se muestra para cada grupo cuáles son los valores de cada variable verdadera.

Tabla 4.1: Valores que toman las observaciones de cada grupo en las variables verdaderas.

		Variables verdaderas								
		1	2	3	4	5	6	7	8	9
Grupos	1	1	1	0	1	1	0	1	1	0
	2	1	0	1	1	0	1	1	0	1
	3	0	1	0	0	1	0	0	1	0

Por ejemplo, cuando se consideran solo 3 variables verdaderas, cualquier individuo del grupo 2 tomará valores 1, 0 y 1 en cada una de ellas. El resto de variables hasta 17 toman un valor obtenido a partir de una Bernoulli de parámetro 0,5. De esta forma, en principio, lo único que tienen en común los individuos de un grupo son los valores de las variables verdaderas. En el Anexo C se incluye el código en R utilizado para la construcción de 9 conjuntos de datos, ya que se han generado 3 colecciones de datos para cada valor del número de variables verdaderas y se han denominado B1, ..., B9.

4.3. Aplicación de los modelos

4.3.1. Aproximación multiobjetivo del método MCA *k*-medias

La función objetivo, sea Z, del método MCA k-medias es una suma ponderada de dos funciones objetivo (3.1). Este es un tratamiento habitual en programación multiobjetivo. Los autores proponen ejecutar el método utilizando el valor de $\alpha = 0,5$. En este apartado, se ha trabajado con el fichero Diamond completo y únicamente con el procedimiento MCA k-medias, con el uso de la librería *clustrd*, con el propósito de analizar las diferencias entre los valores de la función objetivo y las soluciones propuestas cuando se modifica el valor de α. En todos ellos, se ha considerado la definición de tres grupos y se ha reducido la dimensión a 2 variables, que era el valor recomendado por la función *tuneclus()*

disponible en la misma librería.

Uno de los objetivos, sea Z_1, se refiere a la pérdida de información cuando se reduce el número de variables al aplicar el método MCA:

$$Z_1 = \frac{1}{p}\sum_{j=1}^{p} \|\mathbf{Y} - \mathbf{Z}_j\mathbf{B}_j\|^2$$

El segundo, que se denota por Z_2, recoge las diferencias entre las observaciones y los centroides de su grupo, a partir del procedimiento *k*-medias:

$$Z_2 = \|\mathbf{Y} - \mathbf{Z}_K\mathbf{G}\|^2$$

De este modo,

$$Z = \alpha Z_1 + (1-\alpha)Z_2.$$

El objetivo de la programación multiobjetivo es encontrar una solución factible que minimice ambos objetivos. Sin embargo, esta solución suele ser inalcanzable cuando los objetivos son conflictivos. Se busca entonces encontrar un conjunto de soluciones factibles no dominadas, que cuando las funciones son convexas se pueden obtener al cambiar el peso dado a cada objetivo, en este caso, al mover el valor de α [9]. Este conjunto de soluciones se conoce como *conjunto de Pareto*. Una solución es eficiente u óptima Pareto si no es posible encontrar otra solución factible tal que sea no peor en todos los objetivos y mejor estrictamente en al menos uno. En otras palabras, una solución es óptima Pareto en un problema con dos objetivos si en caso de existir otra solución que mejore en un criterio, será a costa de empeorar en otro.

Se ha modificado el código de R de la función *clusmca()*, de forma que devuelva los valores de Z, Z_1 y Z_2 por separado. Las modificaciones realizadas se pueden ver en el anexo D. Las nuevas funciones se han aplicado al conjunto de datos Diamond, moviendo el valor de α entre 0 y 1, con pasos de 0,1. Para los extremos del intervalo, en vez de tomar $\alpha = 0$ y $\alpha = 1$, que requerían realizar cambios más complejos en el código de R, se han tomado las aproximaciones $\alpha = 0{,}001$ y $\alpha = 0{,}999$. Los resultados obtenidos se muestran en la Tabla 4.2.

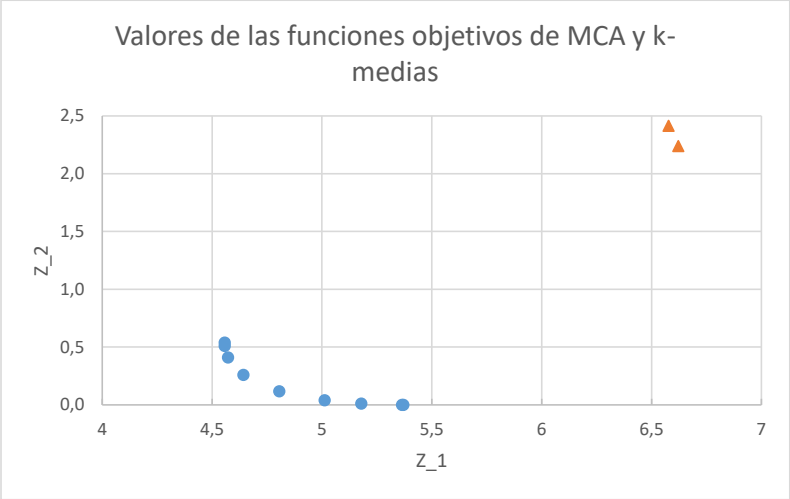
Tabla 4.2: Valor que toman las distintas funciones objetivo y porcentaje de variabilidad explicada por el modelo de agrupamiento, según el valor de α.

α	Z	Z_1	Z_2	VE
0,0001	5,370543e-04	5,3705430	-5,551115e-16	100,00
0,1	0,536444	5,3643580	9,530846e-06	100,00
0,2	3,1148300	6,6224950	2,2379140	50,00
0,3	3,6630690	6,5783910	2,4136460	49,98
0,4	2,0776073	5,1788976	0,0100804	99,50
0,5	2,5256672	5,0118744	0,0394601	98,03
0,6	2,9312440	4,8063790	0,1185415	94,07
0,7	3,3277526	4,6428121	0,2592806	87,04
0,8	3,7398048	4,5721473	0,4104349	79,48
0,9	4,1530134	4,5578595	0,5093987	74,53
0,9999	4,5567253	4,5571271	0,5395769	73,02

Como era de esperar, con $\alpha = 0{,}0001$, puesto que el método se centra en priorizar el algoritmo *k*-medias, Z_2 toma los valores más pequeños, y viceversa para Z_1, que ha de tomar su máximo valor.

Progresivamente, conforme aumenta el valor de α, deberían ir 'cambiando los papeles' entre ambas hasta llegar a $\alpha = 0,9999$. Con $\alpha = 1$ el método sigue la aproximación *tandem*, dando prioridad a la parte de MCA, y Z_1 debería alcanzar su mínimo. De igual manera para la última columna, que se corresponde con el porcentaje de variabilidad explicada, se esperaría que para $\alpha = 0,0001$, que prioriza la agrupación de las observaciones, el modelo diese una buena explicación de las asignaciones que realiza, y que este porcentaje fuera empeorando conforme se reduce el peso del método de agrupamiento. Sin embargo, destacan las soluciones para $\alpha = 0,2$ y $\alpha = 0,3$, que no concuerdan con lo esperado. De hecho, como se observa en la Figura 4.1, estas soluciones están dominadas por el resto de soluciones obtenidas, en el sentido de que se mejoran ambos objetivos para cualquier otro valor de α. También son llamativos los porcentajes bajos de variabilidad explicada para estos dos mismos valores de α.

Figura 4.1: Valores obtenidos de las funciones objetivo de MCA y *k*-medias. En naranja con forma de triángulo, aparecen los valores cuando $\alpha = 0,2$ y $0,3$.



Notemos que, por ejemplo, para $\alpha = 0,4$ se obtiene una solución, sea x^*, que cumple que el valor de Z_1 es 5,17889758 y el de Z_2 es 0,01008044. Esta solución es la que el algoritmo considera óptima para $\alpha = 0,4$, por lo que es una solución factible. Por tanto, también debe ser una solución factible cuando cambiamos el valor de α, por ejemplo a 0,2. Se tiene entonces que en x^* el valor de la función objetivo conjunta es $0,2 \times 5,17889758 + 0,8 \times 0,01008044 = 1,043843868 < 3,114830$, siendo este último valor el que nos da como resultado del óptimo para 0,2. Hemos encontrado así un punto, x^*, que mejora el valor de la función objetivo según los resultados proporcionados por R.

Si se mira con mayor detalle el código fuente de R para esta función, puede verse como sus autores proponen un acercamiento heurístico para calcular los valores de estas funciones. En el cálculo de Z, Z_1 y Z_2, se sigue un proceso iterativo. Se crea un bucle que acaba sólo cuando hay convergencia en Z, es decir, cuando la diferencia del valor calculado de Z en dos iteraciones sucesivas es menor que un valor que denominan *ceps*. Este puede ser el motivo de que las soluciones proporcionadas por el algoritmo no sean realmente óptimas, sino aproximaciones más o menos cercanas a los óptimos reales, que para el caso particular de este conjunto de datos no son suficientemente finas.

4.3.2. Comparación entre ACSV-w y ACSV-rb

En este apartado se compara, con los conjuntos de datos simulados Brusco, las diferencias en el coste computacional de las dos alternativas de linealización del problema de programación entera ACSV. Los resultados de este experimento de comparación se incluyen en la Tabla 4.3. Se ha establecido un límite de computación de 1 hora. Todos aquellos conjuntos de datos para los que el modelo no es capaz de alcanzar la solución óptima en ese tiempo, se muestra en la columna correspondiente como tiempo >1h. Para cada una de las formulaciones se muestra en la columna indicada por Z^* el valor óptimo de la función objetivo cuando la resolución ha finalizado antes de 1 hora y el mejor valor obtenido, si la ejecución en CPLEX se ha interrumpido tras una hora.

Tabla 4.3: Resultados obtenidos al aplicar las dos formulaciones del modelo ACSV al conjunto de datos Brusco.

		Modelo ACSV-w		Modelo ACSV-rb	
Conjunto	Nº de variables verdaderas	Z^*	T_{opt} (segundos)	Z^*	T_{opt} (segundos)
1	3	92	>1h	0	301
2	3	0	3600	0	2629
3	3	70	>1h	0	365
4	6	43	>1h	0	296
5	6	180	>1h	0	562
6	6	177	>1h	0	153
7	9	100	>1h	0	16
8	9	307	>1h	0	13
9	9	0	1665	0	13

De la Tabla 4.3 se concluye que la formulación ACSV-rb es mucho más eficiente que la formulación ACSV-w, en especial en los conjuntos de datos con 9 variables verdaderas. Cuanto menor es el número de variables verdaderas mayor es el tiempo necesario para alcanzar el óptimo. En relación con la solución óptima, la formulación ACSV-rb ha sido capaz de seleccionar correctamente las variables verdaderas y asigna adecuadamente cada observación en su grupo correspondiente.

4.3.3. Comparación entre MCA *k*-medias y ACSV en los conjuntos Diamond

En primer lugar se considera la colección de datos Diamond y se aplica el método MCA *k*-medias para el que estaban originalmente propuestos. A continuación, se comparan los resultados con los obtenidos al aplicar el modelo ACSV.

Para decidir el número de grupos y la dimensión reducida, la librería *clustrd* tiene incorporada la función *tuneclus()*, que sugiere para qué valores de estos parámetros se obtiene la mejor solución. Para estos conjuntos de datos, los mejores resultados se obtienen con 3 grupos en 2 dimensiones. Se aplicó entonces el método MCA *k*-medias, con el valor de $\alpha = 0,5$, que es la opción por defecto. Además, se utilizó también la función *plot()*, que devuelve un gráfico de dispersión de las observaciones proyectadas sobre las dimensiones reducidas y unos gráficos de barras que representan los mayores residuos estandarizados al contrastar la independencia de cada categoría en cada grupo. Si una categoría tiene un residuo grande en un grupo, se puede interpretar como que aparece un número de veces superior a la media, y por tanto caracteriza el grupo. Los resultados obtenidos se han resumido en las Tablas 4.4, 4.5 y 4.6 y la salida de R se puede ver con más detalle en el anexo B.

Tabla 4.4: Las tres variables con mayor peso (en valor absoluto) en cada dimensión para cada conjunto de datos Diamond.

	D1		D2		D3	
Dimensión	Variable	Peso	Variable	Peso	Variable	Peso
Dim1	Certif.IGI	0,114	Certif.IGI	0,123	Clard.IF	0,146
	Clard.IF	0,111	Clard.IF	0,108	Certif.IGI	0,121
	Tam.Peq	0,100	Tam.Peq	0,088	Tam.Peq	0,103
Dim2	Clard.VVS1	0,098	Certif.HRD	0,097	Certif.HRD	0,084
	Certif.HRD	0,084	Clard.VS1	0,079	Color.D	0,083
	Clard.VS1	0,084	Certif.GIA	0,078	Clard.VS1	0,078

Tabla 4.5: Coordenadas de los centroides y número de observaciones en cada conjunto de datos Diamond.

	D1			D2			D3		
Grupo	Dim1	Dim2	Nº obs	Dim1	Dim2	Nº obs	Dim1	Dim2	Nº obs
1	0,087	0,110	40	0,071	0,116	40	-0,041	0,119	43
2	0,054	-0,162	31	0,052	-0,138	36	-0,066	-0,123	38
3	-0,178	0,021	29	-0,196	0,014	24	0,223	-0,023	19

Tabla 4.6: Las tres variables principales que caracterizan cada grupo.

Grupo	D1	D2	D3
1	Certif.GIA	Certif.GIA	Clard.VS1
	Clard.VS1	Clard.VS1	Certif.GIA
	Tam.Gr	Clard.VS2	Tam.Gr
2	Certif.HRD	Certif.HRD	Certif.HRD
	Clard.VVS1	Tam.Gr	Tam.Med
	Tam.Med	Clard.VVS2	Color.G
3	Certif.IGI	Certif.IGI	Certif.IGI
	Tam.Peq	Tam.Peq	Tam.Peq
	Clard.IF	Clard.IF	Clard.IF

En la Tabla 4.4 se observa que en la dimensión 1 son siempre las mismas categorías las que tienen mayor peso, y no ocurre lo mismo en la dimensión 2. Esta diferencia se traslada a cómo se realiza la asignación de las observaciones a los grupos. En la Tabla 4.5, atendiendo a las coordenadas de los centroides, los grupos 1 y 2 se caracterizan por tener un valor absoluto alto en la segunda dimensión, mientras que el grupo 3 tiene un valor absoluto alto en la primera. Es por ello que el grupo 3 lo caracterizan las mismas categorías en los tres conjuntos, mientras que la caracterización de los grupos 1 y 2 varía más según el conjunto de datos. Como resumen final, en la Tabla 4.6 se muestra para cada conjunto de datos Diamond las tres categorías más relevantes al determinar los tres grupos. En general, se observa que las categorías de certificación son las más importantes a la hora de identificar los grupos.

A continuación se aplica el modelo de programación matemática. Los cálculos se han realizado con la formulación del modelo ACSV-rb por ser más eficiente. En el modelo se ha impuesto definir 3 grupos

identificando 3, 6 y 9 variables verdaderas. Los resultados obtenidos se pueden ver en la Tabla 4.7.

Tabla 4.7: Resultados obtenidos al aplicar ACSV-rb sobre los conjuntos de datos Diamond, según el número de variables seleccionadas. Las variables Peq, Med, Gr se refieren al tamaño; D, E, F, G, H, I al color; IF, VVS1, VVS2, VS1, VS2 a la claridad y GIA, IGI y HRD a la certificación.

Conjunto	Nº var. selec.	Variables	Z_{opt}	T_{opt} (segundos)
D1	3	Peq, Med, Gr	0	1
D2	3	GIA, IGI, HRD	0	1
D3	3	GIA, IGI, HRD	0	1
D1	6	Peq, Med, Gr, D, I, IGI	31	223
D2	6	D,I,VVS1,GIA, IGI, HRD	51	187
D3	6	eq, Med, Gr, D, IF, VS2	28	190
D1	9	Peq, Med, Gr, D, E, I, IF, VVS1, IGI	77	239
D2	9	D,E,I,IF,VVS1,VS2,GIA,IGI,HRD	77	158
D3	9	Peq, Med, Gr, D, E, G, IF, VS2, IGI	76	106

Como se ha comentado, el procedimiento de programación matemática está diseñado para trabajar sobre datos de naturaleza binaria, que no es el caso del conjunto Diamond. En este conjunto hay variables que tienen exactamente 3 niveles. Al binarizarlas e indicar al modelo la selección de tres variables verdaderas, puede ocurrir que las tres correspondan a la misma variable. Se observa en la Tabla 4.7 para la variable Tamaño y las variables seleccionadas en el conjunto D1. En este caso, el valor óptimo es 0 porque todas las observaciones están bien clasificadas, al pertenecer seguro a una y solo una de las categorías. Sucede lo mismo en los conjuntos D2 y D3 con la variable Certificación y sus categorías. De hecho, incluso cuando se escogen 9 variables, el método clasifica las observaciones atendiendo sólo a las tres variables de tamaño o certificación, según cuáles de ellas se hayan seleccionado. Esto se confirma al observar los valores que toman los centroides de cada grupo en las variables seleccionadas. En la Tabla 4.8 se incluye el valor de los centroides en el caso de seleccionar 9 variables verdaderas en el conjunto D3. En el grupo 3 están los pequeños, los medianos en el grupo 1 y los grandes en el 2.

Tabla 4.8: Valores que toman los centroides de cada grupo en las 9 variables verdaderas seleccionadas para el conjunto D3.

		Variables seleccionadas								
		Peq	Med	Gr	D	E	IF	VS2	IGI	HRD
Grupos	1	0	1	0	0	0	0	0	0	0
	2	0	0	1	0	0	0	0	0	0
	3	1	0	0	0	0	0	0	0	0

El efecto anterior es debido a la binarización realizada. Para reducir este efecto se han añadido restricciones al modelo de programación matemática sobre el número de variables binarias que podía seleccionar en cada una de las categorías. Estas restricciones exigen que, para cada variable categórica, el número máximo de categorías que puede seleccionar es inferior al total. Los resultados obtenidos al aplicar estas restricciones se incluyen en la Tabla 4.9.

Como era de esperar, al restringir el espacio de factibilidad, los valores de Z_{opt} mostrados en la Tabla 4.9 son peores que los correspondientes de la Tabla 4.7 y también ha aumentado el tiempo computacio-

Tabla 4.9: Resultados obtenidos al aplicar ACSV-rb restringido sobre los conjuntos de datos Diamond, según el número de variables seleccionadas. Las variables Peq, Med, Gr se refieren al tamaño; D, E, F, G, H, I al color; IF, VVS1, VVS2, VS1, VS2 a la claridad y GIA, IGI y HRD a la certificación.

Conjunto	Nº var. selec.	Variables	Z_{opt}	T_{opt} (segundos)
D1	3	Peq, Med, D	5	174
D2	3	D, F, G	5	139
D3	3	D, IGI, HRD	7	86
D1	6	Med, Gr, D, I, IF, IGI	45	265
D2	6	Peq, Med, D, I, IF, VSS1	44	258
D3	6	D, E, IF, VS2, GIA, HRD	40	251
D1	9	Med, Gr, D, E, I, IF, VVS1, VS2, IGI	95	279
D2	9	Gr, D, E, I, IF, VVS1,VS2, GIA, IGI	95	280
D3	9	Peq, D, E, G, IF, VVS1, VS2, GIA, HRD	94	283

nal. Estos resultados se han interpretado en términos de los valores que toman los centroides de cada grupo en las variables seleccionadas. Dadas las limitaciones de espacio, se expone a continuación el proceso realizado para el conjunto D3 al seleccionar 9 variables verdaderas.

Los valores que toman los centroides de cada grupo en las variables seleccionadas se muestran en la Tabla 4.10. Las observaciones asignadas al grupo 1 se caracterizan por tener la certificación GIA, y las del grupo 2 por tener la certificación HRD. En el caso del grupo 3, el centroide toma valores en la categoría tamaño pequeño y claridad IF. Sin embargo, puesto que toma valores 0 en las certificaciones GIA y HRD, se deduce que necesariamente tomará valor 1 en la categoría de certificación IGI. Por tanto el modelo con restricciones ha sido igualmente capaz de seleccionar las tres variables de certificación para determinar los grupos, pero además permite seleccionar otras como el tamaño pequeño o la claridad IF, que son también características del grupo 3. Se consigue de esta forma, a partir del mismo número de variables seleccionadas, una mayor cantidad de información sobre las características de los grupos. En este caso, la asignación realizada por MCA *k*-medias y ACSV-rb coincide en un total de 73 de las 100 observaciones

Tabla 4.10: Valores que toman los centroides de cada grupo en las variables verdaderas del conjunto D3.

		Variables seleccionadas								
		Peq	D	E	G	IF	VVS1	VS2	GIA	HRD
Grupos	1	0	0	0	0	0	0	0	1	0
	2	0	0	0	0	0	0	0	0	1
	3	1	0	0	0	1	0	0	0	0

El proceso anterior se ha realizado sobre los tres conjuntos de datos con 3, 6 y 9 variables verdaderas. Las categorías que definen cada grupo se muestran de manera resumida en la Tabla 4.11.

4.3.4. Comparación entre MCA *k*-medias y ACSV-rb en los conjuntos Brusco.

En este apartado se comparan el modelo ACSV-rb con MCA *k*-medias con las colecciones de datos Brusco. En la Tabla 4.3 se han incluido los resultados de aplicar ACSV-rb. El procedimiento es capaz de identificar las variables verdaderas y agrupar correctamente las observaciones. Se emplea ahora el

Tabla 4.11: Variables principales que caracterizan cada grupo de los conjuntos de datos Diamond. Las variables Peq, Med, Gr se refieren al tamaño; D, E, F, G, H, I al color; IF, VVS1, VVS2, VS1, VS2 a la claridad y GIA, IGI y HRD a la certificación.

Nº var selec	Grupos	D1	D2	D3
3	1	Med	F	GIA
	2	Gr	Color.G	HRD
	3	Peq	D, E, H, I	IGI
6	1	Med	Med	GIA
	2	Gr	Gr	HRD
	3	Peq, IF, IGI	Peq	IGI
9	1	Med	GIA	GIA
	2	Gr	HRD	HRD
	3	Peq, IF, IGI	Peq, IGI	Peq, IF, IGI

método MCA *k*-medias. El interés en este caso reside en determinar si este método es capaz de identificar correctamente los grupos o si se producen efectos de enmascaramiento debidos a la presencia de las variables ruido. Los resultados obtenidos se recogen en la Tabla 4.12.

Tabla 4.12: Resultados obtenidos al aplicar el método MCA *k*-medias a la colección de datos Brusco.

			Nº obs. mal clasificadas			
Conjunto	Nº variables verdaderas	Comete errores	Grupo1	Grupo2	Grupo3	Total
B1	3	Sí	14	1	9	24
B2	3	Sí	10	3	7	20
B3	3	Sí	9	0	9	18
B4	6	Sí	1	2	3	6
B5	6	Sí	1	0	0	1
B6	6	Sí	3	0	2	5
B7	9	No	-	-	-	-
B8	9	No	-	-	-	-
B9	9	No	-	-	-	-

El análisis de los resultados indica que en aquellos conjuntos de datos con menor número de variables verdaderas, y por tanto, más variables ruido, como los tres primeros, se producen un número de errores de clasificación notable. Conforme aumenta el número de variables verdaderas, y aumenta por tanto el ratio entre variables verdaderas y variables ruido, el método es cada vez más capaz de determinar correctamente los grupos.

A continuación, se analiza con mayor detalle los conjuntos con 3 variables verdaderas, que son con los que se han producido el mayor número de errores de clasificación de las observaciones. En este caso, también se ha ejecutado el método indicando la determinación de 3 grupos y un espacio de dimensión reducida igual a 2. Los resultados obtenidos se resumen en las Tablas 4.13 y 4.14.

La Tabla 4.13 incluye los coeficientes de la combinación lineal de las variables que da lugar a la

Tabla 4.13: Las tres variables con mayor peso (en valor absoluto) en cada dimensión para los conjuntos de datos Brusco.

	B1		B2		B3	
Dimensión	Variable	Peso	Variable	Peso	Variable	Peso
Dim1	Var2	0,149	Var2	0,143	Var2	0,152
	Var3	0,149	Var3	0,143	Var3	0,152
	Var1	0,096	Var1	0,093	Var1	0,098
Dim2	Var11	0,071	Var14	0,066	Var7	0,049
	Var17	0,072	Var4	0,069	Var10	0,054
	Var12	0,082	Var7	0,055	Var1	0,044

Tabla 4.14: Coordenadas de los centroides de cada grupo al aplicar MCA *k*-medias sobre los conjuntos de datos Brusco.

	B1		B2		B3	
Grupo	Dim1	Dim2	Dim1	Dim2	Dim1	Dim2
1	0,071	0,002	0,076	0,005	0,071	0,006
2	-0,058	-0,090	-0,063	0,082	-0,054	-0,082
3	-0,062	0,087	-0,051	-0,093	-0,069	0,098

dimensión 1 y se observa que en los tres conjuntos el método da más peso a las variables verdaderas. Las diferencias surgen en la dimensión 2, donde son otras variables las que tienen más peso. Como consecuencia, si se analizan las coordenadas de los centroides en las dos dimensiones mostradas en la Tabla 4.14, el grupo 1, cuyo centroide se caracteriza por tener un valor alto en la dimensión 1 y un valor bajo en la dimensión 2, es el grupo con el que comete menos errores al realizar la clasificación. Sin embargo, los otros dos grupos de observaciones, cuyos centroides sí que toman un valor alto en la dimensión 2, se confunden y clasifican incorrectamente más frecuentemente.

Conforme aumenta el número de variables verdaderas, el método asigna los mayores pesos a las variables verdaderas en ambas dimensiones. Es por esto que deja de cometer errores de clasificación para los conjuntos con 9 variables verdaderas.

4.4. Conclusión y trabajo futuro

El estudio realizado ha permitido conocer las aproximaciones desde dos disciplinas al mismo problema, desde la perspectiva de la programación matemática y la estadística. Aunque cada una de ellas tenía diferente objetivo y proporciona mejores resultados cuando se aplica al tipo de datos para el ha sido diseñada. Se ha podido identificar las bondades e incovenientes de cada una de ellas.

En el caso del método MCA *k*-medias, hay que tener en cuenta que es un procedimiento heurístico, y que los resultados que proporciona son aproximaciones a los óptimos reales. Una de las grandes ventajas de este método reside en el resto de funciones que vienen implementadas en la librería clustrd y que complementan su aplicación. Definir el número de grupos en una muestra sobre la que no se tiene información es una de las decisiones más importantes que se debe tomar a la hora de aplicar un método de análisis clúster, y a la vez, una de las más difíciles. La función tuneclus() resulta ser una herramienta

muy útil en este caso para determinar el número de grupos más apropiado. Por otra parte, las opciones gráficas que ofrece la librería también son un herramienta muy útil para visualizar las características que definen los distintos grupos presentes en la muestra. Sin embargo, al trabajar con conjuntos de datos en los que existe un alto número de variables ruido, se pueden cometer un número alto de clasificaciones incorrectas. En ese caso, los modelos de selección de variables proporcionan mejores resultados.

Por su parte, las formulaciones lineales del modelo ACSV han mostrado ser muy útiles a la hora de determinar los grupos en conjuntos de datos binarios con un alto número de variables ruido, como es el caso de la colección de datos Brusco con la que se ha trabajado. Especialmente con la formulación del radio, ACSV-rb, el modelo es capaz de alcanzar la solución óptima de conjuntos de datos relativamente grandes en un intervalo de tiempo corto. Sólo en una de las colecciones de datos, para el conjunto B2, el modelo requiere un tiempo de cálculo elevado. Se ha intentado sin éxito por el momento dar una explicación de las razones que han dado lugar a este resultado, por lo que queda pendiente un estudio más detallado de este conjunto de datos en particular. En el caso de conjuntos de datos con variables categóricas, como la colección Diamond, se ha visto que el modelo precisaba de un estudio más fino. Como se ha visto antes de exigir las restricciones, la imposición realizada sobre el número de grupos y el número de variables que seleccionar impedía recoger parte de la información acerca de las características que definen cada grupo. Hay que tener en cuenta que en este caso se desconoce el número de variables ruido dentro del conjunto de datos. Cuando se impone seleccionar un número bajo de variables, dado que el procedimiento no tiene en cuenta el resto, es posible que se pierda información relevante para clasificar correctamente los datos. Al añadir las restricciones, el método ha sido capaz de recoger una mayor cantidad de información sobre los grupos para el mismo número de variables seleccionadas. La capacidad que ofrecen los modelos de programación matemática de añadir restricciones adicionales a las del propio modelo es precisamente una de sus principales ventajas y puede ser interesante como punto de partida para otro estudio futuro.

Cabe recordar también que este modelo está diseñado para trabajar con datos con variables binarias y para conjuntos de datos con variables que se expresen en la escala de Likert. Aunque en este trabajo no se ha explorado esta otra opción, estos conjuntos de datos son también muy comunes como resultado de la realización de encuestas, por lo que podría ser de igual interés extender el estudio aquí realizado a éstos otros conjuntos de datos.

Bibliografía

[1] A. Iodice D'Enza, F. Palumbo, Iterative factor clustering of binary data, *Comput Stat* 28 (2013), 789–807.

[2] A. Markos, A.I. D'Enza, M. Van de Velden, Beyond tandem analysis: joint dimension reduction and clustering in R, *Journal of Statistical Software* 258 (10), 2019.

[3] B. S. Everitt, S. Landau, M. Leese, D. Stahl, *Cluster Analysis*, 5th ed., Wiley, 2011.

[4] CRAN R Project, *Package 'clustrd'*, `https://cran.r-project.org/web/packages/clustrd/clustrd.pdf`.

[5] D. Peña, *Análisis de datos multivariantes*, 1ª ed., Mc Graw Hill, 2002.

[6] E. Dimitriadou, S. Dolnicar, A. Weingessel, An examination of indexes for determining the number of clusters in binary data sets, *Psychometrika* 67 (3) (2002), 137–160.

[7] H. Hwang, W.R. Dillon, Y. Takane, An extension of multiple correspondence analysis for identifying heterogeneous subgroups of respondents, *Psychometrika* 71 (1) (2006), 161–171.

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

[9] M. Ehrgott, *Multicriteria Optimization*, 2nd ed., Springer, 2005.

[10] M. J. Brusco, Clustering Binary Data in the Presence of Masking Variables. *Psychological Methods* 9 (4) (2004), 510–523.

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

[12] S. Benati, S. García, J. Puerto, Mixed integer linear programming and heuristic methods for feature selection in clustering, *Journal of the Operational Research Society* 69 (9) (2018), 1379–1395.

[13] S. de la Fuente Fernández, *Análisis correpondencias simples y múltiples,* Facultad de Ciencias Económicas y Empresariales, Universitat Autónoma de Madrid, 2011.

[14] T. Hastie, R. Tibshirani, J. Friedman, *The elements of statistical learning. Data mining, inference and prediction,* 2 nd ed., Springer Series in Statistics, 2009.

[15] T. Malón Melendo, *Manual de uso IBM ILOG CPLEX,* Departamento de Métodos estadísticos, Facultad de Ciencias, Universidad de Zaragoza, 2019-2020.

[16] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, *Data Mining and Knowledge Discovery* 2 (1998), 283–304.

Apéndice A

Implementación de los modelos de optimización en CPLEX

A continuación, se puede ver el código utilizado para implementar las dos formulaciones del modelo de optimización (2.1). El código que se muestra corresponde a los archivos con extensión *.mod* generados en CPLEX. Además, para introducir los datos y los valores concretos de los parámetros se generaron archivos adicionales con extensión *.dat*, que no hemos creído relevante incluir aquí, dada su extensión.

Primera formulación: Linealización directa (ACSV-w) (2.2)

[ ]:
```
//defino los parámetros del problema
int p=...;//numero de variables
int n=...;//numero de observaciones
range obs=1..n;
range variable=1..p;
int r=...;//dimension reducida
int q=...;//numero de cluster
float v[obs][variable]=...;
int dv[obs][obs][variable];
int dva[obs][obs];
int ddd;

execute initializeArray{
        for(var i in obs){
                for(var j in obs){
                        ddd=0;
                        for(var k in variable){
                                        dv[i][j][k]=Opl.abs(v[i][k]-v[j][k]);
                                        ddd=ddd+dv[i][j][k];
                        }
                        dva[i][j]=ddd;
                }
        }
//-------------------- Añadimos límitede tiempo
    cplex.epgap = 0.001;
    cplex.tilim =3600;
    //cplex.LPmethod = 0; //automatic: let CPLEX choose
    //cplex.epagap = 1;
```

```
    //cplex.epagap = 10;
    //--
}

//defino las variables de decisión
dvar boolean z[variable];//1 si la variable se elige
dvar boolean y[obs];//1 si el caso es centroide
dvar float+ x[obs][obs];//1 si la observacion está en el grupo definido por la obs
dvar float+ w[obs][obs][variable];//

//defino la función objetivo del problema
minimize sum(i in obs, j in obs, k in variable) w[i][j][k]*dv[i][j][k];
//defino las restricciones del problema
subject to {
//Se eligen q individuos como centroides
sum(j in obs) y[j]==q;
//Un individuo solo se puede asignar a un centroide
forall (i in obs,j in obs)
    x[i][j]<=y[j];
//Un individuo solo se asigna a un centroide
forall (i in obs)
sum(j in obs) x[i][j]==1;
//Se eligen exactamente r variables
sum(k in variable) z[k]==r;
//Definicion de w_ijk
forall(i in obs, j in obs,k in variable){
w[i][j][k]>=x[i][j]+z[k]-1;
w[i][j][k]<=x[i][j];
 w[i][j][k]<=z[k];
}
}
main {
        var f = new IloOplOutputFile("Resultado.txt");
        thisOplModel.generate();

           var time0 = new Date();
        var OK = cplex.solve();
        var time1 = new Date();
        var tTotal = (time1.getTime()-time0.getTime())/1000;

        if (OK) {
                f.writeln("variables elegidas");
                for(var k in thisOplModel.variable){
                        if(thisOplModel.z[k] != 0){
                                f.writeln("variable: ",k);
                        }
                }
                for(var i in thisOplModel.obs){
                        if(thisOplModel.y[i] != 0){
                                var totclus  = 0;
                                f.writeln("centroides");
```

```
                                f.writeln("individuo: ",i);
                                f.writeln("elementos del cluster"); }
                        for(var j in thisOplModel.obs){
                                if(thisOplModel.x[j][i] != 0){
                                        totclus=totclus +1;
                                    f.writeln("individuo: ",j);
                                                }}
                                if(thisOplModel.y[i] != 0){
          f.writeln("número total de elementos del cluster: ", totclus); }
                                }

        f.writeln("Tiempo: ",tTotal,"segundos. Objetivo: "+cplex.getObjValue( ));
        f.close();
}
}
```

Segunda formulación: Formulación del radio simplificado (ACSV-rb) (2.4)

[ ]:
```
//defino los parámetros del problema
int p=...;//numero de variables
int n=...;//numero de observaciones
range obs=1..n;
range variable=1..p;
int r=...;//dimension reducida
int q=...;//numero de cluster
//float v[1..308][1..17];
float v[obs][variable]=...;
int dv[obs][obs][variable];
int dva[obs][obs];
int ddd;

execute initializeArray{
        for(var i in obs){
                for(var j in obs){
                        ddd=0;
                        for(var k in variable){
                                dv[i][j][k]=Opl.abs(v[i][k]-v[j][k]);
                                ddd=ddd+dv[i][j][k];
                        }
                        dva[i][j]=ddd;
                }
        }

//-------------------- Añadimos límitede tiempo
    cplex.epgap = 0.001;
    cplex.tilim =3600;
    //cplex.LPmethod = 0; //automatic: let CPLEX choose
    //cplex.epagap = 1;
    //cplex.epagap = 10;
    //----------------------------------------
}
```

```
//defino las variables de decisión
dvar boolean z[variable];//1 si la variable se elige
dvar boolean y[obs];//1 si el caso es centroide
dvar float+ x[obs][obs];//1 si la observacion está en el grupo definido por la obs
dvar float+ radio[obs][variable];//radio[i][k] toma el valor 1 cuando la variable
// k es eleigida y la observación i se asigna a un grupo cuyo centro está a
//  distancia igual a 1



//defino la función objetivo del problema
minimize sum(i in obs, k in variable) radio[i][k];
//defino las restricciones del problema
subject to {
//Se eligen q individuos como centroides
sum(j in obs) y[j]==q;
//Un individuo solo se puede asignar a un centroide
forall (i in obs,j in obs)
    x[i][j]<=y[j];
//Un individuo solo se asigna a un centroide
forall (i in obs)
        sum(j in obs) x[i][j]==1;
//Se eligen exactamente r variables
sum(k in variable) z[k]==r;
//Definicion del radio
forall(i in obs, k in variable)
              radio[i][k]+sum(j in obs: dv[i][j][k]<1) x[i][j]>=z[k];
}
main {
        var f = new IloOplOutputFile("Resultado.txt");
        thisOplModel.generate();

           var time0 = new Date();
        var OK = cplex.solve();
        var time1 = new Date();

        var tTotal = (time1.getTime()-time0.getTime())/1000;


        if (OK) {
                f.writeln("variables elegidas");
                for(var k in thisOplModel.variable){
                        if(thisOplModel.z[k] != 0){
                                f.writeln("variable: ",k);
                        }

                }
                for(var i in thisOplModel.obs){
                        if(thisOplModel.y[i] != 0){
                                var totclus  = 0;
                                f.writeln("centroides");
```

```
                                f.writeln("individuo: ",i);
                                f.writeln("elementos del cluster"); }
                        for(var j in thisOplModel.obs){
                                if(thisOplModel.x[j][i] != 0){
                                        totclus=totclus +1;
                                    f.writeln("individuo: ",j);
                                                }}
                                if(thisOplModel.y[i] != 0){
     f.writeln("número total de elementos del cluster: ", totclus); }

                }
        f.writeln("Tiempo: ",tTotal,"segundos. Objetivo: "+cplex.getObjValue( ));
        f.close();
}
}
```

Modificación del modelo ACSV-rb para añadirle restricciones sobre el número máximo de categorías para cada variable que puede escoger

Se muestra sólo el apartado de restricciones, puesto que el resto del código coincide con el anterior.

[ ]:
```
//defino las restricciones del problema
subject to {
//Se eligen q individuos como centroides
sum(j in obs) y[j]==q;
//Un individuo solo se puede asignar a un centroide
forall (i in obs,j in obs)
    x[i][j]<=y[j];
//Un individuo solo se asigna a un centroide
forall (i in obs)
        sum(j in obs) x[i][j]==1;
//Restricciones sobre el número máximo de categorías de cada variable
sum(k in 1..3)z[k]<=2;
sum(k in 4..9)z[k]<=4;
sum(k in 10..14)z[k]<=3;
sum(k in 15..17)z[k]<=2;
//Se eligen exactamente r variables
sum(k in variable) z[k]==r;
//Definicion del radio
forall(i in obs, k in variable)
              radio[i][k]+sum(j in obs: dv[i][j][k]<1) x[i][j]>=z[k];
}
```

Apéndice B

Resultados sobre los conjuntos Diamond en R

Se presenta aquí la salida de R extendida para uno de los tres conjuntos de diamantes estudiado.

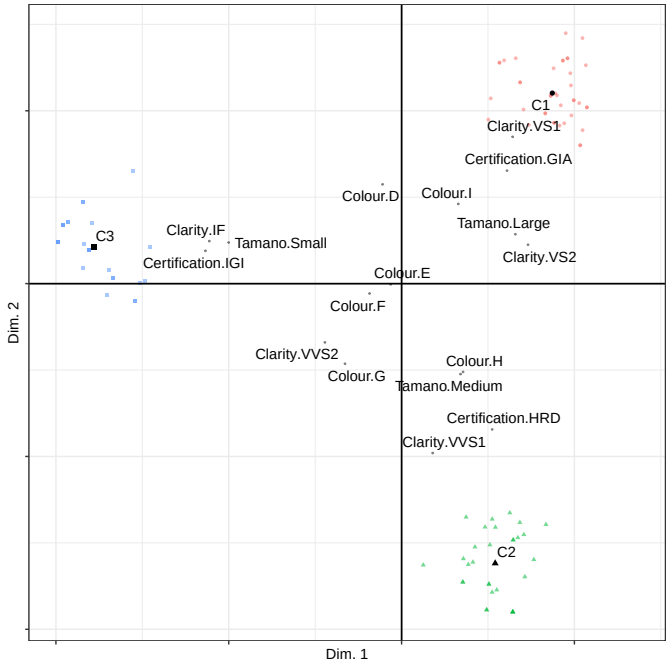
Notemos que se han descartado parte de los resultados de la salida, por carecer de relevancia en nuestro estudio para este caso concreto.

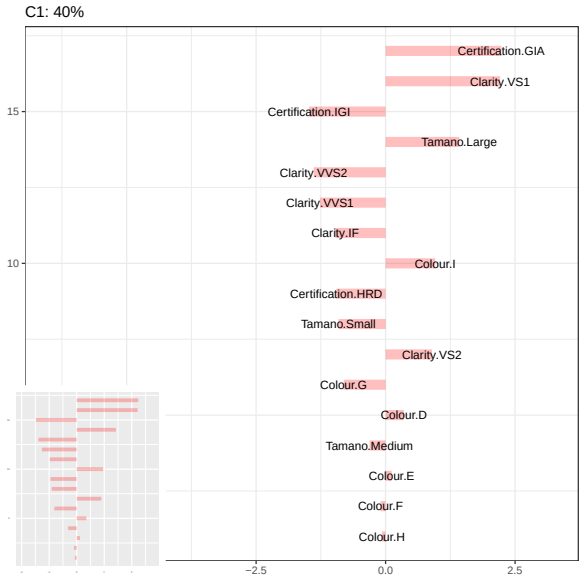
```
bD1 <- tuneclus(D1, 2:6 , 1:3, method = "MCAk", criterion = "asw", dst = "full",
 nstart = 10, seed = 1234)

bD1

## The best solution was obtained for 3 clusters of sizes
40 (40%), 31 (31%), 29 (29%) in 2 dimensions,
 for an average Silhouette width value of 0.317.
##
## Cluster quality criterion values across the specified range of
 clusters (rows) and dimensions (columns):
##      X1    X2    X3
## 2 0.307
## 3  0.19 0.317
## 4 0.156 0.238 0.275
## 5  0.13 0.182 0.252
## 6 0.061 0.143 0.197
```

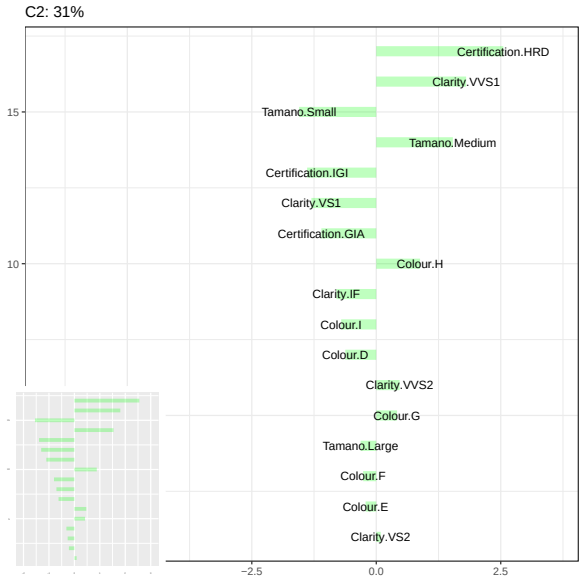
```
MCAkD1 <- clusmca(D1, 3, 2, method='MCAk')

summary(MCAkD1)

## Solution with 3 clusters of sizes
40 (40%), 31 (31%), 29 (29%) in 2 dimensions.
##
## Cluster centroids:
##             Dim.1   Dim.2
## Cluster 1  0.0872  0.1102
## Cluster 2  0.0542 -0.1618
## Cluster 3 -0.1782  0.0209
##
## Variable scores:
##                     Dim.1   Dim.2
## Colour.D          -0.0110  0.0574
## Colour.E          -0.0063 -0.0004
## Colour.F          -0.0185 -0.0057
```

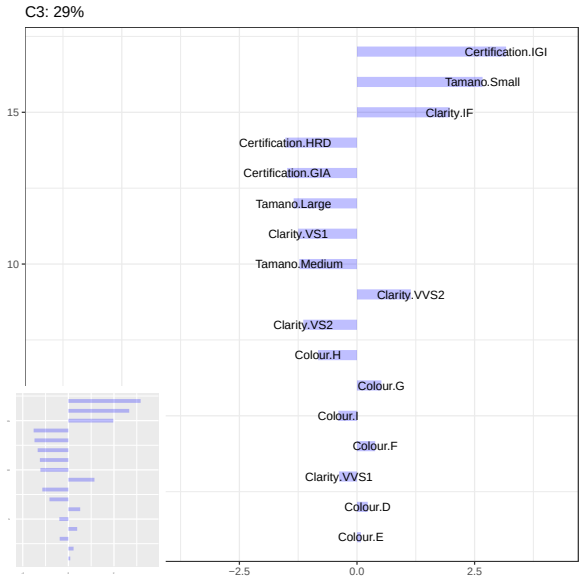
```
## Colour.G          -0.0327 -0.0463
## Colour.H           0.0356 -0.0511
## Colour.I           0.0329  0.0461
## Clarity.IF        -0.1112  0.0246
## Clarity.VS1        0.0643  0.0849
## Clarity.VS2        0.0732  0.0225
## Clarity.VVS1       0.0180 -0.0980
## Clarity.VVS2      -0.0443 -0.0340
## Certification.GIA  0.0610  0.0654
## Certification.HRD  0.0524 -0.0844
## Certification.IGI -0.1135  0.0190
## Tamano.Large       0.0659  0.0286
## Tamano.Medium      0.0342 -0.0524
## Tamano.Small      -0.1000  0.0238
##
## Within cluster sum of squares by cluster:
## [1] 0.0212 0.0177 0.0169
##  (between_SS / total_SS =   97.92 %)

## Objective criterion value: 2.2336
```

```
plot(MCAkD1, cludesc = TRUE, topstdres = 20, subplot = TRUE)
```




C1: 40%
Certification.GIA
Clarity.VS1
Certification.IGI
Tamano.Large
Clarity.VVS2
Clarity.VVS1
Clarity.IF
Colour.I
Certification.HRD
Tamano.Small
Clarity.VS2
Colour.G
Colour.D
Tamano.Medium
Colour.E
Colour.F
Colour.H
15
10
−2.5
0.0
2.5


C2: 31%
Certification.HRD
Clarity.VVS1
Tamano.Small
Tamano.Medium
Certification.IGI
Clarity.VS1
Certification.GIA
Colour.H
Clarity.IF
Colour.I
Colour.D
Clarity.VVS2
Colour.G
Tamano.Large
Colour.F
Colour.E
Clarity.VS2
15
10
−2.5
0.0
2.5


C3: 29%
Certification.IGI
Tamano.Small
Clarity.IF
Certification.HRD
Certification.GIA
Tamano.Large
Clarity.VS1
Tamano.Medium
Clarity.VVS2
Clarity.VS2
Colour.H
Colour.G
Colour.I
Colour.F
Clarity.VVS1
Colour.D
Colour.E
15
10
−2.5
0.0
2.5

Apéndice C

Resultados sobre los conjuntos de datos Brusco en R

Se muestra a continuación el código utilizado para generar los conjuntos y la salida de R completa para 3 de los 9 conjuntos de datos con los que se ha trabajado, que sirven de ejemplo para el resto. Se ha seleccionado la salida del conjunto 1, que constaba de 3 variables verdaderas, el conjunto 4, constituido por 6, y el conjunto 7, que contaba con 9 variables verdaderas.

Código para generar los conjuntos de datos

Como ejemplo, enseñamos aquí la creación del primer conjunto de datos, que contiene 3 variables verdaderas. El resto de conjuntos se han creado de manera análoga cambiando el valor de la semilla de aleatoriedad.

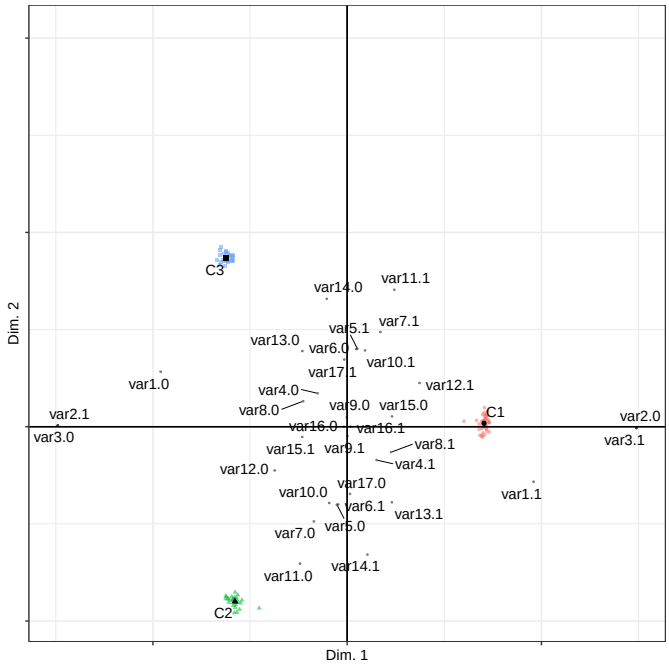
```
set.seed(1234)
var1<- c(rep(1,31),  rep(1, 46), rep(0,23))
var2<- c(rep(1,31),  rep(0, 46), rep(1,23))
var3<- c(rep(0,31),  rep(1, 46), rep(0,23))
var4<- rbinom(100, 1, 0.5)
var5<- rbinom(100, 1, 0.5)
var6<- rbinom(100, 1, 0.5)
var7<- rbinom(100, 1, 0.5)
var8<- rbinom(100, 1, 0.5)
var9<- rbinom(100, 1, 0.5)
var10<- rbinom(100, 1, 0.5)
var11<- rbinom(100, 1, 0.5)
var12<- rbinom(100, 1, 0.5)
var13<- rbinom(100, 1, 0.5)
var14<- rbinom(100, 1, 0.5)
var15<- rbinom(100, 1, 0.5)
var16<- rbinom(100, 1, 0.5)
var17<- rbinom(100, 1, 0.5)



bin31<- data.frame(var1, var2, var3, var4, var5, var6, var7, var8, var9, var10,var11,
var12, var13, var14, var15, var16, var17)
```

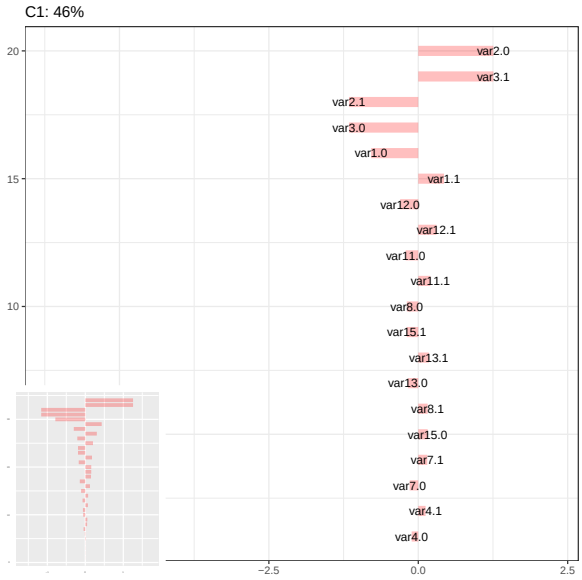
CONJUNTO 1

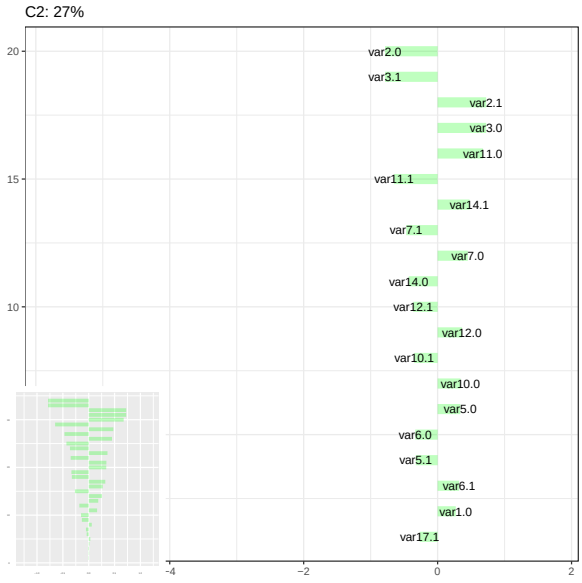
```
MCAkbin31 <- clusmca(bin31, 3, 2, method='MCAk')

summary(MCAkbin31)

## Solution with 3 clusters of sizes 46 (46%), 27 (27%), 27 (27%) in 2 dimensions.
##
```

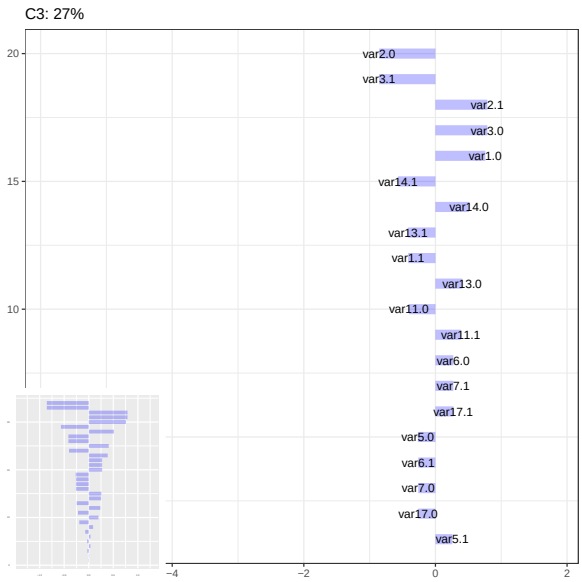
```
## Cluster centroids:
##             Dim.1   Dim.2
## Cluster 1  0.0705  0.0018
## Cluster 2 -0.0578 -0.0900
## Cluster 3 -0.0622  0.0870
##
## Variable scores:
##           Dim.1   Dim.2
## var1.0  -0.0960  0.0283
## var1.1   0.0960 -0.0283
## var2.0   0.1489 -0.0007
## var2.1  -0.1489  0.0007
## var3.0  -0.1489  0.0007
## var3.1   0.1489 -0.0007
## var4.0  -0.0152  0.0172
## var4.1   0.0152 -0.0172
## var5.0  -0.0053 -0.0402
## var5.1   0.0053  0.0402
## var6.0   0.0046  0.0400
## var6.1  -0.0046 -0.0400
## var7.0  -0.0171 -0.0487
## var7.1   0.0171  0.0487
## var8.0  -0.0226  0.0131
## var8.1   0.0226 -0.0131
## var9.0  -0.0001  0.0048
## var9.1   0.0001 -0.0048
## var10.0 -0.0092 -0.0393
## var10.1  0.0092  0.0393
## var11.0 -0.0243 -0.0705
## var11.1  0.0243  0.0705
## var12.0 -0.0373 -0.0225
## var12.1  0.0373  0.0225
## var13.0 -0.0230  0.0389
## var13.1  0.0230 -0.0389
## var14.0 -0.0105  0.0658
## var14.1  0.0105 -0.0658
## var15.0  0.0231  0.0053
## var15.1 -0.0231 -0.0053
## var16.0 -0.0017  0.0000
## var16.1  0.0017  0.0000
## var17.0  0.0015 -0.0346
## var17.1 -0.0015  0.0346
##
## Within cluster sum of squares by cluster:
## [1] 8e-04 5e-04 3e-04
##  (between_SS / total_SS =   99.82 %)
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   2   3   1   2   2   2   2   3   2   2   3   3   3   2   2   2   2   2   3   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   3   3   3   3   2   3   2   2   2   3   3   1   1   1   1   1   1   1   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   3   2   3
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   2   2   3   3   3   2   3   2   3   3   3   3   2   3   3   3   2   3
##
## Objective criterion value: 15.401
```

```
plot(MCAkbin31, cludesc = TRUE, topstdres = 20, subplot = TRUE)
```




C1: 46%
var2.0
var3.1
var2.1
var3.0
var1.0
var1.1
var12.0
var12.1
var11.0
var11.1
var8.0
var15.1
var13.1
var13.0
var8.1
var15.0
var7.1
var7.0
var4.1
var4.0
20
15
10
−2.5
0.0
2.5


C2: 27%
var2.0
var3.1
var2.1
var3.0
var11.0
var11.1
var14.1
var7.1
var7.0
var14.0
var12.1
var12.0
var10.1
var10.0
var5.0
var6.0
var5.1
var6.1
var1.0
var17.1
20
15
10
−4
−2
0
2

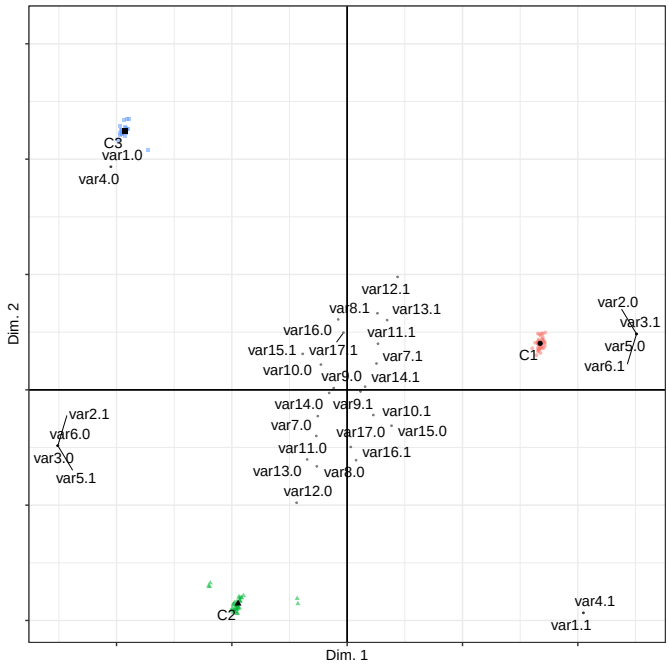

C3: 27%
var2.0
var3.1
var2.1
var3.0
var1.0
var14.1
var14.0
var13.1
var1.1
var13.0
var11.0
var11.1
var6.0
var7.1
var17.1
var5.0
var6.1
var7.0
var17.0
var5.1
20
15
10
−4
−2
0
2

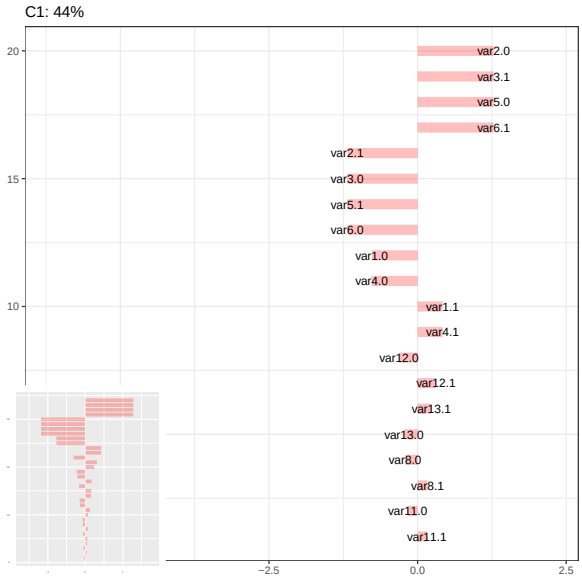
CONJUNTO 4

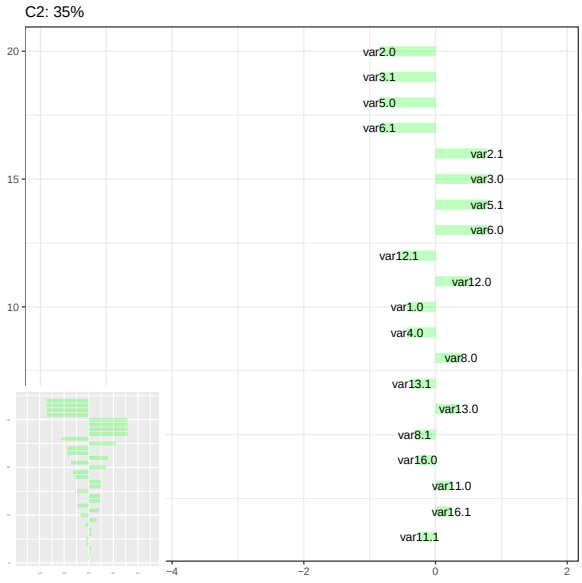
```
MCAkbin61 <- clusmca(bin61, 3, 2, method='MCAk')

summary(MCAkbin61)

## Solution with 3 clusters of sizes 44 (44%), 35 (35%), 21 (21%) in 2 dimensions.
##
## Cluster centroids:
##             Dim.1   Dim.2
## Cluster 1  0.0837  0.0201
## Cluster 2 -0.0473 -0.0927
## Cluster 3 -0.0964  0.1123
##
## Variable scores:
##           Dim.1   Dim.2
## var1.0  -0.1024  0.0967
## var1.1   0.1024 -0.0967
## var2.0   0.1254  0.0242
## var2.1  -0.1254 -0.0242
## var3.0  -0.1254 -0.0242
## var3.1   0.1254  0.0242
## var4.0  -0.1024  0.0967
## var4.1   0.1024 -0.0967
## var5.0   0.1254  0.0242
## var5.1  -0.1254 -0.0242
## var6.0  -0.1254 -0.0242
## var6.1   0.1254  0.0242
## var7.0  -0.0127 -0.0114
## var7.1   0.0127  0.0114
## var8.0  -0.0131 -0.0332
## var8.1   0.0131  0.0332
## var9.0  -0.0058  0.0008
## var9.1   0.0058 -0.0008
## var10.0 -0.0114  0.0109
## var10.1  0.0114 -0.0109
## var11.0 -0.0134 -0.0200
## var11.1  0.0134  0.0200
## var12.0 -0.0219 -0.0489
## var12.1  0.0219  0.0489
## var13.0 -0.0173 -0.0302
## var13.1  0.0173  0.0302
## var14.0 -0.0078 -0.0014
## var14.1  0.0078  0.0014
## var15.0  0.0193 -0.0156
## var15.1 -0.0193  0.0156
## var16.0 -0.0039  0.0305
## var16.1  0.0039 -0.0305
## var17.0  0.0016 -0.0248
## var17.1 -0.0016  0.0248
##
## Within cluster sum of squares by cluster:
## [1] 0.0003 0.0022 0.0003
##  (between_SS / total_SS =   99.76 %)
##
##
##
##
##
##
##
##
##
```

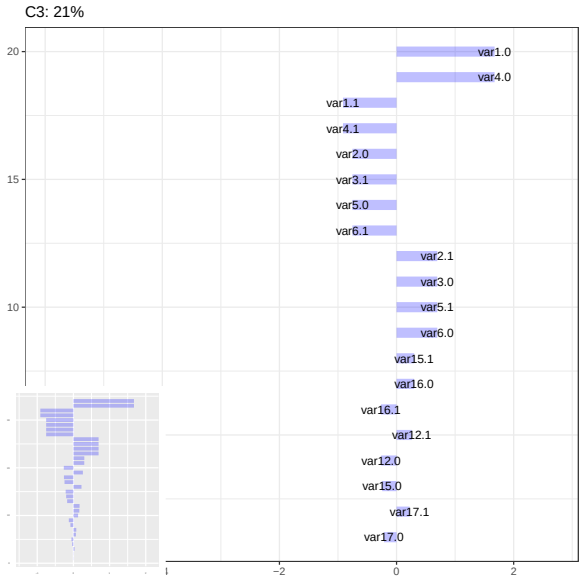
```
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   2   2   2   2   2   2   2   2   2   2   3   2   2   2   2   2   2   2   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   2   2   2   2   2   2   2   2   1   1   1   2   1   1   1   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   3   3
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   2   3   3   3   3
##
## Objective criterion value: 14.0351
```

```
plot(MCAkbin61, cludesc = TRUE, topstdres = 20, subplot = TRUE)
```




C1: 44%
20
15
10
var2.0
var3.1
var5.0
var6.1
var2.1
var3.0
var5.1
var6.0
var1.0
var4.0
var1.1
var4.1
var12.0
var12.1
var13.1
var13.0
var8.0
var8.1
var11.0
var11.1
−2.5
0.0
2.5


C2: 35%
20
15
10
var2.0
var3.1
var5.0
var6.1
var2.1
var3.0
var5.1
var6.0
var12.1
var12.0
var1.0
var4.0
var8.0
var13.1
var13.0
var8.1
var16.0
var11.0
var16.1
var11.1
−4
−2
0
2

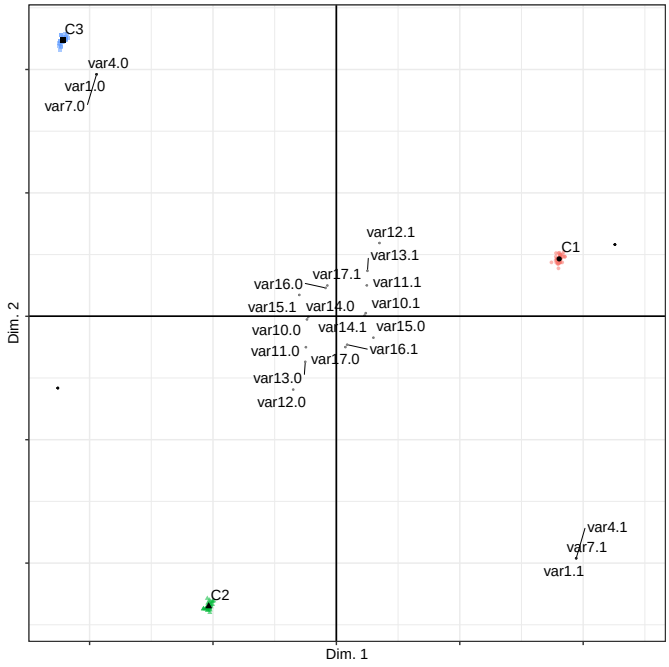

C3: 21%
20
15
10
var1.0
var4.0
var1.1
var4.1
var2.0
var3.1
var5.0
var6.1
var2.1
var3.0
var5.1
var6.0
var15.1
var16.0
var16.1
var12.1
var12.0
var15.0
var17.1
var17.0
−2
0
2

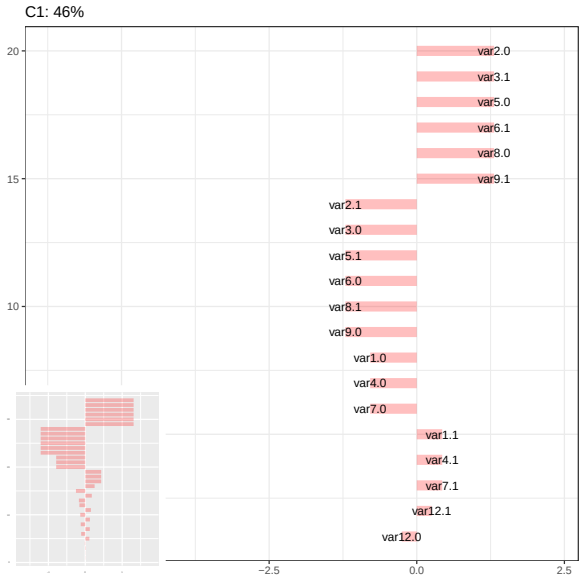
CONJUNTO 7

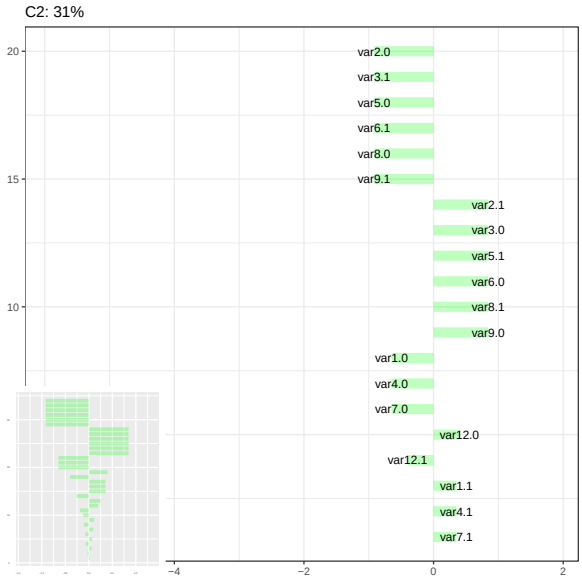
```
MCAkbin91 <- clusmca(bin91, 3, 2, method='MCAk')

summary(MCAkbin91)

## Solution with 3 clusters of sizes 46 (46%), 31 (31%), 23 (23%) in 2 dimensions.
##
## Cluster centroids:
##             Dim.1   Dim.2
## Cluster 1  0.0903  0.0233
## Cluster 2 -0.0517 -0.1175
## Cluster 3 -0.1108  0.1118
##
## Variable scores:
##           Dim.1   Dim.2
## var1.0  -0.0972  0.0980
## var1.1   0.0972 -0.0980
## var2.0   0.1129  0.0291
## var2.1  -0.1129 -0.0291
## var3.0  -0.1129 -0.0291
## var3.1   0.1129  0.0291
## var4.0  -0.0972  0.0980
## var4.1   0.0972 -0.0980
## var5.0   0.1129  0.0291
## var5.1  -0.1129 -0.0291
## var6.0  -0.1129 -0.0291
## var6.1   0.1129  0.0291
## var7.0  -0.0972  0.0980
## var7.1   0.0972 -0.0980
## var8.0   0.1129  0.0291
## var8.1  -0.1129 -0.0291
## var9.0  -0.1129 -0.0291
## var9.1   0.1129  0.0291
## var10.0 -0.0119 -0.0013
## var10.1  0.0119  0.0013
## var11.0 -0.0123 -0.0125
## var11.1  0.0123  0.0125
## var12.0 -0.0175 -0.0297
## var12.1  0.0175  0.0297
## var13.0 -0.0126 -0.0184
## var13.1  0.0126  0.0184
## var14.0 -0.0114 -0.0003
## var14.1  0.0114  0.0003
## var15.0  0.0150 -0.0087
## var15.1 -0.0150  0.0087
## var16.0 -0.0044  0.0115
## var16.1  0.0044 -0.0115
## var17.0  0.0037 -0.0125
## var17.1 -0.0037  0.0125
##
## Within cluster sum of squares by cluster:
## [1] 1e-04 1e-04 1e-04
##  (between_SS / total_SS =   99.98 %)
##
##
##
##
##
##
##
##
##
```

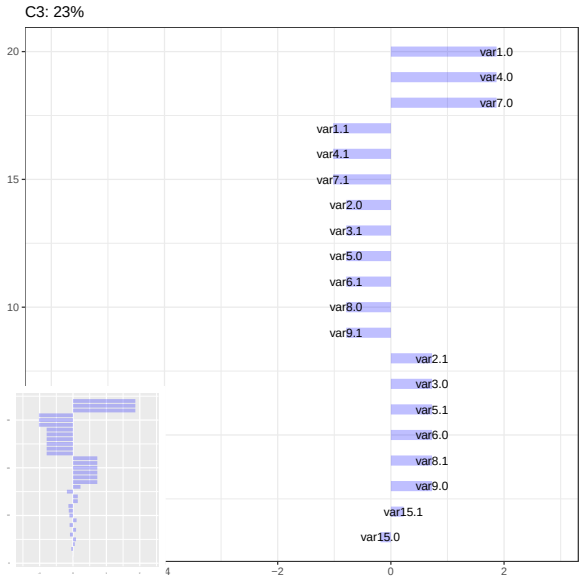
```
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   2   2   2   2   2   2   2   2   1   1   1   1   1   1   1   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   3   3   3
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##
## Objective criterion value: 12.3946
```

```
plot(MCAkbin91, cludesc = TRUE, topstdres = 20, subplot = TRUE)
```




C1: 46%
var2.0
var3.1
var5.0
var6.1
var8.0
var9.1
var2.1
var3.0
var5.1
var6.0
var8.1
var9.0
var1.0
var4.0
var7.0
var1.1
var4.1
var7.1
var12.1
var12.0
20
15
10
−2.5
0.0
2.5


C2: 31%
var2.0
var3.1
var5.0
var6.1
var8.0
var9.1
var2.1
var3.0
var5.1
var6.0
var8.1
var9.0
var1.0
var4.0
var7.0
var12.0
var12.1
var1.1
var4.1
var7.1
20
15
10
−4
−2
0
2


C3: 23%
var1.0
var4.0
var7.0
var1.1
var4.1
var7.1
var2.0
var3.1
var5.0
var6.1
var8.0
var9.1
var2.1
var3.0
var5.1
var6.0
var8.1
var9.0
var15.1
var15.0
20
15
10
4
−2
0
2

Apéndice D

Modificaciones realizadas a la función clusmca() y MCAk() en R

La función que se deseaba modificar para obtener el conjunto de soluciones de Pareto es *clusmca()*. Esta función, cuando se aplica el método MCA k-medias, llama a otra función, *MCAk()*. Con el fin de obtener los resultados deseados, se han tenido que modificar ambas, que han sido renombradas *clusmcaIrene()* y *MCAkIrene()*. Como las modificaciones en el código eran pocas, se ha decidido mostrar aquí sólo partes seleccionadas del código, donde tenían lugar dichas modificaciones. El código original en su totalidad puede verse en `https://rdrr.io/cran/clustrd/src/R/clusmca.R` (para *clusmca()*) y `https://rdrr.io/cran/clustrd/src/R/MCAk.r` (para la función *MCAk()*).

Los cambios efectuados en el código original aparecen marcados en color rojo.

```
clusmcaIrene  <- function(data,nclus,ndim,method=c("clusCA","iFCB","MCAkIrene"),
alphak = .5,nstart=100,smartStart=NULL,gamma = TRUE,inboot=FALSE,seed=NULL)

 {   ### A single cluster gives the MCA solution
  if (nclus == 1) { ...
  } else {
     ...


    method <- match.arg(method, c("clusCA", "clusca","CLUSCA","CLUSca",
 "ifcb","iFCB","IFCB","mcakirene","MCAkIrene", "MCAKIRENE","mcaKirene"), several.ok = T)[1]
    method <- tolower(method)

  ...

    if(method=="mcakirene"){
        out=MCAkIrene(data=data,nclus=nclus,ndim=ndim,nstart=nstart,alphak = alphak,
smartStart=smartStart, gamma = gamma,seed=seed)

    }
    return(out)
  }
}
```

```
MCAkIrene <- function(data, nclus = 3, ndim = 2, alphak = .5, nstart = 100,
         smartStart=NULL,gamma = TRUE, seed=NULL, inboot = FALSE)

{
  ...

  if (alphak == 1) { #Tandem approach MCA + k-means
  ...
  } else {

   ...

      itmax = 100
      it = 0
      ceps = 1e-04
      imp = 1e+05
      f0 = 1e+05

      while((it <= itmax ) && ( imp > ceps ) ){
        it=it+1
        ########################################################################
        ## STEP 1: update of U #################################################
        ########################################################################
        #use Lloyd's k-means algorithm to get the results of Hwang and Takane (2006)
        #outK=try(kmeans(Fm,centers=center,algorithm="Lloyd",nstart=100),silent=T)
        outK = try(kmeans(Fm,centers=center,nstart=100),silent=T)

        if(is.list(outK)==F){
          outK = EmptyKmeans(Fm,centers=center)
          #  break
        }
        center=outK$centers
        index = outK$cluster
        U = tab.disjonctif(index)
        U0 = scale(U,center=TRUE, scale=FALSE)
        uu = colSums(crossprod(U)) #colSums(t(U)%*% U)
        invsqDru = 1/sqrt(c(rr,uu))

        ########################################################################
        ## STEP 2: update of Fm and Wj #########################################
        ########################################################################
        MU = cbind(alphak*M,(1-alphak)*U0)
        Pzu = t(t(MU) * as.vector(invsqDru))

        Pzu[is.nan(Pzu)] <- 0
        PPzu = t(Pzu) %*% Pzu
        svdPzu = svd(PPzu)
        #invsqD = diag(1/sqrt(svdPzu$d))
```

```
  Fm = t(t(Pzu %*% svdPzu$u[,1:ndim]) * as.vector(1/sqrt(svdPzu$d[1:ndim])))
  ft1 = 0
  k = 1
  kk = 0
  kk = kk+zncati[1]
  Tm = z[,k:kk]
  #chol2inv(chol(crossprod(Tm))) gives an error!
  W = pseudoInverse(crossprod(Tm))%*% t(Tm) %*% Fm
  A = W
  ft1 = ft1+sum(diag((crossprod(Fm))-(t(Fm) %*% Tm %*% W)))
  k = kk+1
  for(j in 2:zitem){
    kk = kk+zncati[j]
    Tm = z[,k:kk]
    W = pseudoInverse(crossprod(Tm))%*% t(Tm) %*% Fm
    A = rbind(A,W)
    ft1 = ft1 + sum(diag((crossprod(Fm))-(t(Fm) %*% Tm %*% W))) ## MCA
    k=kk+1
  }
  ft2 = sum(diag((crossprod(Fm)) - (t(Fm) %*% U %*% center)))

  #check again
  f =  alphak*ft1 + (1-alphak)*ft2


  imp=f0-f
  f0=f
  Fv = cbind(Fv,f)
} # end WHILE

if (f < oldf){
  #####gamma scaling
  if (gamma == TRUE) {
    distB = sum(diag(crossprod(A)))
    distG = sum(diag(crossprod(center)))
    g = ((nclus/zitem)* distB/distG)^.25

    A = (1/g)*A
    center = g*center #is this needed
    Fm = g*Fm
  }
  ##########################
  oldF = Fm
  oldindex = index
  oldf = f

  oldft1 = ft1
  oldft2 = ft2

  Uold = U
  Aold = A
```

```
        centerold = center
      }
    } ##end of FOR


    Fm = oldF
    index = oldindex
    cluster = as.numeric(index)
    f = oldf

    ft1 = oldft1
    ft2 = oldft2

    U = Uold
    A = Aold
    # it=itold
    center = centerold

    size = table(cluster)
    aa = sort(size,decreasing = TRUE)

    cluster = mapvalues(cluster, from = as.integer(names(aa)),
 to = as.integer(names(table(cluster))))
    #reorder centroids
    center = center[as.integer(names(aa)),]
    setTxtProgressBar(pb, 1)

    out$obscoord = Fm # observations coordinates
    out$attcoord = A # attributes coordinates
    rownames(out$obscoord) = rownames(data)
    rownames(out$attcoord) = paste(lab1,lab2,sep=".")
    out$centroid = center # centroids
    cluster = as.integer(cluster)
    names(cluster) = rownames(data)
    out$cluster = cluster #as.numeric(index) # cluster membership

    out$criterion = c(f, ft1, ft2) # criterion

    out$size = as.integer(aa)  #round((table(cluster)/sum( table(cluster)))*100,digits=1)
    out$odata = data.frame(lapply(data.frame(data),factor),stringsAsFactors = TRUE)
    out$nstart = nstart
    class(out) = "clusmca"
    return(out)
  }
}
```