UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales

Escalado multidimensional métrico en problemas con grandes datos

Tesis presentada para optar al título de Magister de la Universidad de Buenos Aires en Estadística Matemática

Ing. Pedro Camilo Cosatto Ammann

Directora de tesis: Dra. Daniela Andrea Rodríguez

Fecha de Defensa: 31 de Julio de 2023.

Resumen

En este trabajo describimos y aplicamos métodos de escalado multidimensional (MDS) para muestras con gran cantidad de datos. El escalado multidimensional es un conjunto de técnicas de representación de objetos basadas en las distancias, similaridades o disimilaridades entre ellos. Estos métodos tienen severas limitaciones cuando el tamaño de la muestra aumenta, debido a las dificultades de cómputo. Analizamos tres algoritmos distintos para sortear este problema: dos de ellos basados en la idea de *división y conquista*, y uno de ellos basado en un método de interpolación. Luego, aplicamos uno de ellos a un problema de agrupamiento. Los métodos estudiados logran reproducir con gran exactitud y precisión la solución que se obtendría con los métodos clásicos, aunque se descubrieron algunos aspectos a mejorar, especialmente con la aparición de datos atípicos. Por lo realizado en el problema de aplicación, creemos que estas variantes aportan ventajas al MDS como método de reducción de la dimensión, poniéndolo al mismo nivel que otras técnicas comúnmente usadas en el tratamiento de muestras grandes, como el análisis de Componentes Principales o t-SNE.

Palabras clave: escalado multidimensional, escalado clásico, grandes datos, reducción de la dimensión, transformaciones de Procrustes, interpolación, aprendizaje no supervisado

Abstract

Multidimensional scaling in big data

In this work, we describe and apply multidimensional scaling (MDS) methods with large samples. Multidimensional scaling is a set of object representation techniques based on distances, similarities, or dissimilarities between them. These methods have severe limitations when the sample size increases, due to computational difficulties. We analyze three different algorithms to overcome this problem: two of them based on the idea of 'divide and conquer', and one of them based on an interpolation method. Then, we apply one of them to a clustering problem. The studied methods accurately and precisely reproduce the solution that would be obtained with classic methods, although some aspects to improve were discovered, especially with the appearance of outliers. Based on the application problem, we believe that these variants provide advantages to MDS as a dimension reduction method, putting it at the same level as other techniques commonly used in the treatment of large samples, such as Principal Component Analysis or t-SNE.

Keywords: Multidimensional scaling, classical scaling, big data, dimensionality reduction, Procrustes transformations, interpolation, unsupervised learning

Índice general

Capítulo 1

Escalado multidimensional

1.1. Introducción

El escalado multidimensional, o escalamiento multidimensional (MDS - *Multidimensional Scaling*) es un conjunto de métodos que buscan encontrar coordenadas en un espacio vectorial para objetos o individuos únicamente a partir de datos de proximidad o lejanía entre ellos. Estos métodos surgieron en el campo de la psicometría a comienzos del siglo XX, cuando se buscaba dar una ubicación espacial a las personas que respondían pruebas de comportamiento. Con una visualización en dos o tres dimensiones era posible entender con mayor claridad los resultados y corroborar hipótesis experimentales. Hoy en día el MDS cubre una amplia variedad de problemas prácticos, entre los que se encuentran:

- Mapeo de percepciones para exploración de mercado, en los que un grupo de consumidores realiza un juicio de similitud entre distintos productos. El objetivo es mapear a los consumidores y a los productos para identificar grupos y relacionarlos.
- Ordenamiento de especies en biología y ecología según distintas maneras de medir las similitudes entre ellas. Por ejemplo, para visualizar la proximidad entre distintos microorganismos como virus o bacterias.
- Mapeo de genomas.
- Mapeo de similitudes entre eventos meteorológicos catastróficos como terremotos, inundaciones e incendios forestales, para poder entender asociaciones o rasgos comunes entre ellos.
- Mapeo de las correlaciones entre distintas series de tiempo, especialmente climáticas.
- Ubicación de objetos en el espacio para poder localizarlos o rastrearlos, solo con algunas mediciones simples de proximidad con los objetos ya presentes.
- Representación simplificada de redes y grafos. Por ejemplo: redes de comunicaciones, circuitos eléctricos y cableados, redes sociales, gráficos de interacción de proteínas, cadenas de Markov, entre otros.
- Reducción de la dimensión de muestras que tienen muchas variables. Esto puede ser usado sólo con fines descriptivos, o como parte de técnicas de aprendizaje estadístico. Si bien al reducir la cantidad de variables a un número menor se pierde información, en muchos casos se simplifica el cómputo o la calidad de los resultados.

La variedad de métodos de MDS se distinguen según aspectos como el tipo de geometría en el que se realiza la representación, la forma de encontrar una solución, el modelado de los errores a los que pueden estar sujetas las mediciones, entre otras. En este capítulo presentaremos los conceptos básicos del escalado, los métodos más comunes de obtención de la solución y mostraremos algunos ejemplos a modo de introducción. Más ejemplos de aplicaciones se pueden encontrar en [8].

1.1.1. Medidas de proximidad

En los problemas multivariados típicamente se parte de una matriz $\mathbf{X}_{n\times p}$ que tiene a los n individuos de la muestra puestos en filas y las p variables en columnas, digamos una *configuración* de puntos en $\mathbb{R}^p$. En el problema de MDS, en cambio, la información de partida es una medida de proximidad o lejanía entre objetos, habitualmente en forma de matriz de $n \times n$, y se busca *producir* una configuración del tipo de $\mathbf{X}$. La información de proximidad puede obtenerse de diversas formas.

Disimilaridades y distancias

Dado un conjunto de objetos $\{O_1, O_2, ..., O_n\}$, llamaremos *disimilaridad* δ_{ij} de O_i respecto de O_j a cualquier número real que cumpla:

(a) $\delta_{ij} \geq 0$ (no negatividad),
(b) $\delta_{ij} = \delta_{ji}$ (simetría),
(c) $\delta_{ii} = 0$ para $i = 1, 2, ..., n$ (identificación).

Mientras mayor sea la disimilaridad δ_{ij} significa que *menos se parecen* entre sí los objetos O_i y O_j. Si los pares de objetos tienen asociadas algunas mediciones variables, digamos vectores $\mathbf{x}_i$ y $\mathbf{x}_j$ en $\mathbb{R}^p$, denominaremos *distancia* $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ de O_i a O_j a una función de esas variables que permita obtener una disimilaridad entre ellos. Si una distancia cumple además las siguientes propiedades la denominaremos *distancia métrica*:

(d) $d_{ij} = d_{ji} = 0$ si y solo si $x_i = x_j$, y
(e) para un tercer objeto O_k, se tiene $d_{ik} + d_{jk} \geq d_{ij}$ (desigualdad triangular).

Un ejemplo de disimilaridad es la puntuación de un observador en una escala, por ejemplo en un rango de 0 (idénticos) a 5 (nada parecidos), para dos objetos. Otro ejemplo podría ser el conteo de cuántos atributos en común tienen los dos objetos, sobre un total predefinido de atributos. En estos ejemplos las medidas no se consideran distancias porque no tienen variables asociadas. Las distancias más comunes son la distancia euclídea

$$d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{a=1}^{p} (x_{ia} - x_{ja})^2} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^t (\mathbf{x}_i - \mathbf{x}_j)}, \tag{1.1}$$

que equivale a medir la distancia *en línea recta* entre los puntos representados en un espacio euclídeo; o la distancia de Mahalanobis

$$d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^t \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}, \tag{1.2}$$

donde $\mathbf{S}$ es la matriz de covarianzas obtenida en la configuración donde se encuentran $\mathbf{x}_i$ y $\mathbf{x}_j$, usada para estandarizar la distancia euclídea original.

Similaridades

Por otro lado, hay problemas donde surgen medidas de proximidad que aumentan mientras más *se parecen* dos objetos entre sí. Para estos casos decimos que tenemos una *similaridad* s_{ij} entre O_i y O_j con las siguientes propiedades:

(a) $s_{ij} \geq 0$,
(b) $s_{ij} = s_{ji}$ (simetría),
(c) $s_{ij} \leq s_{ii}$ y $s_{ji} \leq s_{jj}$ para todo $i \neq j$ (identificación).

Esta última condición podría interpretarse como que *ningún objeto se parece más a otro que a sí mismo*. Dependiendo del problema, se podrán fijar restricciones adicionales para el coeficiente de similaridad. Por ejemplo, una restricción común es imponer que $s_{ij} \leq 1$, donde el valor 1 indica que dos objetos son idénticos. Otro caso frecuente es imponer que $s_{ij} \in [-1, 1]$. La ventaja de estos casos es que se pueden derivar disimilaridades δ_{ij} de manera muy sencilla. Por ejemplo con $\delta ij = 1 - s_{ij}$, o con la transformación

$$\delta_{ij} = \sqrt{s_{ii} + s_{jj} - 2s_{ij}} \tag{1.3}$$

que además devuelve una disimilaridad con propiedades de distancia, y que corresponderá a distancias euclídeas bajo ciertas condiciones. En [6] y en [2] se puede encontrar un acercamiento exhaustivo a estos temas. Utilizaremos la siguiente notación para diferenciar los distintos casos:

- $\mathbf{\Delta}_{n\times n}$ para una matriz de disimilaridades δ_{ij} en general.
- $\mathbf{D}_{n\times n}$ para una matriz de distancias d_{ij}. Necesariamente estará vinculada a una configuración de puntos, por lo que podríamos escribir, por ejemplo, $\mathbf{D_X}$ para decir que es una distancia asociada a $\mathbf{X}$.
- $\mathbf{S}_{n\times n}$ para referirnos a una matriz de similaridades s_{ij}.

Definición (Escalado k-dimensional)**.** Una configuración $\mathbf{Z}_{n\times k}$ es un *escalado k-dimensional* de una matriz de disimilaridades $\mathbf{\Delta}_{n\times n}$ si una matriz de distancias $\mathbf{D_Z}$ se aproxima a ella de manera óptima en el sentido de alguna función de pérdida $L(\mathbf{D_Z}, \mathbf{\Delta}) : \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \to \mathbb{R}$. La distancia usada para el escalado es típicamente la euclídea, y es la que usaremos en este trabajo. Llamaremos $\mathcal{Z}$ al espacio vectorial donde se encuentra el escalado.

El problema de MDS constituye entonces en encontrar un escalado $\mathbf{Z}$ para poder visualizar los puntos. Las columnas de $\mathbf{Z}$ podrían pensarse como observaciones de nuevas variables $Z_1, Z_2, ..., Z_k$ para los n objetos, pero a diferencia de otros métodos, no se hará énfasis en su interpretación. En los problemas de aplicación normalmente se impone que $k \leq 3$ para poder graficar los objetos en un espacio visible. Otro aspecto interesante de los métodos de MDS es que ayudan a determinar cuántas dimensiones son importantes para representar los objetos, plantear hipótesis sobre estas estructuras o corroborar hipótesis previas. En este capítulo haremos, a modo de introducción, un recorrido sobre las principales técnicas de MDS y mostraremos algunos ejemplos.

1.2. Escalado clásico

El primer método de escalado multidimensional es el escalado clásico (CMDS - *Classic Multidimensional Scaling*), debido a Torgerson (1952,1958) y Gower (1966), que está muy ligado a otras técnicas como el análisis de componentes principales (PCA) o el análisis factorial (FA).

1.2.1. Representación de distancias euclídeas

Como introducción, supongamos que partimos de las distancias euclídeas d_{ij} de un grupo de n objetos. Nuestro primer problema es construir una configuración vectorial de puntos $\mathbf{X}$ que tenga esa matriz de distancias. Reescribamos 1.1, elevando al cuadrado, y veamos que se compone de tres partes:

$$\begin{aligned} d_{ij}^2 = d^2(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i - \mathbf{x}_j)^t(\mathbf{x}_i - \mathbf{x}_j) \\ &= \mathbf{x}_i^t\mathbf{x}_i + \mathbf{x}_j^t\mathbf{x}_j - 2\mathbf{x}_i^t\mathbf{x}_j \\ &= \sum_{a=1}^{p} x_{ia}^2 + \sum_{a=1}^{p} x_{ja}^2 - 2\sum_{a=1}^{p} x_{ia}\, x_{ja}. \end{aligned} \tag{1.4}$$

Los dos primeros términos de la última igualdad son las normas al cuadrado de ambos vectores, $b_{ii} = ||\mathbf{x}_i||^2$ y $b_{jj} = ||\mathbf{x}_j||^2$, mientras que el tercer término tiene el producto escalar, llamemos $b_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, entre ambos. Para el producto escalar, consideraremos que el punto de referencia es el centro de la configuración de puntos, o el vector dado por las medias de las columnas de $\mathbf{X}$. Por simplicidad también suponemos que dicho vector es el origen. Busquemos ahora una expresión matricial para esta relación. Tenemos $\mathbf{D}^{(2)} = \{d_{ij}^2\}$, la matriz de distancias al cuadrado, y nos conviene definir a $\mathbf{c}$ como vector columna que contiene las n normas $b_{ii} = ||x_i||^2$ para cada punto, y a $\mathbf{1}_n$ un vector columna de unos. De esta forma se puede escribir a $\mathbf{D}^{(2)}$ como:

$$\mathbf{D}^{(2)} = \mathbf{c}\mathbf{1}^t + \mathbf{1}\mathbf{c}^t - 2\mathbf{X}\mathbf{X}^t. \tag{1.5}$$

Los dos primeros términos $\mathbf{c}\mathbf{1}^t$ y $\mathbf{1}\mathbf{c}^t$ son dos matrices de n×n que tienen las normas de los vectores al cuadrado. En el tercer término, la matriz $\mathbf{X}\mathbf{X}^t$ es simétrica y contiene todos los productos internos b_{ij} fuera de la diagonal, y las normas al cuadrado b_{ii} en la diagonal. Veamos que si aplicamos el operador de centrado $\mathbf{H} = \mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^t$ a izquierda y a derecha de $\mathbf{D}^{(2)}$, lo que consiste en hacer el centrado de sus filas y luego el centrado de las columnas resultantes, queda

$$\begin{aligned} \mathbf{H}\mathbf{D}^{(2)}\mathbf{H} &= \mathbf{H}(\mathbf{c}\mathbf{1}^t + \mathbf{1}\mathbf{c}^t - 2\mathbf{X}\mathbf{X}^t)\mathbf{H} \\ &= \mathbf{H}\mathbf{c}\mathbf{1}^t\mathbf{H} + \mathbf{H}\mathbf{1}\mathbf{c}^t\mathbf{H} - 2\mathbf{H}\mathbf{X}\mathbf{X}^t\mathbf{H} \\ &= -2\mathbf{X}\mathbf{X}^t. \end{aligned}$$

El primer y el segundo término se anulan ya que $\mathbf{1}^t\mathbf{H} = \mathbf{H}\mathbf{1} = \mathbf{0}$, y como $\mathbf{X}$ representa vectores centrados en el origen nos queda $\mathbf{H}\mathbf{X}\mathbf{X}^t\mathbf{H} = \mathbf{X}\mathbf{X}^t$. Esto nos permite obtener

$$-\frac{1}{2}\mathbf{H}\mathbf{D}^{(2)}\mathbf{H} = \mathbf{X}\mathbf{X}^t. \tag{1.6}$$

Recordemos que si los objetos tienen coordenadas en $\mathbb{R}^p$ con $p < n$, la matriz de productos internos $\mathbf{X}\mathbf{X}^t$ tiene p autovalores positivos y n − p autovalores nulos. Podemos descomponerla como

$$\mathbf{X}\mathbf{X}^{\mathrm{t}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{t}}, \tag{1.7}$$

donde $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{\mathrm{n}})$ es la matriz diagonal que tiene los autovalores ordenados en forma decreciente y $\mathbf{V}$ es la matriz cuyas columnas $\mathbf{v}_{\mathrm{i}}$ son los autovectores, normalizados de manera que $\mathbf{v}_{\mathrm{i}}^{\mathrm{t}}\mathbf{v}_{\mathrm{i}} = 1$. Con esto se obtiene $\mathbf{X}$ como

$$\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}, \tag{1.8}$$

que tendrá las columnas centradas y ortogonales entre sí. Notemos que también podemos tomar cualquier rotación de $\mathbf{X}$ como solución. Por ejemplo, si proponemos $\mathbf{X}\mathbf{A}$ con $\mathbf{A}$ ortogonal de $\mathrm{p} \times \mathrm{p}$, nos queda $\mathbf{X}\mathbf{A}\mathbf{A}^{\mathrm{t}}\mathbf{X}^{\mathrm{t}} = \mathbf{X}\mathbf{X}^{\mathrm{t}}$.

1.2.2. Escalado clásico de dismilaridades

Ahora partimos de una matriz de disimilaridades $\mathbf{\Delta}$ que puede no tener propiedades de distancia y buscamos encontrar una configuración de puntos $\mathbf{Z}$ cuyas distancias euclídeas las representen *lo mejor posible* en algún sentido. Por el momento, no impongamos ninguna restricción sobre la cantidad de columnas que debe tener $\mathbf{Z}$. Con las ideas de la sección anterior, podemos calcular

$$\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{\Delta}^{(2)}\mathbf{H}, \tag{1.9}$$

y obtener su descomposición en autovalores y autovectores, igual que en 1.7. La diferencia radica en que $\mathbf{\Delta}$ puede tener autovalores negativos, ya que no necesariamente es una matriz de distancias. Si conservamos únicamente los r autovalores positivos con sus respectivos autovectores, podríamos proponer

$$\mathbf{U} = \mathbf{V}_{\mathrm{r}}\mathbf{\Lambda}_{\mathrm{r}}^{1/2} \tag{1.10}$$

como posible solución. Observemos que necesariamente $\mathrm{r} \leq \mathrm{n} - 1$, ya que $\mathbf{B}$ tendrá al menos un autovalor nulo debido a la operación de doble centrado. De esta manera se verifica que

$$\mathbf{U}\mathbf{U}^{\mathrm{t}} = -\frac{1}{2}\mathbf{H}\mathbf{D}_{\mathrm{U}}^{(2)}\mathbf{H} \tag{1.11}$$

y podríamos pensar que los productos internos $\mathbf{U}\mathbf{U}^{\mathrm{t}}$ aproximan a $\mathbf{B}$. Debemos ver si esta solución es apropiada, y ver en qué sentido estamos aproximando.

Definición (Norma de Frobenbius)**.** Sea $\mathbf{A}_{\mathrm{n}\times\mathrm{p}}$, se define la norma de Frobenius de $\mathbf{A}$ como

$$||\mathbf{A}||_{\mathrm{F}} = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{\mathrm{j}=1}^{\mathrm{p}} \mathrm{a}_{\mathrm{ij}}^2} = \sqrt{\mathrm{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{A})} = \sqrt{\sum_{\mathrm{i}=1}^{\min(\mathrm{n},\mathrm{p})} \delta_{\mathrm{i}}^2},$$

donde δ_{i} son los autovalores no nulos de $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. Si se tiene otra matriz del mismo tamaño, llamemos $\mathbf{C}_{\mathrm{n}\times\mathrm{p}}$, la distancia en norma de Frobenius de $\mathbf{C}$ a $\mathbf{A}$ resultará

$$||\mathbf{C} - \mathbf{A}||_{\mathrm{F}} = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{\mathrm{j}=1}^{\mathrm{p}} (\mathrm{c}_{\mathrm{ij}} - \mathrm{a}_{\mathrm{ij}})^2} = \sqrt{\mathrm{tr}(\mathbf{C} - \mathbf{A})^{\mathrm{T}}(\mathbf{C} - \mathbf{A})}.$$

La norma de Frobenius es el análogo para matrices de la distancia euclídea entre dos vectores, y la usaremos para establecer un criterio de aproximación. Llamemos $\mathcal{U}$ al conjunto de todas las matrices de n filas y k columnas, con rango completo k. Sea $\mathbf{D}_\mathbf{U}$ la matriz de distancias euclídeas y $\mathbf{U}\mathbf{U}^\mathrm{t}$ la matriz de productos internos de cualquier $\mathbf{U} \in \mathcal{U}$. El escalado, si existe, deberá ser

$$\begin{aligned} \mathbf{Z} &= \underset{\mathbf{U}\in\mathcal{U}}{\operatorname{argmin}}\Big\{\mathrm{L}(\mathbf{D}_\mathbf{U}, \boldsymbol{\Delta})\Big\}, \\ \text{con} \quad \mathrm{L}(\mathbf{D}_\mathbf{U}, \boldsymbol{\Delta}) &= \Big\|\mathbf{B} - \mathbf{U}\mathbf{U}^\mathrm{t}\Big\|_\mathrm{F}^2 = \left\|\frac{1}{2}\mathbf{H}\big(\mathbf{D}_\mathbf{U}^{(2)} - \boldsymbol{\Delta}^{(2)}\big)\mathbf{H}\right\|_\mathrm{F}^2. \end{aligned} \tag{1.12}$$

A esta función de pérdida se denomina *Strain*, $\mathrm{L}(\mathbf{D}_\mathbf{U}, \boldsymbol{\Delta}) = \mathrm{Strain}(\mathbf{D}_\mathbf{U}, \boldsymbol{\Delta})$.

Teorema 1.1 (Teorema de Eckart-Young-Mirsky para la norma Frobenius)**.** Dada una matriz $\mathbf{A}_{\mathrm{n}\times\mathrm{p}}$ con $\mathrm{n} \geq \mathrm{p}$, de rango m, expresada en descomposición de valores singulares

$$\mathbf{A} = \mathbf{L}_\mathrm{m}\boldsymbol{\Phi}_\mathrm{m}\mathbf{W}_\mathrm{m}^\mathrm{t} = \sum_{\mathrm{j}=1}^{\mathrm{m}} \varphi_\mathrm{j}\mathbf{l}_\mathrm{j}\mathbf{w}_\mathrm{j}^\mathrm{t},$$

donde $\boldsymbol{\Phi}_\mathrm{m} = \mathrm{diag}(\varphi_1, \varphi_2, ..., \varphi_\mathrm{m})$ tiene únicamente los valores singulares positivos, $\mathbf{L}_\mathrm{m} \in \mathbb{R}^{\mathrm{n}\times\mathrm{m}}$ tiene los autovectores izquierdos ortogonales $\mathbf{l}_\mathrm{j}$ en sus columnas y $\mathbf{W}_\mathrm{m}^\mathrm{t} \in \mathbb{R}^{\mathrm{m}\times\mathrm{p}}$ tiene los autovectores derechos ortogonales $\mathbf{w}_\mathrm{j}$ en sus filas. Sea $\mathcal{U}$ el conjunto de todas las matrices de n filas y rango $\mathrm{r} < \mathrm{m}$, la matriz $\mathbf{A}_{(\mathrm{r})} \in \mathcal{U}$ que mejor aproxima a $\mathbf{A}$ es

$$\mathbf{A}_{(\mathrm{r})} = \underset{\mathbf{U}\in\mathcal{U}}{\operatorname{argmin}} ||\mathbf{U} - \mathbf{A}||_\mathrm{F}^2 = \sum_{\mathrm{j}=1}^{\mathrm{r}} \varphi_\mathrm{j}\mathbf{l}_\mathrm{j}\mathbf{w}_\mathrm{j}^\mathrm{t} = \mathbf{L}_\mathrm{r}\boldsymbol{\Phi}_\mathrm{r}\mathbf{W}_\mathrm{r}^\mathrm{t},$$

y se cumple que

$$\min_{\mathbf{U}\in\mathcal{U}} ||\mathbf{U} - \mathbf{A}||_\mathrm{F}^2 = \sum_{\mathrm{j}=\mathrm{r}+1}^{\mathrm{r}} \varphi_\mathrm{j}^2.$$

Demostración. La demostración se puede encontrar en el apéndice 4 de [4]. □

Veamos que podemos expresar

$$\mathbf{B} = -\frac{1}{2}\mathbf{H}\boldsymbol{\Delta}^2\mathbf{H} = \sum_{\mathrm{j}=1}^{\mathrm{n}-1} \lambda_\mathrm{j}\mathbf{v}_\mathrm{j}\mathbf{v}_\mathrm{j}^\mathrm{t} \qquad \text{y} \qquad \mathbf{U}\mathbf{U}^\mathrm{t} = (\mathbf{V}_\mathrm{r}\boldsymbol{\Lambda}_\mathrm{r}^{1/2})(\mathbf{V}_\mathrm{r}\boldsymbol{\Lambda}_\mathrm{r}^{1/2})^\mathrm{t} = \sum_{\mathrm{j}=1}^{\mathrm{r}} \lambda_\mathrm{j}\mathbf{v}_\mathrm{j}\mathbf{v}_\mathrm{j}^\mathrm{t}.$$

Con esto reemplazamos

$$\begin{aligned} \mathrm{Strain}(\mathbf{D}_\mathbf{U}, \boldsymbol{\Delta}) &= ||\ \mathbf{B} - \mathbf{U}\mathbf{U}^\mathrm{t}\ ||_\mathrm{F}^2 \\ &= \left\| \sum_{\mathrm{j}=1}^{\mathrm{n}-1} \lambda_\mathrm{j}\mathbf{v}_\mathrm{j}\mathbf{v}_\mathrm{j}^\mathrm{t} - \sum_{\mathrm{j}=1}^{\mathrm{r}} \lambda_\mathrm{j}\mathbf{v}_\mathrm{j}\mathbf{v}_\mathrm{j}^\mathrm{t} \right\|_\mathrm{F}^2 \\ &= \left\| \sum_{\mathrm{j}=\mathrm{r}+1}^{\mathrm{n}-1} \lambda_\mathrm{j}\mathbf{v}_\mathrm{j}\mathbf{v}_\mathrm{j}^\mathrm{t} \right\|_\mathrm{F}^2 \\ &= \sum_{\mathrm{j}=\mathrm{r}+1}^{\mathrm{n}-1} \lambda_\mathrm{j}^2, \end{aligned}$$

que verifica ser un mínimo de acuerdo con del teorema 1.1. Con esto mostramos que la matriz $\mathbf{U}$ es efectivamente el escalado $\mathbf{Z}$ que cumple 1.12. Además, de lo visto se desprende que si tenemos una dimensionalidad prefijada $k < r$ que típicamente será 2 o 3, la matriz $\mathbf{V}_k\mathbf{\Lambda}_k^{1/2}$ es la que aproxima de manera óptima entre todas las matrices de rango k que cumplan las características pedidas. Una medida de error asociada a un escalado k dimensional obtenido de esta forma es

$$\sigma_{\mathbf{B}} = \frac{||\mathbf{B} - \mathbf{Z}\mathbf{Z}^T||_F}{||\mathbf{B}||_F} = \sqrt{\frac{\sum_{j=k+1}^{n-1}\lambda_j^2}{\sum_{j=1}^{n-1}\lambda_j^2}}. \tag{1.13}$$

Notemos que si la disimilaridad $\mathbf{\Delta}$ está *muy lejos* de representar distancias euclídeas entre vectores, los autovalores negativos tendrán mucho peso y el coeficiente se acercará a 1. En este caso, el escalado no será una buena representación de los objetos.

Podemos resumir las propiedades de la solución:

(a) Las columnas de $\mathbf{Z}$ tienen media 0.

(b) Para la matriz de covarianzas tenemos

$$\hat{\Sigma}_{\mathbf{Z}} = \frac{1}{n}\mathbf{Z}^t\mathbf{Z} = \frac{1}{n}\mathbf{\Lambda}_k^{1/2}\mathbf{V}_k^t\mathbf{V}_k\mathbf{\Lambda}_k^{1/2} = \frac{1}{n}\mathbf{\Lambda}_k. \tag{1.14}$$

(c) Si $\mathbf{T}$ es una matriz ortogonal de $k \times k$ y $\mathbf{t}$ es un vector de $\mathbb{R}^k$, la transformación $\mathbf{Z'} = \mathbf{Z}\mathbf{T} + \mathbf{t}$ también es solución óptima.

(d) Las soluciones óptimas para distintos valores de $k \leq r$ están anidadas. Es decir, solo basta con eliminar el k ésimo autovalor y su autovector asociado para obtener la solución óptima de dimensión $k - 1$ y tendrá las mismas propiedades mencionadas antes.

Algoritmo 1 Escalado multidimensional clásico - CMDS

Entrada: Disimilaridad $\mathbf{\Delta}_{n\times n}$ y dimensión deseada k.

1. Calcular $\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{\Delta}^2\mathbf{H}$.
2. Realizar la descomposición $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^t$ con todos los autovalores y autovectores.
3. Verificar que la cantidad de autovalores positivos en $\mathbf{\Lambda}$ sea mayor o igual a k. Si esto no se cumple, salir.
4. Extraer los primeros k autovalores en la diagonal de $\mathbf{\Lambda}_k$ y sus autovectores como columnas de $\mathbf{V}_k$.
5. Calcular $\mathbf{Z} = \mathbf{V}_k\mathbf{\Lambda}_k^{1/2}$.

Salida: $\mathbf{Z}$, configuración de n puntos en k dimensiones.

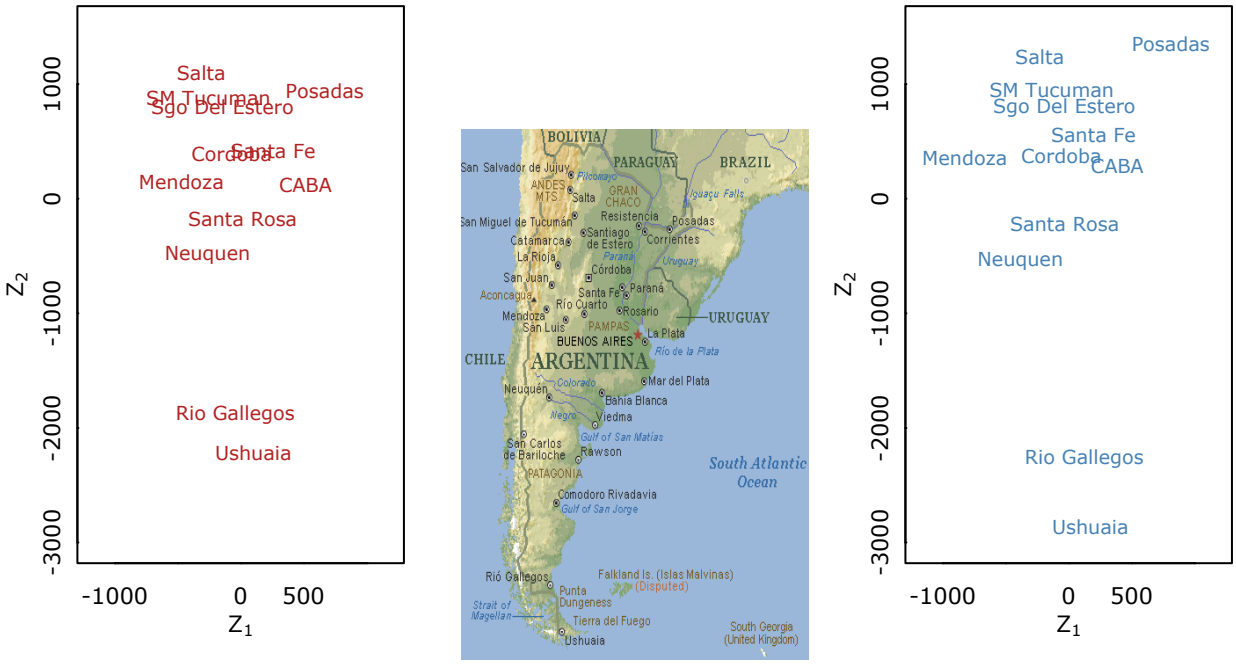
Ejemplo (Ciudades de Argentina)**.** En las siguientes tablas se dan las distancias entre diez ciudades de Argentina, en línea recta y en automóvil por rutas nacionales:

	CABA	Cordoba	Mendoza	Neuquen	Posadas	Gallegos	Tucuman	Salta	Santa Fe	Sta.Rosa	Sgo.Estero	Ushuaia
CABA	0											
Cordoba	646	0										
Mendoza	985	466	0									
Neuquen	989	907	676	0								
Posadas	834	919	1384	1709	0							
Gallegos	2082	2281	2081	1410	2914	0						
Tucuman	1080	517	756	1370	924	2773	0					
Salta	1282	745	957	1591	992	2997	228	0				
Santa Fe	393	330	775	1049	664	2325	689	889	0			
Sta.Rosa	579	577	586	422	1293	1712	1088	1316	641	0		
Sgo.Estero	939	407	713	1286	827	2677	141	353	547	977	0	
Ushuaia	2373	2618	2435	1762	3207	359	3116	3341	2641	2044	3016	0

Tabla 1.1: Distancias entre ciudades de Argentina en línea recta

	CABA	Cordoba	Mendoza	Neuquen	Posadas	Gallegos	Tucuman	Salta	Santa Fe	Sta.Rosa	Sgo.Estero	Ushuaia
CABA	0											
Cordoba	715	0										
Mendoza	1050	670	0									
Neuquen	1158	1137	855	0								
Posadas	1040	1213	1925	2198	0							
Gallegos	2635	3635	2800	1930	3675	0						
Tucuman	1203	590	1005	1860	1106	3225	0					
Salta	1510	897	1227	2082	1118	3532	307	0				
Santa Fe	478	330	885	1347	883	2845	767	1074	0			
Sta.Rosa	620	600	765	537	1660	2035	1190	1497	810	0		
Sgo.Estero	1043	430	977	1567	948	3065	160	467	607	1030	0	
Ushuaia	3228	3228	3393	2523	4268	593	3818	4125	3438	2628	3658	0

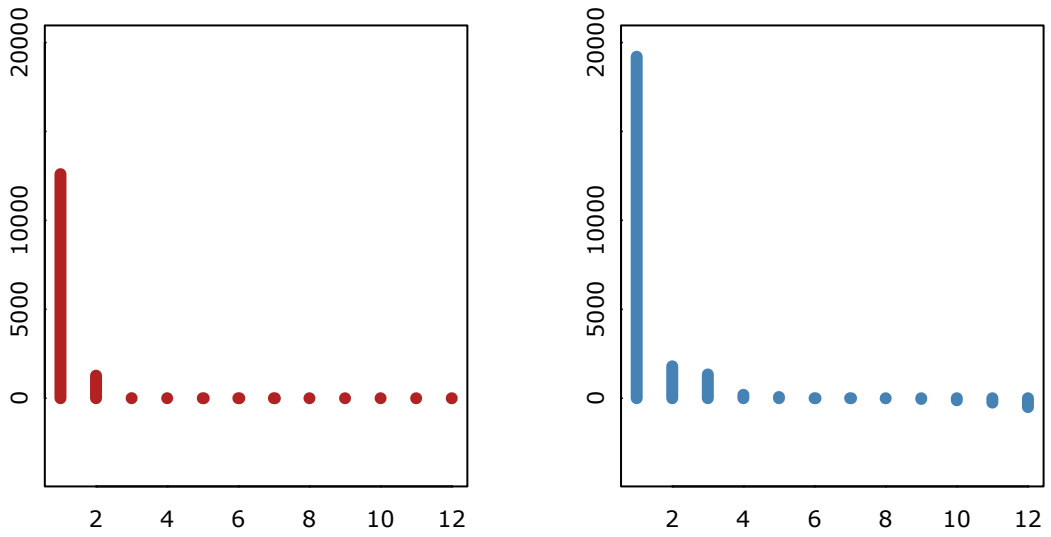
Tabla 1.2: Distancias entre ciudades de Argentina por rutas nacionales



(a) CMDS distancias rectas **(b)** Mapa de Argentina **(c)** CMDS distancias por ruta

Figura 1.1

En las figuras se observan los resultados de aplicar el CMDS con k = 2 para ambas matrices de distancia. Observemos que medir las distancias por ruta puede pensarse como agregar error a los valores originales, y esto hace que aparezcan autovalores negativos en **B**.



(a) Rectas **(b)** Rutas

Figura 1.2: Autovalores (en miles)

El coeficiente de error en la pérdida Strain es $\sigma_{\mathbf{B}} = 0.0007108976$ para las distancias en línea recta y $\sigma_{\mathbf{B}} = 0.07594782$ para las distancias por ruta. Para analizar los errores de representación de la figura 1.1 con mayor detalle, podemos tomar la diferencia

$$\mathrm{err}_{ij} = \delta_{ij} - d_{ij}, \tag{1.15}$$

donde d_{ij} son distancias euclídeas tomadas en $\mathcal{Z}$.

	CABA	Cordoba	Mendoza	Neuquen	Posadas	Gallegos	Tucuman	Salta	Sta.Fe	Sta.Rosa	Sgo.Estero	Ushuaia
CABA	0.00											
Cordoba	262.49	0.00										
Mendoza	-162.80	-97.21	0.00									
Neuquen	30.02	168.51	-140.23	0.00								
Posadas	-104.38	-92.83	10.64	-35.36	0.00							
Gallegos	69.85	-10.63	13.87	129.50	-4.05	0.00						
Tucuman	363.18	13.33	95.78	355.56	78.51	3.67	0.00					
Salta	375.43	14.77	160.35	297.98	71.86	9.49	-1.85	0.00				
Sta.Fe	147.01	15.31	-157.25	106.29	-117.87	20.46	257.16	270.16	0.00			
Sta.Rosa	-34.05	8.38	-212.58	60.39	-118.18	-19.23	22.84	27.21	1.70	0.00		
Sgo.Estero	382.15	7.26	71.55	187.94	-62.06	-0.57	-19.70	-17.29	274.87	16.87	0.00	
Ushuaia	72.58	-16.06	25.74	133.38	9.09	-5.47	-1.81	4.10	17.85	-24.70	-5.82	0.00

Tabla 1.3: Errores de representación, en km, del MDS en $\mathbb{R}^2$ de las distancias en rutas

Ejemplo (Percepción de colores)**.** Ekman, G. (1954) presenta similaridades para 14 colores basadas en la calificación de 31 personas. En el experimento, cada persona calificó cada par de colores en una escala de 5 puntos, desde 0 = *no hay similaridad* hasta 4 = *idénticos*. Luego, se calculó el promedio y las similaridades se dividieron por 4 para quedar en el intervalo unitario. En la diagonal se pusieron similaridades iguales a 1. Cada color se identifica con la longitud de onda, entre 434 y 674 nm.

	434	445	465	472	490	504	537	555	584	600	610	628	651	674
434	1.00													
445	0.86	1.00												
465	0.42	0.50	1.00											
472	0.42	0.44	0.81	1.00										
490	0.18	0.22	0.47	0.54	1.00									
504	0.06	0.09	0.17	0.25	0.61	1.00								
537	0.07	0.07	0.10	0.10	0.31	0.62	1.00							
555	0.04	0.07	0.08	0.09	0.26	0.45	0.73	1.00						
584	0.02	0.02	0.02	0.02	0.07	0.14	0.22	0.33	1.00					
600	0.07	0.04	0.01	0.01	0.02	0.08	0.14	0.19	0.58	1.00				
610	0.09	0.07	0.02	0.00	0.02	0.02	0.05	0.04	0.37	0.74	1.00			
628	0.12	0.11	0.01	0.01	0.01	0.02	0.02	0.03	0.27	0.50	0.76	1.00		
651	0.13	0.13	0.05	0.02	0.02	0.02	0.02	0.02	0.20	0.41	0.62	0.85	1.00	
674	0.16	0.14	0.03	0.04	0.00	0.01	0.00	0.02	0.23	0.28	0.55	0.68	0.76	1.00

Tabla 1.4: Similaridades entre colores, datos originales

Realizamos CMDS en dos y tres dimensiones, transformando las similaridades originales en disimilaridades como:

$$\delta_{ij} = 1 - s_{ij}$$

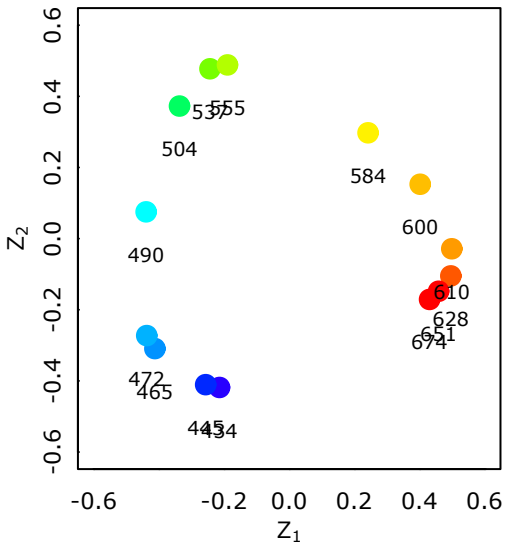


Figura 1.3: Representación de colores en $\mathbb{R}^2$

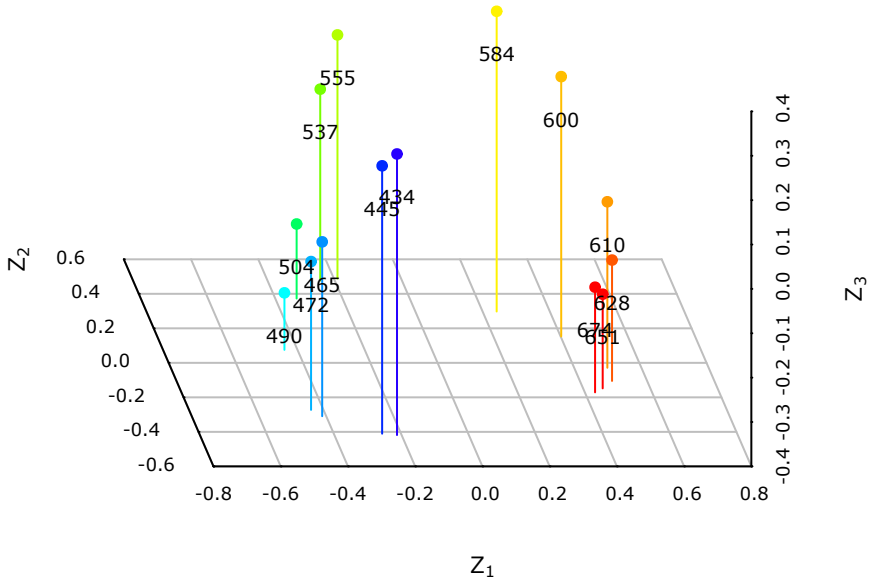


Figura 1.4: Representación de colores en $\mathbb{R}^3$

Los coeficientes de bondad de ajuste en pérdida Strain son $\sigma_{\mathbf{B}} = 0.25048$ para k=2 y $\sigma_{\mathbf{B}} = 0.1740711$ para k = 3. Si se busca una representación visual, solo basta con calcular estos errores. Sin embargo, en ejemplos similares a éste se busca analizar qué dimensionalidad intrínseca tienen los objetos; es decir, si hay un valor de k a partir del cual la representación no agrega información significativa.

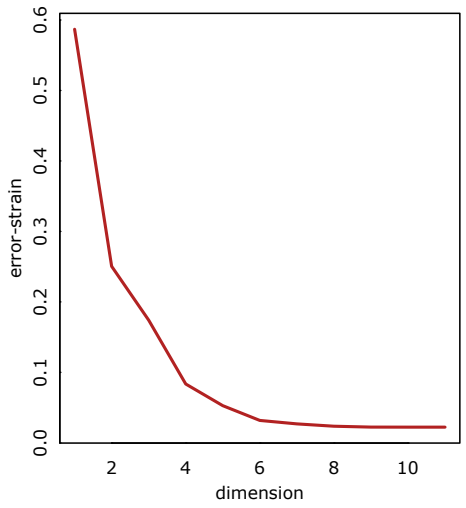


Figura 1.5: Coeficiente de error Strain en función de k

Identificar la dimensionalidad no siempre es claro. En el gráfico 1.5 se observa cómo a partir de $k = 6$ prácticamente no se explica error. De todas formas, podríamos preguntarnos si la estructura que se observa en $\mathbb{R}^3$ captura algún aspecto del problema que pueda ser explicado, o sólo se trata de ruido.

1.2.3. Relación del escalado clásico con componentes principales

Dada una configuración de puntos de dimensión p centrados en el origen, con coordenadas $\mathbf{X}_{n\times p}$, que tienen varianza

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}\mathbf{X}^t\mathbf{X},$$

las componentes principales muestrales se obtienen con las direcciones dadas por sus autovectores $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, ..., \boldsymbol{\gamma}_p$, conocidas como *direcciones principales*. Sean $a_1, a_2, ..., a_p$ sus respectivos autovalores. Una observación $\mathbf{x}$ se proyecta a cada una de estas direcciones como $\mathbf{x}^t\boldsymbol{\gamma}_j$, formando la j-ésima *componente principal* de $\mathbf{x}$, que es una nueva variable real que resume información del vector.

Se puede ver que PCA tiene una relación dual con CMDS. Para eso, supongamos que obtenemos las distancias euclídeas $\mathbf{D}_X$ a través de 1.4, y hacemos CMDS como si fueran disimilaridades de partida, con el objetivo de reconstruir $\mathbf{X}$. Obtenemos

$$\mathbf{B} = -\tfrac{1}{2}\mathbf{H}\mathbf{D}^{(2)}\mathbf{H},$$

de la que extraemos los primeros p autovectores $\mathbf{v}_j$ y sus autovalores λ_j y construimos el escalado $\mathbf{Z}_p = \mathbf{\Lambda}_p^{1/2}\mathbf{V}_p$. Veamos que, al ser autovectores se cumple

$$\mathbf{B}\mathbf{v}_j = \mathbf{X}\mathbf{X}^t\mathbf{v}_j = \lambda_j\mathbf{v}_j,$$

que pre-multiplicando por $\mathbf{X}^t/n$ queda

$$\frac{1}{n}\mathbf{X}^t\mathbf{X}(\mathbf{X}^t\mathbf{v}_j) = \frac{\lambda_j}{n}(\mathbf{X}^t\mathbf{v}_j),$$

donde se ve que $\mathbf{X}^t\mathbf{v}_j = \boldsymbol{\gamma}_j$ y que $\lambda_j/n = a_j$ para $1 \leq j \leq p$. Si ponemos los p autovectores $\boldsymbol{\gamma}_j$ como columnas de una matriz $\mathbf{\Gamma}$, podemos obtener las coordenadas de $\mathbf{X}$ proyectadas a las componentes principales, también llamadas *scores*, como

$$\begin{aligned}\mathbf{X}\mathbf{\Gamma} &= [\mathbf{X}\boldsymbol{\gamma}_1,\ \mathbf{X}\boldsymbol{\gamma}_2,\ ...\ ,\ \mathbf{X}\boldsymbol{\gamma}_p] \\ &= [\mathbf{X}\mathbf{X}^t\mathbf{v}_1, \mathbf{X}\mathbf{X}^t\mathbf{v}_2,\ ...\ ,\ \mathbf{X}\mathbf{X}^t\mathbf{v}_p] \\ &= [\lambda_1\mathbf{v}_1,\ \lambda_2\mathbf{v}_2,\ ...\ ,\ \lambda_p\mathbf{v}_p].\end{aligned}$$

Si estandarizamos estas columnas, dividiendo cada una por $\lambda_j^{1/2}$ obtenemos las columnas de $\mathbf{Z}$:

$$\mathbf{X}\mathbf{\Gamma}\mathbf{\Lambda}^{-1/2} = \mathbf{X}\mathbf{S}^{-1/2} = [\lambda_1^{1/2}\mathbf{v}_1,\ \lambda_2^{1/2}\mathbf{v}_2,\ ...\ ,\ \lambda_p^{1/2}\mathbf{v}_p] = \mathbf{Z}_p \tag{1.16}$$

Las coordenadas del escalado clásico de $\mathbf{D}_\mathbf{X}$ son el resultado de la estandarización multivariante de $\mathbf{X}$, que a su vez son los *scores* llevados a varianza unitaria. Si el escalado tiene $k \leq p$ dimensiones, esto es equivalente a conservar únicamente los primeros k *scores*. Gower (1968) llama *coordenadas principales* a estos puntos y Análisis de Coordenadas Principales (PCO) al CMDS, para darle un marco en relación a otras técnicas clásicas.

1.3. Escalado de distancias

En lo visto hasta ahora, estuvo implícita la idea de que detrás de un conjunto de disimilaridades existe una configuración desconocida de puntos que se desea aproximar. A partir de este planteo surge un panorama más amplio para el problema de MDS con los trabajos de Kruskal (1964) y Shepard (1962), que sugieren una regresión directa para las distancias.

Planteo general del modelo

Dado un conjunto de objetos $\{O_1, O_2, ..., O_n\}$, supongamos que tienen asociada una configuración vectorial en dimensión finita en un espacio $\mathcal{Z}$, con sus distancias euclídeas d_{ij}. Para estos objetos se tienen *observaciones* de disimilaridad δ_{ij} de la forma

$$\delta_{ij} = d_{ij} + \varepsilon_{ij}, \tag{1.17}$$

donde ε_{ij} son variables aleatorias que capturan errores de medición, efectos de muestreo o el sesgo debido al método de medición. Con esto, se buscará un escalado $\mathbf{Z}$ cuyas distancias, llamemos $d_{ij(\mathbf{Z})}$ para diferenciarlas de las verdaderas, las *aproximen* minimizando la suma de cuadrados

$$L(\mathbf{D_Z}, \boldsymbol{\Delta}) = \sigma^2_{abs} = \sum_{j<i} \left(\delta_{ij} - d_{ij(\mathbf{Z})}\right)^2, \tag{1.18}$$

que se denomina *Stress*. Para encontrar $\mathbf{Z}$ se plantean ecuaciones normales (ver Gower en [4]) y se implementan algoritmos de optimización. Una cobertura de los distintos métodos de resolución que surgieron en el campo del MDS para la minimización del *Stress* y sus variantes se puede encontrar en [2] y en [8]. El enfoque de Kruskal es hoy en día el denominador común de la mayoría de los métodos de MDS. Para algunos autores se denomina *escalado de cuadrados mínimos* o *escalado de distancias* para diferenciarlo del CMDS.

Incorporación de una función de mapeo

En lugar de intentar ajustar directamente las disimilaridades originales, un planteo más flexible es transformarlas previamente. Para eso se utiliza una *función de mapeo* f, no decreciente y continua, de manera que ahora

$$f(\delta_{ij}) = d_{ij} + \varepsilon_{ij}. \tag{1.19}$$

Un caso sencillo es proponer una relación lineal en parámetros β_0 y β_1. Con esta idea, observemos que sin perder generalidad se puede reformular 1.19 de la siguiente manera:

$$d_{ij} = f(\delta_{ij}) + \varepsilon_{ij} = \beta_0 + \beta_1\ \delta_{ij} + \varepsilon_{ij}. \tag{1.20}$$

Transformaciones como esta permiten darle propiedades de métrica a disimilaridades que pueden no tenerlas (ver capítulo 19 de [1]). Otros ejemplos de modelado paramétrico involucran polinomios monótonos en δ_{ij}, *splines*, y funciones exponenciales o logarítmicas. Si el mapeo es estrictamente creciente, se dice que el escalado es *métrico*, mientras que si sólo busca preservar el orden de las disimilaridades, se dice que el escalado es *ordinal* o no métrico. Las características del modelado dependerán en gran medida de la aplicación puntual.

Podemos ver que el problema tiene dos partes que se trabajan simultáneamente. Por un lado, se debe ajustar la función de mapeo propuesta, ya sea paramétrica o no. Esto produce nuevas disimilaridades $\hat{f}(\delta_{ij})$ llamadas *disparidades* $\hat{d}_{ij}$. Luego, se debe encontrar una configuración de puntos en $\mathbb{R}^{n\times k}$ cuyas distancias euclídeas aproximen a estas disparidades. Supongamos que f es paramétrica y depende de ciertos coeficientes $\vartheta \in \Theta$, y sea $\mathcal{U}$ el conjunto de todas las configuraciones de n puntos de dimensión k, podemos reformular el problema 2.2 como

$$\min_{\mathbf{U}\in\mathcal{U},\vartheta\in\Theta}\Big\{\text{Stress}(\mathbf{D_U},\boldsymbol{\Delta}) = \sum_{j<i} e_{ij}^2 = \sum_{j<i}\Big(\hat{f}_\vartheta(\delta_{ij}) - d_{ij(\mathbf{U})}\Big)^2\Big\}, \tag{1.21}$$

donde e_{ij} son los residuos del ajuste. Para encontrar la solución con un modelo lineal como 1.20, en este trabajo utilizaremos el método SMACOF, *stress majorization of a complicated function*, aplicado a MDS por De Leeuw (1977,1988). Este método se encuentra desarrollado en detalle en el Capítulo 8 de [1].

Se puede postular un ANOVA para el escalado multidimensional (ver Gower en [4]) como

$$\sum_{j<l}\delta_{ij}^2 = \sum_{j<l} d_{ij(\mathbf{Z})}^2 + \sum_{j<l}\Big(\delta_{ij} - d_{ij(\mathbf{Z})}\Big)^2 \tag{1.22}$$

donde puede reemplazarse δ_{ij} por $\hat{d}_{ij}$ para el caso de disimilaridades transformadas. De esta descomposición se desprende una medida de error ampliamente utilizada en MDS, el *Stress-1* de Kruskal

$$\sigma_1 = \sqrt{\frac{\sum_{j<i}\Big(\delta_{ij} - d_{ij}(\mathbf{Z})\Big)^2}{\sum_{j<i} d_{ij}(\mathbf{Z})^2}} = \frac{||\boldsymbol{\Delta} - \mathbf{D_Z}||_F}{||\mathbf{D_Z}||_F}, \tag{1.23}$$

que juega un papel análogo al coeficiente del *Strain* normalizado de 1.13 para el CMDS. Algunas características generales y propiedades de las soluciones del escalado de distancias son:

(a) Si se aplica el algoritmo SMACOF, la solución óptima tiene las columnas centradas en el origen.

(b) Si $\mathbf{T}$ es una matriz ortogonal de $k \times k$ y $\mathbf{t}$ es un vector de $\mathbb{R}^k$, la transformación $\mathbf{Z'} = \mathbf{ZT} + \mathbf{t}$ también es solución óptima, al igual que en CMDS.

(c) Las soluciones óptimas para distintos valores de $k \leq r$ no están anidadas como en CMDS, al cambiar la dimensionalidad debe realizarse nuevamente el proceso de optimización.

Continuemos con el ejemplo de las rutas argentinas para mostrar las particularidades de este planteo e ilustrar las diferencias con el CMDS.

Ejemplo (Continuación - Ciudades de Argentina)**.** Para el ejemplo de las ciudades de Argentina, planteamos el escalado de las distancias por ruta dadas en la tabla 1.2 ajustando el modelo dado por 1.20. Para la resolución se utilizó el algoritmo SMACOF.

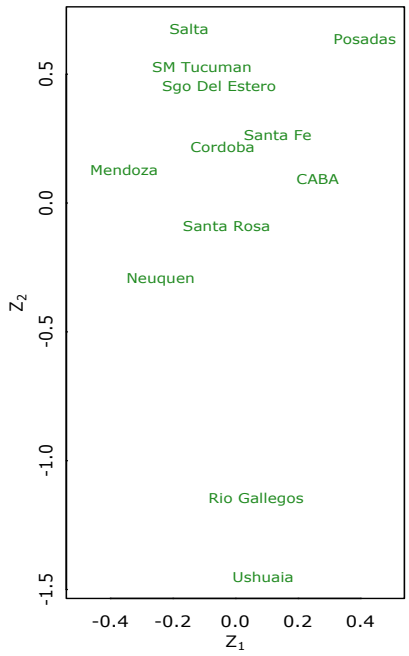
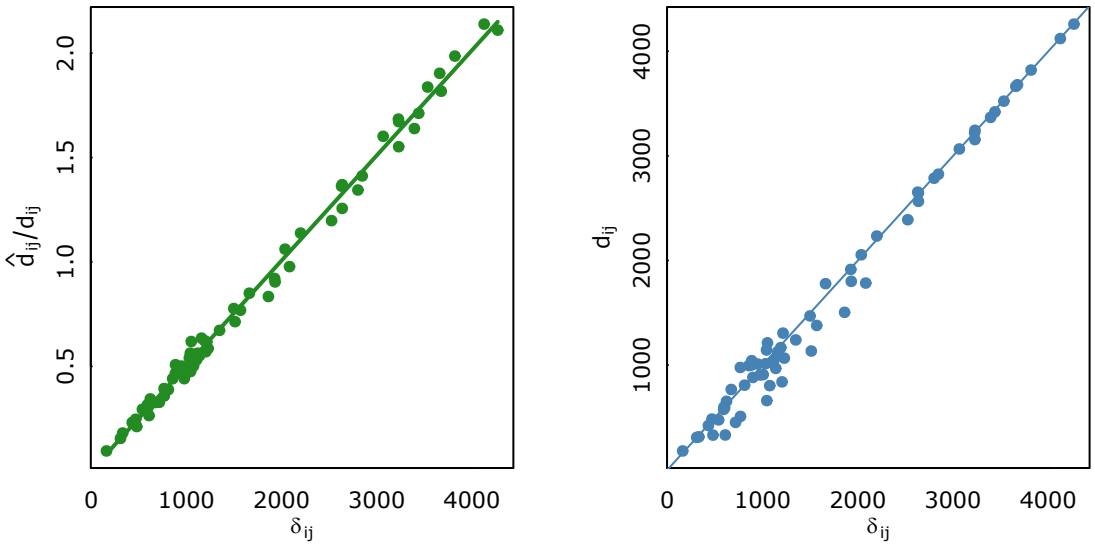


Figura 1.6: Escalado de distancias por ruta admitiendo una transformación lineal



(a) Escalado de distancias **(b)** CMDS

Figura 1.7: Diagramas de Shepard, MDS de las distancias por ruta

Los gráficos de la figura 1.7 permiten ver el ajuste de distancias (o disparidades) en relación a las disimilaridades de partida.

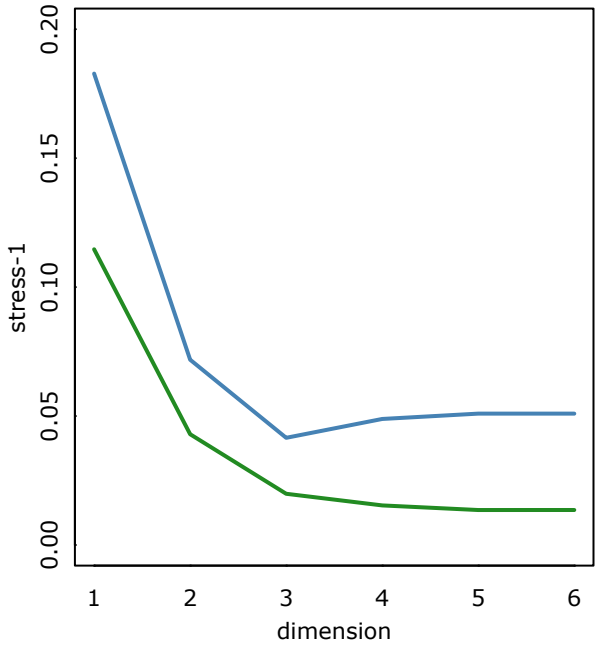


Figura 1.8: Stress-1 en función de k

En el gráfico se observa cómo disminuye el coeficiente Stress-1 dado en 1.23 con escalado de distancias (en verde) y con CMDS (en azul). Cabe remarcar que el CMDS no trabaja minimizando esta pérdida.

1.4. MDS como técnica de reducción de la dimensión

Dada una configuración de puntos $\mathbf{X}_{n\times p}$, mostremos un ejemplo de cómo utilizar el escalado multidimensional para encontrar una reducción de dimensión similar a componentes principales y construir un biplot.

Ejemplo (Calidad de vinos)**.** Se tienen datos de 100 botellas de vino tinto provenientes de distintas bodegas europeas. Para cada botella se consideraron 4 parámetros de calidad: Dióxido de azufre libre (`free.sulfur.dioxide`), densidad (`density`), `pH`, y `alcohol`. Para cada botella se mide una escala discreta del 1 al 5 en los parámetros nombrados; donde 1 indica un nivel muy bajo o nulo, y 5 indica un nivel muy alto.

obs	`total.sulfur.dioxide`	`density`	`pH`	`alcohol`
1	2	1	5	5
2	2	1	4	1
3	3	2	1	3
4	1	2	2	1
5	2	1	4	2
6	...	...	...	...

Buscamos una representación en dos dimensiones de los datos. Primero realizamos un análisis de componentes principales con la matriz de correlaciones de los datos centrados, obteniendo:

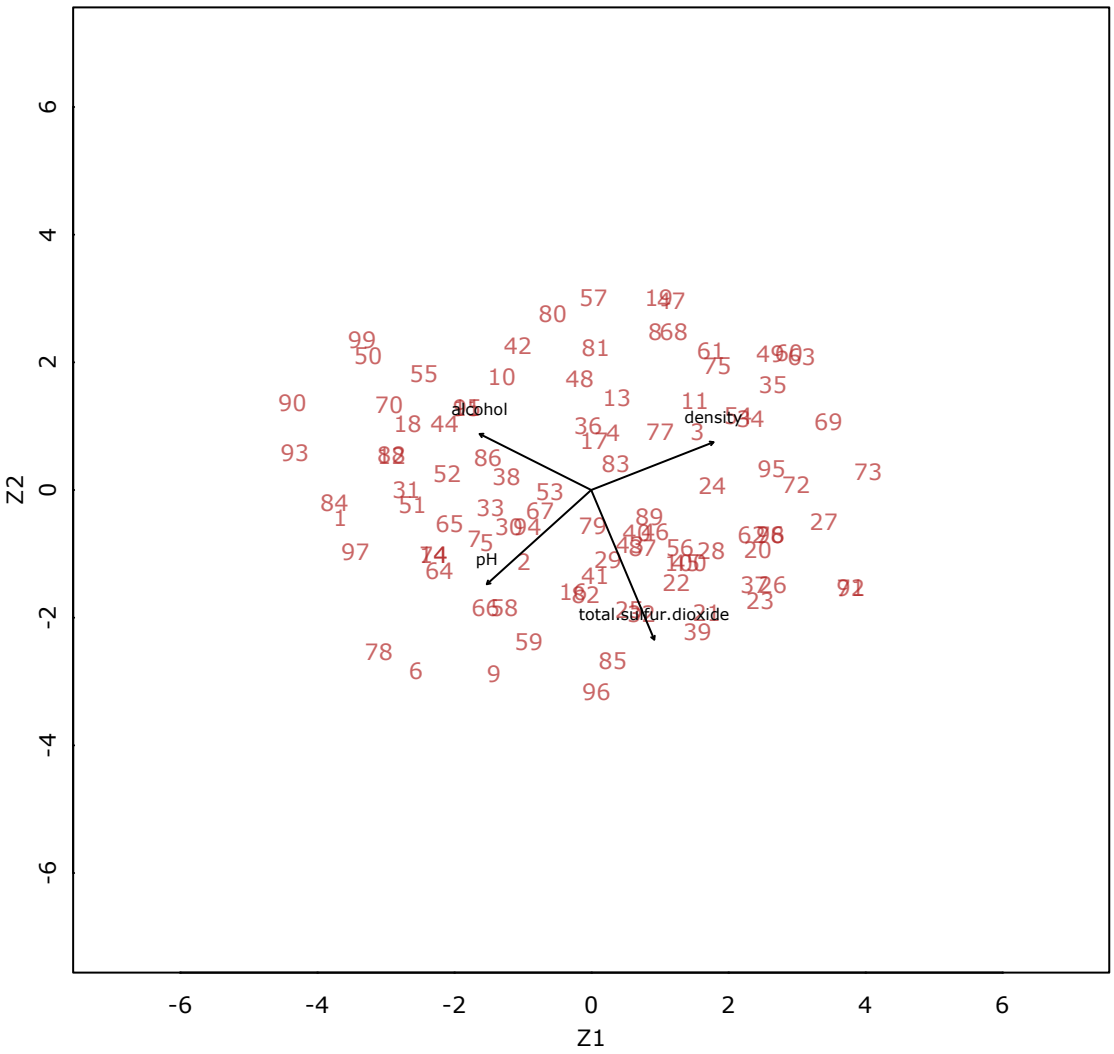


Figura 1.9: Un biplot de Componentes Principales

Sea $\hat{\mathbf{\Sigma}}$ la matriz de covarianzas de los datos, la descomponemos mediante

$$\hat{\mathbf{\Sigma}} = \mathbf{\Gamma}_2 \mathbf{\Lambda}_2 \mathbf{\Gamma}_2^{\mathrm{t}},$$

donde $\mathbf{\Gamma}_2$ tiene los autovectores asociados a los dos primeros autovalores en columnas y $\mathbf{\Lambda}_2$ es diagonal. Para la construcción del biplot de la figura 1.9 proyectamos $\mathbf{X}$ al espacio de dimensión 2 según $\mathbf{\Lambda}_2$, formando los puntos que representan a las observaciones, y luego proyectamos la matriz $\mathrm{diag}(3,3,3,3)$ con la misma transformación, para representar con flechas las marcas canónicas de tamaño 3. Se eligió esta escala para ajustar la relación entre el tamaño de los puntos y el tamaño de las flechas. La representación dada por el análisis de componentes principales y el biplot es tal que las distancias euclídeas de los *scores* se aproximan a las distancias euclídeas de los datos originales, mientras que los ángulos entre las flechas preservan las correlaciones entre las variables.

Biplot lineal de MDS

En este problema en particular puede ser interesante proponer una métrica distinta para las observaciones. Consideremos la distancia de Minkowski de orden k para $\mathbf{x}_i$ y $\mathbf{x}_j$ dada por

$$\left(\sum_{k=1}^{p} |X_{ik} - X_{jk}|^k\right)^{1/k}, \tag{1.24}$$

que de acuerdo con el orden k, produce distintas formas para las curvas de nivel de distancias. Ajustar esta distancia puede resultar de interés para problemas puntuales donde la distancia euclídea puede no ser la más adecuada. Para k=1 la distancia de Minkowski se llama distancia *city-block* o Manhattan, ya que mide las distancia de un punto a otro recorriendo una cuadrícula. Para k=2 se trata de la distancia euclídea clásica, mientras que para valores mayores de k se le da más importancia al mayor de las dos componentes en el cálculo de la distancia.

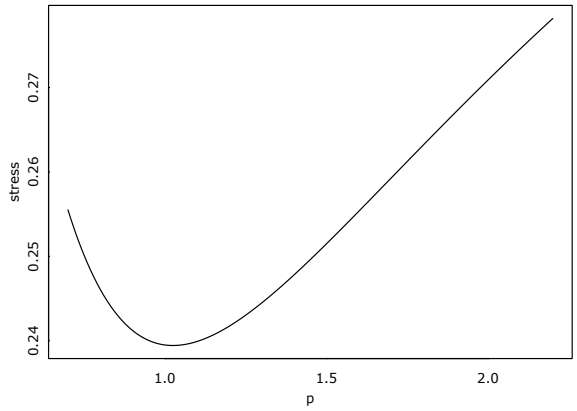


Figura 1.10: Ajuste de la potencia de Minkowski

En el gráfico 1.10 vemos los niveles de Stress-1 para distintas configuraciones de MDS que se obtienen variando el valor de k. El nivel óptimo para este problema es k = 1.02. Realizamos MDS con esta métrica y buscamos el biplot análogo al de la figura 1.9. Sea $\mathbf{Z}$ el escalado bidimensional obtenido, necesitamos tener direcciones de proyección para poder representar las marcas canónicas en $\mathbb{R}^2$. Llamamos $\mathbf{B}$ a la matriz de p filas y 2 columnas que tiene direcciones de proyección para los datos originales, buscamos una solución aproximada de

$$\mathbf{XB} = \mathbf{Z}$$

por cuadrados mínimos, obteniendo $\hat{\mathbf{B}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{Z}$. Los puntos del biplot corresponden a las coordenadas de $\mathbf{Z}$, y las flechas marcan los puntos resultantes de proyectar $(3, 3, 3, 3)$ según $\hat{\mathbf{B}}$.

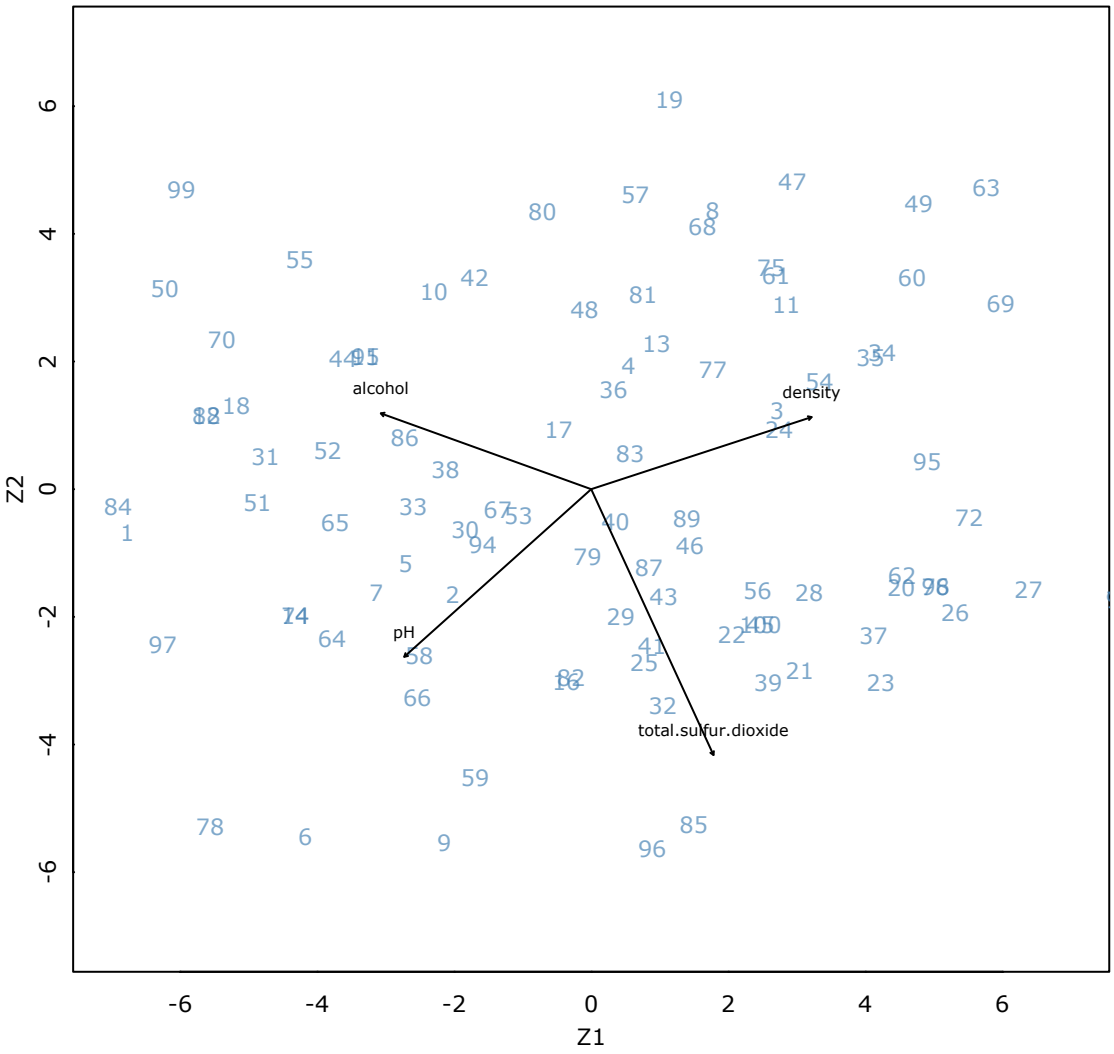


Figura 1.11: Biplot de MDS

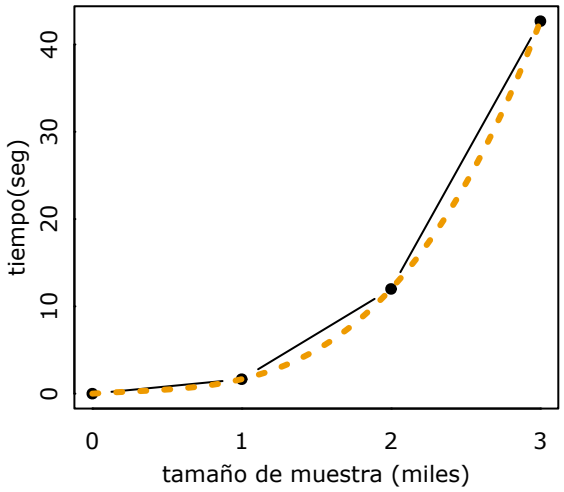
La representación de la figura 1.11 tiene un error de aproximación debido a la regresión lineal que se hizo al representar las marcas canónicas. Esto puede esconder estructuras no lineales en los datos. En [4] se sugieren estrategias para realizar biplots en MDS donde las marcas canónicas de las variables sean representadas mediante trayectorias no lineales. Uno de esos métodos se basa en incorporar los vectores canónicos $\mathbf{e}_k$ y sus múltiplos a la muestra original y realizar MDS con ellos.

Capítulo 2

Problemas con grandes datos

2.1. Introducción

Cuando el tamaño de la muestra aumenta, el escalado multidimensional se vuelve costoso con los algoritmos estándar porque requiere una gran capacidad de memoria y tiempo. Para el método CMDS, el tiempo de cómputo es $O(n^3)$ puesto que se requiere hacer la descomposición en autovalores y autovectores de una matriz de disimilaridades de $n \times n$. Si el problema también requiere el cálculo de esa matriz de disimilaridad, el requerimiento es aún mayor.



En el gráfico se muestran los tiempos medios de procesamiento con CMDS, en dimensión $k = 5$, para muestras de tamaño creciente, realizados con una MacBook Air con procesador Intel Core i5 y 8 GB de RAM. En naranja se muestra el polinomio

$$g(n) = 1.2n - 1.48n^2 + 1.94n^3$$

superpuesto a los puntos. Para una muestra de 50000 datos, el tiempo de procesamiento estimado aumenta a más de 2 días y medio. Esto limita en gran medida las aplicaciones de MDS. En este capítulo veremos estrategias para tratar este problema: primero, dos métodos basados en la idea de *división y conquista*; y luego dos métodos basados en interpolación. En todos los casos, partiremos de un grupo de objetos $\{O_1, O_2, ..., O_n\}$, con matriz de disimilaridades $\mathbf{\Delta}$, para los que se busca un escalado $\mathbf{Z}$ de k dimensiones.

2.2. Métodos de *división y conquista*

Un primer grupo de métodos, llamados *MDS de división y conquista* (DCO-MDS), se basan en dividir la muestra inicial en bloques pequeños y luego unir las distintas configuraciones en una solución única. El procedimiento es el siguiente:

- Elegir de la muestra inicial un grupo de $\mathtt{c}$ puntos de referencia, en inglés llamados *landmarks*, en forma aleatoria.
- Separar los puntos restantes en $\mathtt{p}$ bloques e incluir los puntos de referencia en cada uno. Quedarán conformados grupos de tamaño máximo $\mathtt{l}$, suficientemente pequeño como para realizar MDS de manera eficiente.
- Aplicar MDS a cada grupo por separado y obtener los escalados $\mathbf{Z_1}, \mathbf{Z_2}, ..., \mathbf{Z_p}$. En estas configuraciones se repetirá la solución para los $\mathtt{c}$ puntos de referencia, pero con distinta orientación y ubicación. Llamemos $\mathbf{Z_i^c}$ a esas soluciones para el i-ésimo grupo.
- Empezando por el segundo grupo, separar los puntos $\mathbf{Z_2^c}$ y encontrar la rotación, traslación y/o dilatación necesaria para que los vectores coincidan con los de $\mathbf{Z_1^c}$.
- Aplicar la transformación hallada a los puntos restantes de $\mathbf{Z_2}$. De esta forma se unen las primeras dos soluciones
- Repetir la operación para los grupos restantes $\mathbf{Z_3^c}, \mathbf{Z_4^c}, ..., \mathbf{Z_p^c}$, todos en relación al primero, y así obtener la solución final.

El mecanismo usado para alinear las distintas soluciones puede variar. En [3] utilizan transformaciones de Procrustes, mientras que en [9] se basan en la descomposición QR. Notemos que este algoritmo puede aplicarse para cualquier método de MDS, siempre que sea posible acoplar las soluciones de manera correcta.

La elección de $\mathtt{l}$ y de $\mathtt{c}$ afectará la eficiencia global del algoritmo. Además, el valor de $\mathtt{c}$ debe ser mayor o igual a la dimensión k deseada para el escalado. Aplicando este método para un valor de $\mathtt{l}$ prefijado, los tiempos de cómputo son O(n). Una ventaja importante de este método a nivel computacional es que permite *paralelizar* el proceso, ya que se pueden realizar los escalados de los distintos grupos en forma independiente.

2.2.1. Alineación con transformaciones de Procrustes

La estrategia usada en [3] para alinear las distintas soluciones consiste en encontrar una transformación óptima. Supongamos que tenemos dos nubes de puntos dadas por las matrices $\mathbf{A}$ y $\mathbf{B}$, ambas de $\mathbb{R}^{n\times p}$ con $n \geq p$ en el mismo espacio $\mathcal{R}$. Buscamos transformar $\mathbf{B}$ para que se parezca *lo más posible* a $\mathbf{A}$, punto a punto. En otras palabras, se quiere *deformar* una nube de puntos para que tome la forma de la otra lo más que se pueda, estableciendo las restricciones que sean necesarias. Este procedimiento se llama transformación de Procrustes,[1] y se aplica comunmente en procesamiento de imágenes.

[1] Citando a [2]: *Es probablemente el único método estadístico que lleva el nombre de un villano. Según la mitología griega, a cualquier viajero de la ruta de Eleusis a Atenas le esperaba una sorpresa si aceptaba hospedarse con Damastes, que tenía una posada al costado del camino. Si la cama era muy grande para los huéspedes, Damastes estiraba sus cuerpos hasta darles la forma correcta; si la cama era muy pequeña, les cortaba las extremidades. Por esto se le dio el nombre de ¨Procrustes¨, que significa ¨ensanchador¨. Eventualmente, Procrustes tuvo el mismo destino que sus huéspedes de la mano de Teseo.*

Un primer paso es proponer una transformación rígida: una rotación o reflexión. Para eso se deberá encontrar una matriz $\mathbf{T}$ ortogonal que cumpla

$$\mathbf{A} \approx \mathbf{BT}.$$

Buscaremos, entonces, minimizar $||\mathbf{A} - \mathbf{BT}||_{\mathrm{F}}^2$. Veamos que

$$\begin{aligned} ||\mathbf{A} - \mathbf{BT}||_{\mathrm{F}}^2 = \mathrm{tr}(\mathbf{A} - \mathbf{BT})^{\mathrm{t}}(\mathbf{A} - \mathbf{BT}) &= \mathrm{tr}\ \mathbf{A}^{\mathrm{t}}\mathbf{A} - \mathrm{tr}\ \mathbf{T}^{\mathrm{t}}\mathbf{B}^{\mathrm{t}}\mathbf{BT} - 2\mathrm{tr}\ \mathbf{A}^{\mathrm{t}}\mathbf{BT} \\ &= \mathrm{tr}\ \mathbf{A}^{\mathrm{t}}\mathbf{A} - \mathrm{tr}\ \mathbf{B}^{\mathrm{t}}\mathbf{B} - 2\mathrm{tr}\ \mathbf{A}^{\mathrm{t}}\mathbf{BT}. \end{aligned} \tag{2.1}$$

En el segundo término, permutamos los factores usando propiedades de la traza, luego tenemos $\mathbf{T}^{\mathrm{t}}\mathbf{T} = \mathbf{I}$ al ser ortogonal. Como los dos primeros términos no dependen de $\mathbf{T}$, los reducimos a una constante. El problema se reduce a minimizar

$$2\mathrm{tr}\ \mathbf{A}^{\mathrm{t}}\mathbf{BT}. \tag{2.2}$$

Teorema 2.1 (Desiguladad de Kristof)**.** Dada una matriz $\mathbf{Y}$ diagonal con valores no negativos y $\mathbf{R}$ una matriz ortogonal, se verifica

$$-\mathrm{tr}\ \mathbf{RY} \geq -\mathrm{tr}\ \mathbf{Y},$$

y la igualdad se alcanza cuando $\mathbf{R}$ es la identidad $\mathbf{I}$.

Demostración. Tenemos

$$-\mathrm{tr}\ \mathbf{RY} = -\sum_{\mathrm{i}} \mathrm{r_{ii}y_{ii}}$$

y como las columnas de $\mathbf{R}$ son ortogonales, se tiene que $\sum_{\mathrm{i}} \mathrm{r_{ii}^2} = 1$, de manera que $-1 \leq \mathrm{r_{ii}} \leq 1$ y $-\mathrm{r_{ii}y_{ii}} \geq -\mathrm{y_{ii}}$. La igualdad se da cuando $\mathrm{r_{ii}} = 1$, es decir cuando $\mathbf{R} = \mathbf{I}$.

□

Usemos la descomposición en valores singulares

$$\mathbf{A}^{\mathrm{t}}\mathbf{B} = \mathbf{L}\mathbf{\Phi}\mathbf{W}^{\mathrm{t}} \tag{2.3}$$

con $\mathbf{L}$ y $\mathbf{W}$ ortogonales completas y $\mathbf{\Phi}$ diagonal. Veamos que la función objetivo de 2.2 se puede reescribir como

$$\begin{aligned} -\mathrm{tr}\ \mathbf{A}^{\mathrm{t}}\mathbf{BT} &= -\mathrm{tr}\ \mathbf{L}\mathbf{\Phi}\mathbf{W}^{\mathrm{t}}\mathbf{T} \\ &= -\mathrm{tr}\ \mathbf{W}^{\mathrm{t}}\mathbf{TL}\mathbf{\Phi} \\ &\geq -\mathrm{tr}\mathbf{\Phi}. \end{aligned} \tag{2.4}$$

En esta desigualdad, la matriz $\mathbf{W}^{\mathrm{t}}\mathbf{TL}$ juega las veces de $\mathbf{R}$ del teorema 2.1 ya que es ortogonal, y la matriz $\mathbf{\Phi}$ juega las veces de $\mathbf{Y}$. Para minimizar debemos elegir $\mathbf{T}$ para que $\mathbf{W}^{\mathrm{t}}\mathbf{TL}$ sea la identidad. Es sencillo ver que esto se verifica si

$$\mathbf{T} = \mathbf{W}\mathbf{L}^{\mathrm{t}}. \tag{2.5}$$

Un segundo paso es agregar una traslación y un cambio de escala

$$\mathbf{A} \approx \mathrm{s}\mathbf{B}\mathbf{T} + \mathbf{1}_\mathrm{n}\mathbf{t}^\mathrm{t},$$

para buscar que los puntos de ambas configuraciones queden lo más superpuestos que se pueda. En esta expresión, el factor s es un cambio de escala de las coordenadas de $\mathbf{B}$, el vector $\mathbf{t}$ tiene las coordenadas de traslación para todos los puntos, y $\mathbf{1}$ es un vector columna de unos. Buscaremos ahora

$$\underset{\mathrm{s},\mathbf{T},\mathbf{t}}{\operatorname{argmin}} \; ||\mathbf{A} - \mathrm{s}\mathbf{B}\mathbf{T} - \mathbf{1}\mathbf{t}^\mathrm{t}||_\mathrm{F}^2. \tag{2.6}$$

La resolución del problema es sencilla, y puede encontrarse con detalle en el capítulo 20 de [1]. El óptimo se alcanza tomando $\mathrm{s}_\mathrm{opt} = \frac{\mathrm{tr}\ \mathbf{A}^\mathrm{t}\mathbf{H}\mathbf{B}\mathbf{T}}{\mathrm{tr}\ \mathbf{B}^\mathrm{t}\mathbf{H}\mathbf{B}}$, donde $\mathbf{H}$ es la matriz de centrado, $\mathbf{t}_\mathrm{opt} = \mathrm{n}^{-1}(\mathbf{A} - \mathrm{s}\mathbf{B}\mathbf{T})^\mathrm{t}\mathbf{1}_\mathrm{n}$ y $\mathbf{T}_\mathrm{opt}$ según 2.5.

Aplicación a MDS

- Los escalados que se obtienen del DCO-MDS no necesariamente estarán centrados en el origen.
- Los escalados paralelos del DCO-MDS pueden hacerse con cualquier método de MDS. Para escalado de distancias, puede ser conveniente usar la misma función de mapeo *f*.
- Aun utilizando CMDS como método base, las soluciones DCO-MDS no estarán anidadas. Esto significa que no alcanza con tomar las primeras $\mathrm{l} < \mathrm{k}$ columnas de un escalado de dimensión k para obtener un escalado de dimensión l, sino que debe realizarse el procedimiento nuevamente.
- La aplicación de este método puede limitar la dimensionalidad máxima en el caso de CMDS. La dimensión máxima de todo el conjunto será la cantidad de autovalores positivos mínima entre todos los bloques. Como la dimensión buscada habitualmente es 2 o 3, este aspecto puede no ser relevante. Para escalado de distancias esto no es un inconveniente.
- Las soluciones son invariantes ante transformaciones ortogonales y de escala de los distintos bloques.

Algoritmo 2 Transformación de Procrustes con traslación y dilatación

---

Entrada: Matriz objetivo $\mathbf{A}$ y matriz de partida $\mathbf{B}$, ambas de las mismas dimensiones.

1. Obtener la descomposición en valores singulares de $\mathbf{C} = \mathbf{A}^\mathrm{t}\mathbf{H}\mathbf{B} = \mathbf{L}\mathbf{\Phi}\mathbf{W}^\mathrm{t}$, donde $\mathbf{H}$ es la matriz de centrado de $\mathrm{n} \times \mathrm{n}$.
2. Obtener la matriz de rotación como $\mathbf{T} = \mathbf{W}\mathbf{L}^\mathrm{t}$.
3. Obtener el factor de dilatación como $\mathrm{s} = \frac{\mathrm{tr}\ \mathbf{A}^\mathrm{t}\mathbf{H}\mathbf{B}\mathbf{T}}{\mathrm{tr}\ \mathbf{B}^\mathrm{t}\mathbf{H}\mathbf{B}}$.
4. Obtener el vector de traslación como $\mathbf{t} = \mathrm{n}^{-1}(\mathbf{A} - \mathrm{s}\mathbf{B}\mathbf{T})^\mathrm{t}\mathbf{1}_\mathrm{n}$.
5. Calcular $\mathbf{B}^* = \mathrm{s}\mathbf{B}\mathbf{T} + \mathbf{1}_\mathrm{n}\mathbf{t}^\mathrm{t}$.

Salida: $\mathbf{B}^*$, coordenadas de $\mathbf{B}$ ajustadas a las coordenadas de $\mathbf{A}$.

2.2.2. Alineación con descomposición QR

Busquemos nuevamente una estrategia de alineación de dos matrices $\mathbf{A}$ y $\mathbf{B}$. Supongamos que estas dos matrices son las coordenadas de dos nubes de puntos que únicamente difieren por una traslación, rotación o reflexión. Llamaremos *matriz objetivo* a $\mathbf{A}$ y buscaremos transformar $\mathbf{B}$ para alinearla nuevamente con la primera. Sean $\overline{\mathbf{a}}$ y $\overline{\mathbf{b}}$ vectores con los promedios de las p columnas de cada matriz, obtenemos las matrices centradas

$$\begin{aligned} \mathbf{A}_{\mathrm{c}} &= \mathbf{A} - \mathbf{1}_{\mathrm{n}}\overline{\mathbf{a}}^{\mathrm{t}} \\ \mathbf{B}_{\mathrm{c}} &= \mathbf{B} - \mathbf{1}_{\mathrm{n}}\overline{\mathbf{b}}^{\mathrm{t}}. \end{aligned} \tag{2.7}$$

Podemos encontrar una descomposición QR de las *traspuestas* de estas matrices. Para $\mathbf{A}_{\mathrm{c}}$ tenemos

$$\mathbf{A}_{\mathrm{c}}^{\mathrm{t}} = \mathbf{QR}, \tag{2.8}$$

donde $\mathbf{Q}$ es una matriz de $\mathrm{p} \times \mathrm{p}$ cuyas columnas forman una base ortonormal de $\mathbb{R}^{\mathrm{p}}$, y $\mathbf{R}$ es una matriz de $\mathrm{p} \times \mathrm{n}$ que tiene los siguientes bloques:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \ ; \ \mathbf{R}_2 \end{bmatrix},$$

donde $\mathbf{R}_1$ es una matriz triangular superior de $\mathrm{p} \times \mathrm{p}$ y $\mathbf{R}_2$ es de $\mathrm{p} \times (\mathrm{n} - \mathrm{p})$. Se puede ver que:

- La primera columna de $\mathbf{Q}$, llamémosla $\mathbf{q}_1$, que es el primer vector de la base ortonormal de $\mathbb{R}^{\mathrm{p}}$, corresponde a la dirección dada por la *primera fila* de la configuración $\mathbf{A}_{\mathrm{c}}$, llamémosla $\mathbf{a}_1^{\mathrm{t}}$, escalada a norma 1.
- La primera columna de $\mathbf{R}_1$, llamemos $\mathbf{r}_1$ tiene en su primera coordenada el valor $||\mathbf{a}_1||$ y 0 en las coordenadas restantes.
- La columna $\mathbf{q}_2$ tiene un vector ortogonal al primero, elegido de manera que $\mathbf{a}_2 = \mathbf{q}_1\mathbf{q}_1^{\mathrm{t}}\mathbf{a}_2 + \mathbf{q}_2\mathbf{q}_2^{\mathrm{t}}\mathbf{a}_2$, es decir que $\mathbf{a}_2$ pertenezca al subespacio generado por $\mathbf{q}_1$ y $\mathbf{q}_2$.
- La columna $\mathbf{r}_2$ tiene módulo igual a $||\mathbf{q}_1^{\mathrm{t}}\mathbf{a}_2||$ en la primera coordenada, a $||\mathbf{q}_2^{\mathrm{t}}\mathbf{a}_2||$ en la segunda, y 0 en las restantes.
- Las columnas restantes $\mathbf{q}_3, \ \ldots \ , \mathbf{q}_{\mathrm{p}}$ completan sucesivamente el espacio $\mathbb{R}^{\mathrm{p}}$ añadiendo una por una las p primeras filas de $\mathbf{A}_{\mathrm{c}}$ con el mismo procedimiento que vimos, formando el bloque $\mathbf{R}_1$.
- Las columnas del bloque $\mathbf{R}_2$ son las coordenadas de las n-p filas restantes de $\mathbf{A}_{\mathrm{c}}$ en la base dada por las columnas de $\mathbf{Q}$.
- Para realizar esta descomposición, es necesario contar con al menos p filas linealmente independientes en $\mathbf{A}_{\mathrm{c}}$, es decir al menos p observaciones no repetidas en la matriz de datos.

Se puede ver que al descomponer

$$\mathbf{A}_{\mathrm{c}}^{\mathrm{t}} = \mathbf{Q}_{\mathrm{A}}\mathbf{R}_{\mathrm{A}} \quad \text{y} \quad \mathbf{B}_{\mathrm{c}}^{\mathrm{t}} = \mathbf{Q}_{\mathrm{B}}\mathbf{R}_{\mathrm{B}}$$

las matrices $\mathbf{R}_{\mathrm{A}}$ y $\mathbf{R}_{\mathrm{B}}$ *serán iguales*, salvo errores numéricos, y siempre que se preserven los signos; es decir que las sucesivas direcciones $\mathbf{q}_1, \mathbf{q}_2, \ldots$ tengan el mismo sentido en una descomposición y en la otra. Con esto encontramos la transformación que buscamos:

$$
\begin{aligned}
\mathbf{R}_{\mathrm{A}} &= \mathbf{R}_{\mathrm{B}} \\
\mathbf{Q}_{\mathrm{A}}^{\mathrm{t}}\mathbf{A}_{\mathrm{c}}^{\mathrm{t}} &= \mathbf{Q}_{\mathrm{B}}^{\mathrm{t}}\mathbf{B}_{\mathrm{c}}^{\mathrm{t}} \\
\mathbf{A}_{\mathrm{c}}\mathbf{Q}_{\mathrm{A}} &= \mathbf{B}_{\mathrm{c}}\mathbf{Q}_{\mathrm{B}} \\
\mathbf{A}_{\mathrm{c}} &= \mathbf{B}_{\mathrm{c}}\mathbf{Q}_{\mathrm{B}}\mathbf{Q}_{\mathrm{A}}^{\mathrm{t}} \\
\mathbf{A} - \mathbf{1}_{\mathrm{n}}\bar{\mathbf{a}}^{\mathrm{t}} &= (\mathbf{B} - \mathbf{1}_{\mathrm{n}}\bar{\mathbf{b}}^{\mathrm{t}})\mathbf{Q}_{\mathrm{B}}\mathbf{Q}_{\mathrm{A}}^{\mathrm{t}} \\
\mathbf{A} &= \mathbf{B}\mathbf{Q}_{\mathrm{B}}\mathbf{Q}_{\mathrm{A}}^{\mathrm{t}} + \mathbf{1}_{\mathrm{n}}(\bar{\mathbf{a}} - \mathbf{Q}_{\mathrm{B}}^{\mathrm{t}}\mathbf{Q}_{\mathrm{A}}\bar{\mathbf{b}})^{\mathrm{t}}.
\end{aligned}
\tag{2.9}
$$

Llamemos $\mathbf{T} = \mathbf{Q}_{\mathrm{B}}\mathbf{Q}_{\mathrm{A}}^{\mathrm{t}}$ a la matriz de transformación ortogonal de $\mathrm{p}\times\mathrm{p}$ y $\mathbf{t} = \bar{\mathbf{a}} – \mathbf{Q}_{\mathrm{B}}^{\mathrm{t}}\mathbf{Q}_{\mathrm{A}}\bar{\mathbf{b}}$ al vector de traslación de $\mathbb{R}^{\mathrm{p}}$. Este método no considera cambios de escala. Este aspecto puede disminuir su rendimiento en la aplicación MDS. Una solución posible a esto podría ser combinar la rotación y traslación dada por 2.9 con un una transformación de escala óptima según el método de Procrustes. Este procedimiento no se encontró en la bibliografía consultada.

Algoritmo 3 Alineación con descomposición QR

Entrada: Matriz objetivo $\mathbf{A}$ y matriz de partida $\mathbf{B}$, ambas de las mismas dimensiones $\mathrm{n}\times\mathrm{p}$.

1. Centrar por columnas y trasponer las matrices $\mathbf{A}$ y $\mathbf{B}$ para obtener $\mathbf{A}_{\mathrm{c}}^{\mathrm{t}}$ y $\mathbf{B}_{\mathrm{c}}^{\mathrm{t}}$.
2. Realizar la descomposición QR de cada una obteniendo $\mathbf{Q}_{\mathrm{A}}$ y $\mathbf{Q}_{\mathrm{B}}$ de $\mathrm{p}\times\mathrm{p}$, garantizando que coincidan los signos de las direcciones dadas por sus columnas.
3. Obtener la matriz de rotación como $\mathbf{T} = \mathbf{Q}_{\mathrm{B}}\mathbf{Q}_{\mathrm{A}}^{\mathrm{t}}$.
4. Obtener el vector de traslación como $\mathbf{t} = \bar{\mathbf{a}} – \mathbf{Q}_{\mathrm{B}}^{\mathrm{t}}\mathbf{Q}_{\mathrm{A}}\bar{\mathbf{b}}$.
5. Calcular $\mathbf{B}^{*} = \mathbf{B}\mathbf{T} + \mathbf{1}_{\mathrm{n}}\mathbf{t}^{\mathrm{t}}$.

Salida: $\mathbf{B}^{*}$, coordenadas de $\mathbf{B}$ alineadas con las de $\mathbf{A}$

Ejemplo (Simulación)**.** Se simuló una configuración de puntos $\mathbf{X}$ de 10 observaciones con 2 variables provenientes de una distribución Normal. Se calculó la matriz de distancias euclídeas y se sumó a cada componente un error aleatorio con distribución $\mathcal{U}[0,1]$ obteniendo

$$
\mathbf{\Delta} = \begin{pmatrix}
0 & & & & & & & & & \\
2.07 & 0 & & & & & & & & \\
1.85 & 1.69 & 0 & & & & & & & \\
2.47 & 2.97 & 2.27 & 0 & & & & & & \\
3.72 & 2.26 & 3.51 & 5.82 & 0 & & & & & \\
2.12 & 1.81 & 2.05 & 3.6 & 2.87 & 0 & & & & \\
3.4 & 3.98 & 3.81 & 4.28 & 4.67 & 2.82 & 0 & & & \\
3.38 & 5.21 & 4.79 & 3 & 7.15 & 4.76 & 3.46 & 0 & & \\
2.23 & 0.38 & 1.87 & 3.24 & 2.2 & 1.05 & 4.22 & 5.52 & 0 & \\
3.93 & 3.48 & 3.15 & 3.01 & 5.39 & 4.6 & 6.72 & 5.56 & 4 & 0
\end{pmatrix},
$$

que se toma como una matriz de disimilaridades para MDS.

Los autovalores de $\mathbf{B} = -\frac{1}{2}\mathbf{H}\boldsymbol{\Delta}^2\mathbf{H}$ son 35.21, 26.17, 3.47, 2.04, 1.29, 1.22, 0.00, –0.67, –1.92 y– 3.08. Para realizar DCO-MDS se toman $\mathtt{c} = 4$ puntos de referencia superpuestos con los restantes en dos grupos de tamaño $\mathtt{l} = 7$, para los que se obtienen los escalados $\mathbf{Z}_1$ y $\mathbf{Z}_2$ con CMDS que se muestran a continuación:

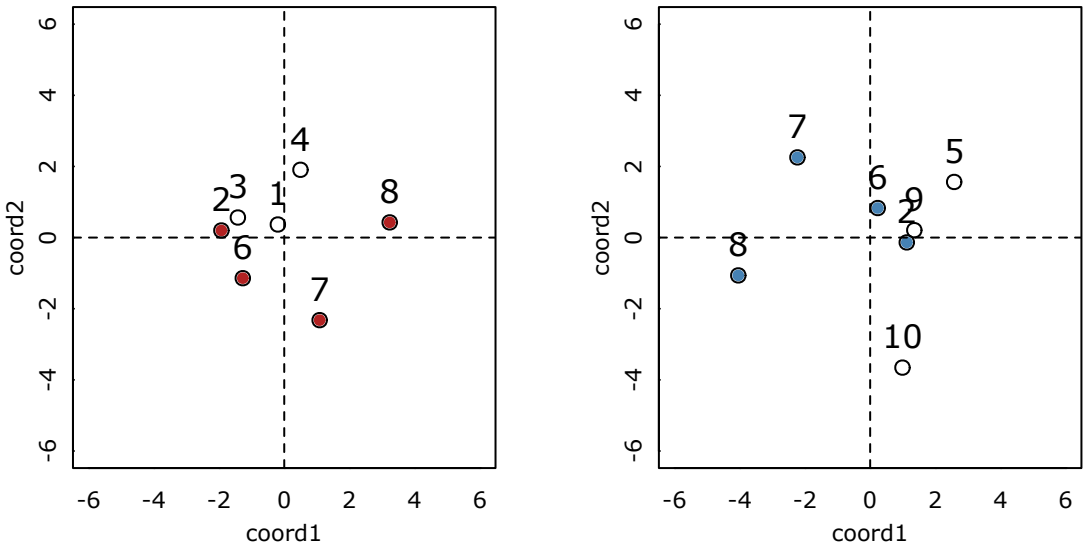


Figura 2.1: División de grupos para DCO-MDS y los *landmarks* en cada grupo

Se busca transformar $\mathbf{Z}_2$ según la alineación óptima que se obtiene de los *landmarks*. Con el método de Procrustes se tiene

$$0.9732\ \mathbf{Z}_2 \begin{pmatrix} -0.9948 & 0.1016 \\ -0.1016 & -0.9948 \end{pmatrix} + \mathbf{1}(-0.8644\ \ -0.1314),$$

y con el método de la descomposición QR la transformación es

$$\mathbf{Z}_2 \begin{pmatrix} -0.9995 & 0.0309 \\ -0.0309 & -0.9995 \end{pmatrix} + \mathbf{1}(-0.9349\ \ -0.20033).$$

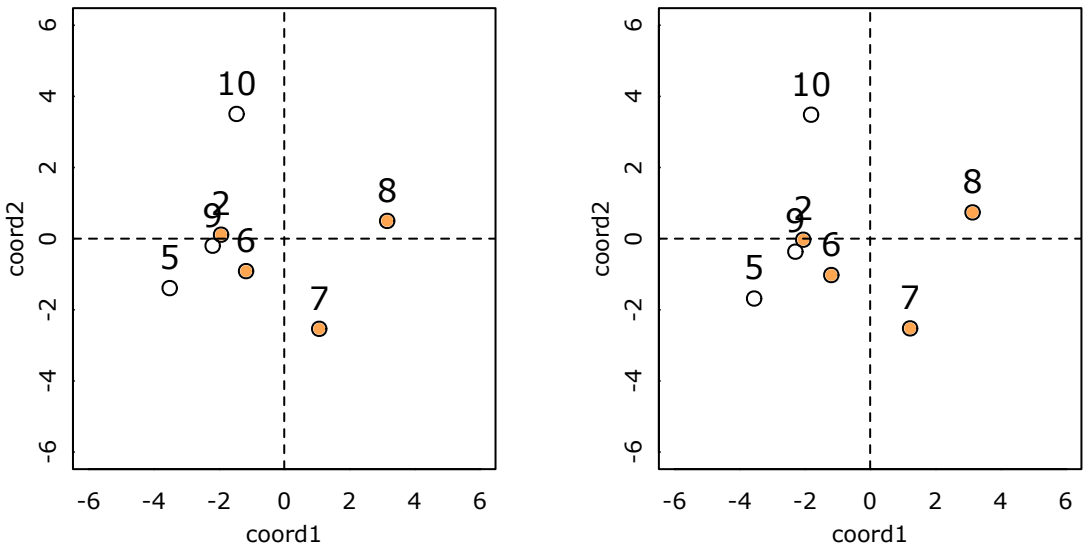


Figura 2.2: Puntos transformados en relación a $\mathbf{Z}_1^c$ (Procrustes a la izquierda, QR a la derecha)

En los gráficos se observan los resultados de transformar $\mathbf{Z}_2$. Finalmente, se obtiene el escalado de $\boldsymbol{\Delta}$ con CMDS, llamemos $\mathbf{Z}_{\text{cmds}}$ y se lo superpone a las configuraciones finales de cada método mediante Procrustes, para obtener:

://campus.exactas.uba.ar/cs[Procrustes]Images/fig2-5.pdf

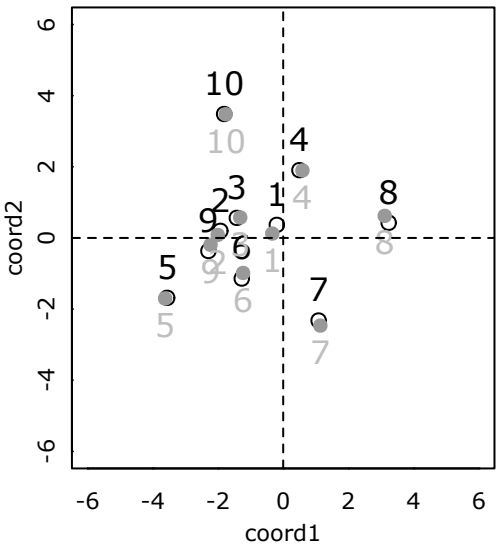


Figura 2.3: DCO-MDS (sin relleno) y CMDS (en gris). A la izquierda Procrustes y a la derecha QR.

Para el método de Procrustes se tiene una pérdida $||\mathbf{Z}_{\text{dco}} - \mathbf{Z}_{\text{cmds}}||_{\text{F}}^2 = 0.2747218$, mientras que para el método QR se tiene $||\mathbf{Z}_{\text{dco}} - \mathbf{Z}_{\text{cmds}}||_{\text{F}}^2 = 0.261428$.

2.3. Métodos basados en interpolación

Supongamos un nuevo grupo de objetos $\{O_{n+1}, O_{n+2}, ..., O_{n+m}\}$ que queremos incluir en un escalado k-dimensional ya existente para un primer grupo de puntos. Para eso se tienen las disimilaridades $\mathbf{\Delta}_{21}$ entre los elementos del segundo grupo y los del primero. El problema de *interpolación* consiste en encontrar coordenadas $\mathbf{Z}_2$ agregando los nuevos puntos, en bloques o de a uno por vez, y así representar las disimilaridades dadas. En los problemas de grandes datos, puede usarse esta idea como sigue:

- Elegir al azar un grupo de `m` puntos de referencia de la muestra inicial.
- Dividir los puntos restantes en `p` grupos de tamaño máximo `m`.
- Aplicar MDS en el primer grupo y obtener una configuración $\mathbf{Z}_1$ con la dimensionalidad k deseada en un espacio $\mathcal{Z}$.
- Para los bloques siguientes, uno a uno, interpolar los puntos a la primera solución en $\mathcal{Z}$, obteniendo sucesivamente $\mathbf{Z}_2, \mathbf{Z}_3, ..., \mathbf{Z}_p$. El escalado final es la superposición de todos los bloques. Según el método usado, puede no ser posible interpolar en bloques y deberán incorporarse los nuevos puntos de a uno, es decir habrá $p = n - m$ grupos de tamaño 1.

En [3] proponen utilizar un método de interpolación dado por Gower, que consiste en una descomposición de las distancias. Por otro lado, en [7] y en [5] usan métodos de optimización similares a los que se usan en el escalado de distancias visto en el la sección 1.3. Notemos que esta metodología también permite paralelizar el proceso, ya que la interpolación de los bloques se hace en relación al grupo original.

2.3.1. Interpolación de Gower

Como primer método, mostramos una serie de resultados que se encuentran en los apéndices 5, 6 y 7 de [4], y que se sugieren en [3] como estrategia para realizar MDS. Mostraremos un método para interpolar un punto y un bloque de puntos externos a una configuración ya existente, a partir de una descomposición de la distancia euclídea similar al ANOVA clásico.

Interpolación de un punto

Supongamos una configuración de n objetos dada por la matriz $\mathbf{A}$, en un espacio k-dimensional $\mathcal{S}$. Se incorpora un nuevo objeto y se buscan sus coordenadas a partir de las distancias al cuadrado respecto de los elementos de $\mathbf{A}$ dadas por

$$\mathbf{d}^2_{n+1} = \left(d^2_{n+1,1}\ d^2_{n+1,2}\ ...\ d^2_{n+1,n}\right)^t.$$

Para agregar el nuevo objeto, será necesaria una dimensión más en el espacio $\mathcal{S}$ para ajustar las distancias. Sea $\left(b_1,\ b_2,\ ...\ b_k,\ b_{k+1}\right)^t = \left(\mathbf{b},\ b_{k+1}\right)^t$ el vector buscado y además impongamos $a_{j_{k+1}} = 0$ para las nuevas coordenadas de todos los vectores de $\mathbf{A}$, tenemos

$$d^2_{n+1,j} = \sum_{c=1}^{k+1} (b_c - a_{j_c})^2 = \sum_{c=1}^{k+1} b_c^2 + \sum_{c=1}^{k} a^2_{j_c} - 2\sum_{c=1}^{k} b_c\ a_{j_c} \tag{2.10}$$

Veamos que si los puntos de la configuración **A** están centrados en el origen, el primer y el segundo término de 2.10 son las distancias de $(\mathbf{b},\ b_{k+1})^t$ y $\mathbf{a}_j$ al origen, respectivamente. Podemos reescribir el primer término (ver [4]) como

$$\sum_{c=1}^{k+1} b_c^2 = \frac{1}{n}\mathbf{d}_{n+1}^2\mathbf{1} - \frac{1}{2n^2}\mathbf{1}^t\mathbf{D}_A^{(2)}\mathbf{1}, \tag{2.11}$$

y el segundo término como

$$\sum_{c=1}^{k} {a_{j_c}}^2 = \overline{d_{j\bullet}^2} - \frac{1}{2n^2}\mathbf{1}^t\mathbf{D}_A^{(2)}\mathbf{1}, \tag{2.12}$$

donde $\overline{d_{j\bullet}^2}$ es el promedio de las distancias al cuadrado de $\mathbf{a}_j$ a los restantes puntos de **A**. Observemos que $\frac{1}{n^2}\mathbf{1}^t\mathbf{D}_A^{(2)}\mathbf{1}$ y $\frac{1}{n}\mathbf{d}_{n+1}^2\mathbf{1}$ son escalares: El promedio de los elementos de la matriz $\mathbf{D}_A^{(2)}$ y de los elementos de $d_{n+1,j}^2$ respectivamente. Usando estas expresiones, podemos escribir el vector $\mathbf{d}_{n+1}^2$ completo como

$$\mathbf{d}_{n+1}^2 = \frac{1}{n}\mathbf{1}\mathbf{1}^t\mathbf{d}_{n+1}^2 + \frac{1}{n}\mathbf{D}_A^{(2)}\mathbf{1} - \frac{1}{n^2}\big(\mathbf{1}^t\mathbf{D}_A^{(2)}\mathbf{1}\big)\mathbf{1} - 2\mathbf{A}\mathbf{b}. \tag{2.13}$$

De esta expresión se pueden obtener las coordenadas **b**, o sea las primeras k coordenadas del nuevo vector, como:

$$\mathbf{b} = \frac{1}{2}\big(\mathbf{A}^t\mathbf{A}\big)^{-1}\mathbf{A}^t\Big[\frac{1}{n}\mathbf{D}_A^{(2)}\mathbf{1} - \frac{1}{n^2}\big(\mathbf{1}^t\mathbf{D}_A^{(2)}\mathbf{1}\big)\mathbf{1} - \mathbf{H}\mathbf{d}_{n+1}^2\Big]. \tag{2.14}$$

Donde **H** es la matriz de centrado de n × n. A partir de esto también se puede encontrar la componente b_{k+1}, que no se utilizó en la aplicación de MDS en la bibliografía consultada (el desarrollo puede verse en [4]).

Interpolación de un bloque de puntos

Busquemos ahora la configuración **B** correspondiente al bloque completo de m puntos, que tendrá ahora dimensión k + m. Partimos de una matriz de distancias de (n+m) × (n+m), en bloques como sigue

$$\mathbf{D}^{(2)} = \begin{bmatrix} \mathbf{D}_A^{(2)} & \mathbf{D}_{AB}^{(2)} \\ \mathbf{D}_{BA}^{(2)} & \mathbf{D}_B^{(2)} \end{bmatrix}.$$

Sea

$$\mathbf{B} = \big[\mathbf{B}_1\ \ \mathbf{B}_2\big],$$

donde $\mathbf{B}_1$ es la matriz de m × k con las coordenadas de los nuevos objetos en el espacio $\mathcal{S}$ con la dimensión original, y $\mathbf{B}_2$ a la matriz de m × m con las dimensiones nuevas. Además, se agregará a la configuración **A** ya existente un bloque de n × m con ceros. Con un razonamiento análogo al que ya se mostró para la interpolación de un punto, se puede obtener

$$\mathbf{B}_1 = \frac{1}{2}\Big[\mathbf{1}_m\mathbf{1}_n^t\Big(\frac{1}{n}\mathbf{D}_A^{(2)} - \frac{1}{n^2}\mathbf{1}_n{}^t\mathbf{D}_A^{(2)}\mathbf{1}_n\Big) - \mathbf{D}_{BA}^{(2)}\mathbf{H}\Big]\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-1}, \tag{2.15}$$

donde $\mathbf{H}$ es la matriz de centrado de n $\times$ n. Esta expresión equivale a lo que se llama *fórmula de interpolación de Gower* en [3]. Las coordenadas del bloque $\mathbf{B}_2$ se pueden encontrar en el correspondiente apéndice de [4].

Aplicación en MDS

- Para aplicar el método de Gower, se realizará MDS con un primer bloque de puntos de referencia, y se deberán interpolar los restantes. Esto puede hacerse punto a punto con 2.14, o en bloques con 2.15.
- En [3] no se menciona cómo influye la dimensión adicional que se obtiene con la interpolación en el ajuste de distancias.
- Como las fórmulas de Gower se obtienen para distancias euclídeas de los elementos del nuevo bloque, se las deberá reemplazar por $\boldsymbol{\Delta}^2_{\mathrm{BA}}$ para MDS, induciendo errores adicionales. No se estudió la influencia que tiene en los resultados el uso de disimilaridades particulares.

Algoritmo 4 Interpolación de disimilaridades con la fórmula de Gower

Entrada: Configuración existente $\mathbf{A}$ en $\mathbb{R}^{\mathrm{n}\times\mathrm{k}}$, sus distancias euclídeas $\mathbf{D}^{(2)}_{\mathrm{A}}$ y las disimilaridades del nuevo grupo $\boldsymbol{\Delta}^{(2)}_{\mathrm{BA}}$ de m $\times$ n.

1. Aplicar 2.15 reemplazando $\mathbf{D}^{(2)}_{\mathrm{BA}}$ por $\boldsymbol{\Delta}^{(2)}_{\mathrm{BA}}$.

Salida: Configuración $\mathbf{B}$ en $\mathbb{R}^{\mathrm{m}\times\mathrm{k}}$.

Ejemplo (Simulación - *continuación*)**.** Continuamos el ejemplo mostrado para DCO-MDS pero aplicando interpolación. Se toman los mismos 4 puntos como bloque inicial y se aplica la fórmula de Gower para los restantes, según el algoritmo 4. La configuración obtenida, en superposición con $\mathbf{Z}_{\text{cmds}}$ es

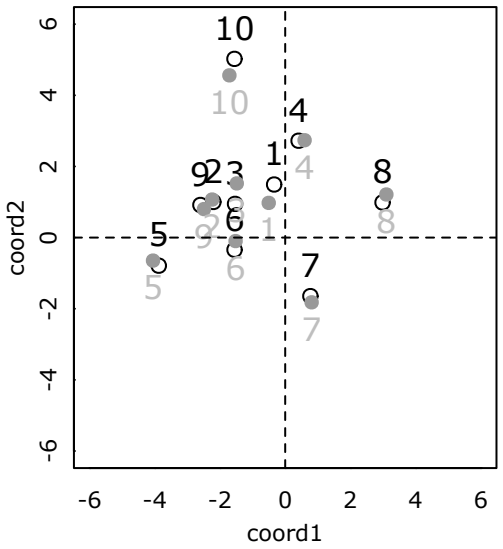


Figura 2.4: INTERP-MDS (sin relleno) y CMDS (en gris)

con una pérdida $||\mathbf{Z}_{\text{interp}} - \mathbf{Z}_{\text{cmds}}||_{\text{F}}^2 = 1.140324$.

2.3.2. Minimización del *Stress adicional*

Supongamos un grupo de puntos para los que se tiene un escalado k-dimensional en un espacio $\mathcal{Z}$, y se agregan nuevos puntos $\text{O}_{\text{n+1}}, \text{O}_{\text{n+1}}, ..., \text{O}_{\text{n+m}}$ para los que se tienen las disimilaridades δ_{ij} entre ellos y con los puntos originales. Al *Stress* del escalado original se adiciona

$$\tau = \tau_1 + \tau_2 = \sum_{\text{i=1}}^{\text{n}} \sum_{\text{j=n+1}}^{\text{n+m}} \left(\hat{\text{f}}(\delta_{\text{ij}}) - \text{d}_{\text{ij}}\right)^2 + \sum_{\text{i=n+1}}^{\text{n+m}} \sum_{\text{j=n+1}}^{\text{n+m}} \left(\hat{\text{f}}(\delta_{\text{ij}}) - \text{d}_{\text{ij}}\right)^2, \tag{2.16}$$

donde $\hat{\text{f}}$ es la función de mapeo previamente ajustada en el escalado inicial y d_{ij} son las distancias euclídeas de los nuevos puntos en $\mathcal{Z}$. Con esto, el problema de interpolación se puede pensar como la minimización de τ. La optimización puede hacerse de manera secuencial añadiendo los nuevos puntos de a uno, por lo que sólo se minimizará su término correspondiente en τ_2. En [7] se usan dos algoritmos de optimización específicos: las rutinas PORT (Gay, 1990) y el método BFGS (Broyden 1970, Fletcher 1970, Goldfarb 1970, Shanno 1970). En [5] se propone, además, un método basado en redes neuronales.

Capítulo 3

Implementación y simulación

3.1. Introducción

En este capítulo detallamos el estudio de simulación realizado con los métodos de MDS para grandes datos. Se busca evaluar los tiempos de procesamiento para distintos tamaños de muestra, analizar la calidad de las soluciones y ver si los métodos son sensibles a la presencia de puntos atípicos. Se analizan los métodos:

- `proc`, DCO-MDS con Procrustes según el algoritmo 2.
- `qr`, DCO-MDS con QR según el algoritmo 3.
- `gow`, INTERP-MDS con la fórmula de Gower de 2.15.

Se compara el desempeño en relación al escalado clásico (CMDS).

3.1.1. Implementación en `R`

En primer lugar, se implementaron los métodos rápidos en `R`. Tomando como base el código disponible en [3], se realizó una adaptación y replanteo del programa para incluir los otros métodos, entre otras mejoras. El trabajo de programación consistió en optimizar los tiempos de cómputo, con resultados similares a las implementaciones de los trabajos originales. Para los hiperparámetros de cada método se tomaron valores razonables de acuerdo con estos trabajos, sin evaluar en profundidad el efecto sobre la solución final. Algunos de los resultados mostrados en este capítulo pueden variar si estos parámetros se optimizan adecuadamente. Se tiene:

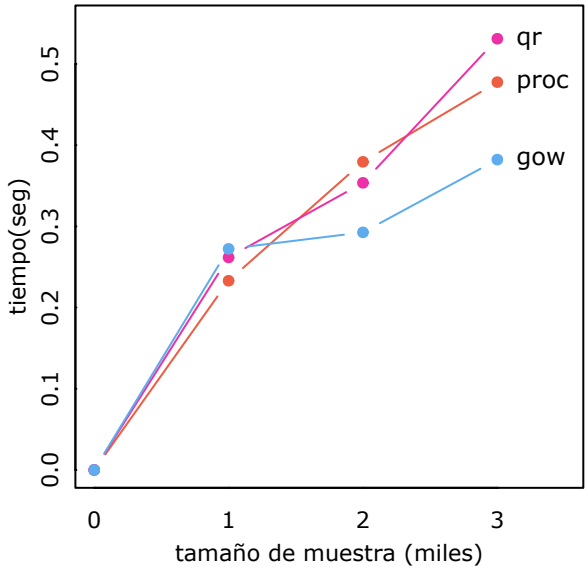
- La cantidad `c` de puntos de referencia para los métodos `proc` y `qr`. Se tomará `c=2k`, donde `k` es la dimensión del escalado. Esto se aplica en todos los casos, y está en línea con lo propuesto en [3].
- El tamaño `l` de los bloques en los métodos `proc` y `qr`. Se toma `l=400`, en línea con lo sugerido en [3].
- El tamaño `m` del primer bloque de puntos para los métodos de interpolación. Se toma `m=400` en línea con lo realizado en [7].

Los códigos de la simulación, así como los ejemplos mostrados en los capítulos anteriores, se encuentran en el repositorio `https://github.com/pcosatto/tesis_mds`. En la sección final de este capítulo hay una referencia más detallada del contenido de los archivos del repositorio.

Entre ellos, la función contenida en `cmdscaling.R` es la que ejecuta los escalados rápidos para las simulaciones de prueba. Para todas la simulaciones se utilizó una MacBook Air con procesador Intel Core i5 y 8 GB de RAM, con 4 núcleos de la CPU trabajando en simultáneo. Este último aspecto es importante ya que el código de `cmdscaling.R` admite paralelización de los escalados de los distintos bloques. El usuario puede especificar la cantidad de núcleos a utilizar con la variable `n_cores`, a través de la librería `parallel`. Esta función no puede utilizarse en Windows.

3.2. Estudio de simulación

En una simulación preliminar, se generan nubes de puntos con distribución Normal de dimensión `p=2` y se producen escalados de dimensión `k=2` a partir de las distancias euclídeas. Las componentes de la muestra son independientes con media 0 y varianza 5. Se realizan `Nrep=10` repeticiones para cada método y se toman los tiempos promedio:



	n=1000	n=2000	n=3000
cmds	1.66	12.00	42.67
proc	0.23	0.38	0.48
qr	0.26	0.35	0.53
gow	0.27	0.29	0.38

Tabla 3.1: Tiempos promedio para `Nrep=10` repeticiones.

A simple vista los métodos rápidos producen un ahorro muy considerable de tiempo. Para 3000 datos, se obtienen resultados en aproximadamente un 1 % del tiempo que demoraría el escalado clásico. Para muestras mayores, esta brecha podría extenderse aún más.

3.2.1. Simulación principal

Buscamos ver qué capacidad tienen los métodos rápidos para reproducir la solución CMDS en forma correcta y cómo responden con muestras de tamaño mayor.

Planteo del modelo

Sea $\mathbf{X}$ la matriz de coordenadas de una nube de `n=1000` puntos en $\mathbb{R}^{\mathrm{d}}$. Supongamos que a cada vector le agregamos una serie de coordenadas adicionales para extender el espacio a dimensión `p` > `d`, resultando en un nuevo vector

$$\mathbf{y} = (x_1, x_2, ..., x_{\mathrm{d}}, \varepsilon_1, \varepsilon_2, ..., \varepsilon_{\mathtt{p-d}})^{\mathrm{t}} = (\boldsymbol{x}\ \mathbf{e})^{\mathrm{t}},$$

donde las componentes de $\boldsymbol{x}$ son las coordenadas preexistentes y las componentes de $\mathbf{e}$ son variables aleatorias con distribución $\mathcal{N}(0,1)$ independientes entre sí y de las restantes. Tendremos que

$$\mathbf{Y} = \left[\begin{array}{cc} \mathbf{X} & \mathbf{E} \end{array}\right] \in \mathbb{R}^{\mathtt{n}\times\mathtt{p}} \tag{3.1}$$

es la matriz de coordenadas final de los `n` puntos, donde $\mathbf{E}$ es el bloque que tiene las nuevas dimensiones. Supongamos que se desea realizar un escalado multidimensional de los objetos cuyas coordenadas *verdaderas* están en $\mathbf{X}$, pero partiendo de la medición de $\mathbf{D_Y}$, la matriz de distancias euclídeas de $\mathbf{Y}$, como disimilaridad inicial. Para estas distancias tendremos un sesgo positivo ya que, al agregar dimensiones, los puntos se extienden a un espacio mayor. Sea $\mathrm{d}_{\mathrm{ij}}^2$ un elemento de $\mathbf{D}_{\mathbf{Y}}^{(2)}$, bajo 3.1 tenemos

$$\mathrm{d}_{\mathrm{ij}}^2 = \sum_{\mathrm{a}=1}^{\mathtt{p}} (\mathrm{Y}_{\mathrm{ai}} - \mathrm{Y}_{\mathrm{aj}})^2 = \sum_{\mathrm{a}=1}^{\mathtt{d}} (x_{\mathrm{ai}} - x_{\mathrm{aj}})^2 + \sum_{\mathrm{a}=\mathtt{d}+1}^{\mathtt{p}} (\varepsilon_{\mathrm{ai}} - \varepsilon_{\mathrm{aj}})^2, \tag{3.2}$$

por lo que

$$\mathbf{D}_{\mathbf{Y}}^{(2)} = \mathbf{D}_{\mathbf{X}}^{(2)} + \mathbf{D}_{\mathbf{E}}^{(2)}. \tag{3.3}$$

Veamos qué ocurre con los elementos de la matriz $\mathbf{B_Y} = \mathbf{Y}\mathbf{Y}^{\mathrm{t}}$, que son las normas al cuadrado $\mathrm{b}_{\mathrm{ii}} = \|\mathbf{y}_{\mathrm{i}}\|^2$ en la diagonal principal , y los productos internos $\mathrm{b}_{\mathrm{ij}} = \mathbf{y}_{\mathrm{i}}^{\mathrm{t}}\mathbf{y}_{\mathrm{j}}$ fuera de ella. En cualquier caso tenemos

$$\mathrm{b}_{\mathrm{ij}} = \mathbf{y}_{\mathrm{i}}^{\mathrm{t}}\mathbf{y}_{\mathrm{j}} = \sum_{\mathrm{a}=1}^{\mathtt{p}} \mathrm{Y}_{\mathrm{ai}}\mathrm{Y}_{\mathrm{aj}} = \sum_{\mathrm{a}=1}^{\mathtt{d}} x_{\mathrm{ai}}x_{\mathrm{aj}} + \sum_{\mathrm{a}=\mathtt{d}+1}^{\mathtt{p}} \varepsilon_{\mathrm{ai}}\varepsilon_{\mathrm{aj}} = \boldsymbol{x}_{\mathrm{i}}^{\mathrm{t}}\boldsymbol{x}_{\mathrm{j}} + \mathbf{e}_{\mathrm{i}}^{\mathrm{t}}\mathbf{e}_{\mathrm{j}}, \tag{3.4}$$

que matricialmente se expresa como

$$\mathbf{B_Y} = \mathbf{Y}\mathbf{Y}^{\mathrm{t}} = \mathbf{X}\mathbf{X}^{\mathrm{t}} + \mathbf{E}\mathbf{E}^{\mathrm{t}} = \mathbf{B_X} + \mathbf{B_E}. \tag{3.5}$$

La componente $\mathbf{B_X}$ es fija y la componente $\mathbf{B_E}$ es aleatoria. Para el elemento $\{\mathbf{B_E}\}_{\mathrm{ij}}$ tenemos

$$\mathsf{E}(\{\mathbf{B_E}\}_{\mathrm{ij}}) = \mathsf{E}(\mathbf{e}_{\mathrm{i}}^{\mathrm{t}}\mathbf{e}_{\mathrm{j}}) = \mathrm{tr}\big(\mathsf{E}(\mathbf{e}_{\mathrm{i}}^{\mathrm{t}}\mathbf{e}_{\mathrm{j}})\big) = \mathsf{E}\big(\mathrm{tr}(\mathbf{e}_{\mathrm{i}}^{\mathrm{t}}\mathbf{e}_{\mathrm{j}})\big) = \mathsf{E}\big(\mathrm{tr}(\mathbf{e}_{\mathrm{i}}\mathbf{e}_{\mathrm{j}}^{\mathrm{t}})\big) = \mathrm{tr}\big(\mathsf{E}(\mathbf{e}_{\mathrm{i}}\mathbf{e}_{\mathrm{j}}^{\mathrm{t}})\big). \tag{3.6}$$

Resulta que $\mathrm{tr}\big(\mathsf{E}(\mathbf{e}_{\mathrm{s}}\mathbf{e}_{\mathrm{r}}^{\mathrm{t}})\big) = 0$ para $\mathrm{s} \neq \mathrm{r}$, ya que los vectores son independientes; y $\mathrm{tr}\big(\mathsf{E}(\mathbf{e}_{\mathrm{s}}\mathbf{e}_{\mathrm{r}}^{\mathrm{t}})\big) = \mathrm{tr}\big(\Sigma_{\mathbf{e}}\big) = \mathtt{p-d}$ para $\mathrm{s} = \mathrm{r}$, ya que las nuevas componentes tienen varianza unitaria.

También se puede mostrar que $\mathsf{Var}(\mathbf{e}_i^t\mathbf{e}_j)$ vale `p-d` para i $\neq$ j y vale 2(`p-d`) para i = j. Todo esto sugiere que el planteo de 3.1 sirve como un modelo de errores de medición para los productos internos entre los pares de objetos en $\mathbf{X}$, ya que se les agrega una distorsión aleatoria positiva o negativa, mientras que las normas al cuadrado aumentan a medida que se agregan dimensiones. Esto lo hace especialmente útil para estudiar la respuesta de los métodos basados en CMDS, ya que el ajuste se produce minimizando la pérdida en los productos internos y no en las distancias.

En este trabajo prefijamos `p=10` y `d=5`, y generamos la matriz $\mathbf{X}$ con distribución Normal de media 0 y varianza diag$(5, 5, ..., 5)$, con una semilla fija en `R`. A modo de ejemplo, si tenemos `n=1000` vectores queda:

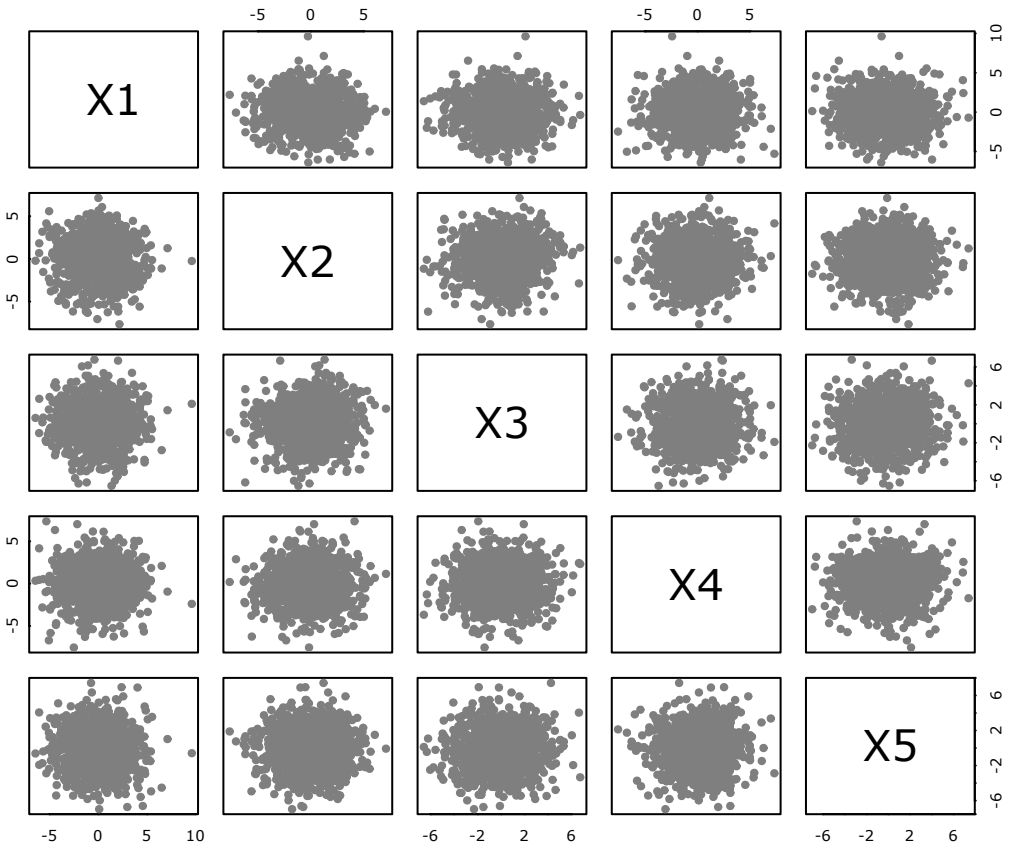


Figura 3.1: Vistas canónicas de nube de `n=1000` puntos sin distorsión.

Para estos puntos tenemos una *variabilidad total* dada por $\text{tr}(\mathbf{X}^t\mathbf{X}) = 24841.93$, mientras que para $\text{tr}(\mathbf{E}^t\mathbf{E})$ el valor esperado es

$$\mathsf{E}(\text{tr}(\mathbf{E}^t\mathbf{E})) = \text{tr}(\mathsf{E}(\mathbf{E}\mathbf{E}^t)) = \text{tr}(\text{diag}(5, 5, ..., 5)) = 5000, \tag{3.7}$$

lo que representa aproximadamente la quinta parte.

Identificación de la dimensión verdadera

Una primera corrida de simulaciones se realizan con tamaño de muestra `n` igual a 1000, como se mostró en el gráfico 3.1, agregándole los errores. En cada simulación se obtiene un escalado multidimensional $\mathbf{Z}$ tomando como punto de partida la información de $\mathbf{D_Y}$ y se calculan las siguientes métricas

$$\sigma_{B_Z} = \frac{||\mathbf{B_X} - \mathbf{Z}\mathbf{Z}^T||_F}{||\mathbf{B_X}||_F}, \tag{3.8}$$

$$\sigma_{D_Z} = \frac{||\mathbf{D_X} - \mathbf{D_Z}||_F}{||\mathbf{D_Z}||_F}, \tag{3.9}$$

$$\sigma_Z = \frac{||\mathbf{Z}_{cmds} - \mathbf{Z}||_F}{||\mathbf{Z}_{cmds}||_F}. \tag{3.10}$$

Se realiza un total de `Nrep=100` repeticiones de cada configuración, donde $\mathbf{Z}_{cmds}$ es la matriz que se obtiene con CMDS. El primer coeficiente es el error en la pérdida Strain, el segundo es el Stress-1 de Kruskal y el tercero es un coeficiente de error del ajuste directo entre las configuraciones de puntos.

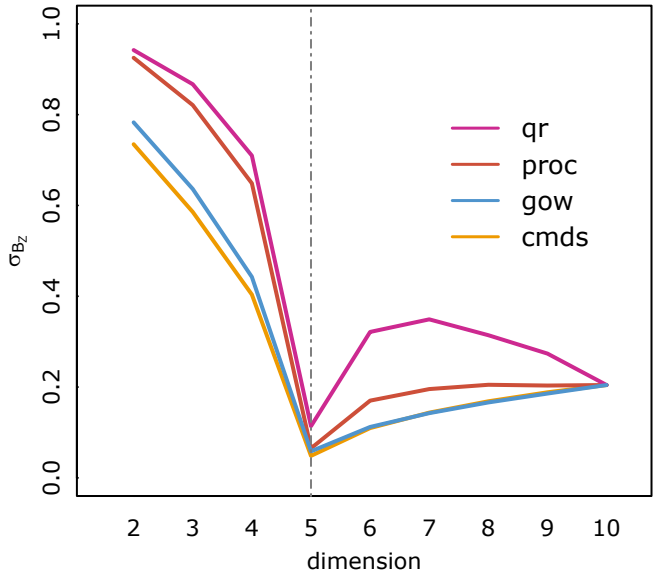


Figura 3.2: Promedios del coeficiente de pérdida Strain para escalados de distintas dimensiones.

	σ_{B_Z}	σ_{D_Z}	σ_Z
`cmds`	0.04845001	0.00789342	0
`proc`	0.06503457	0.01512929	0.03017452
`qr`	0.11382362	0.02963782	0.07386895
`gow`	0.05843108	0.01279837	0.02260295

Tabla 3.2: Promedios de los coeficientes para `k=5`, la dimensión verdadera de $\mathbf{X}$.

.

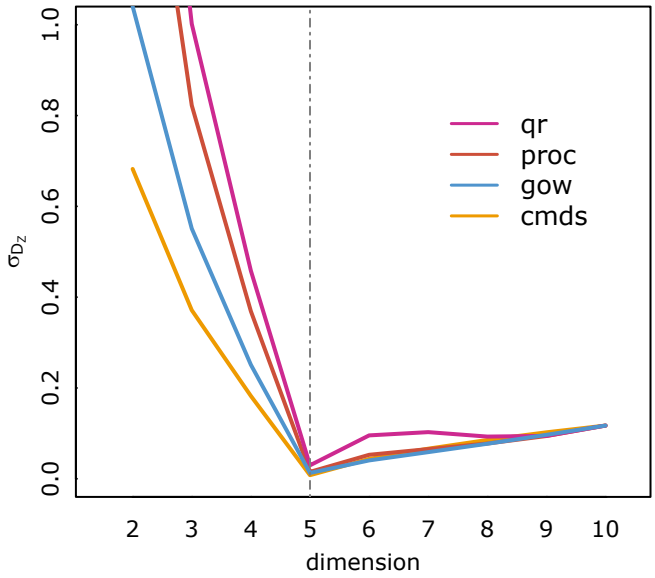


Figura 3.3: Promedios del Stress-1 para escalados de distintas dimensiones.

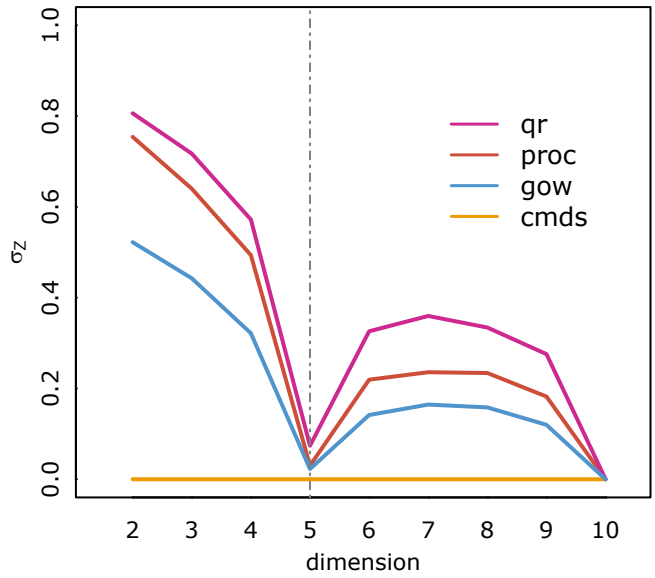


Figura 3.4: Promedios del coeficiente $\sigma_{\mathbf{Z}}$ para escalados de distintas dimensiones.

En estos gráficos puede verse que el método que mejor reproduce la solución de CMDS es el de interpolación con la fórmula de Gower. Los métodos basados en alineación tienen un rendimiento menor, y entre ellos el que se basa en la descomposición QR parece tener un peor ajuste.

Muestras de tamaño mayor

En segundo lugar, se busca estudiar el rendimiento de los métodos rápidos para muestras de tamaño mayor. El esquema general es el siguiente:

- Tamaños de muestra `n` crecientes: 10000, 50000 y 100000.
- Número de repeticiones `Nrep` igual a 100.
- Escenario 1: La distribución de $\mathbf{y}$ se mantiene igual.
- Escenario 2: El 90 % de las observaciones siguen el esquema del escenario 1, pero el 10 % se simulan con varianza 25 para las primeras 2 componentes de error, es decir las variables ε_1 y ε_2. Esto introduce puntos atípicos.

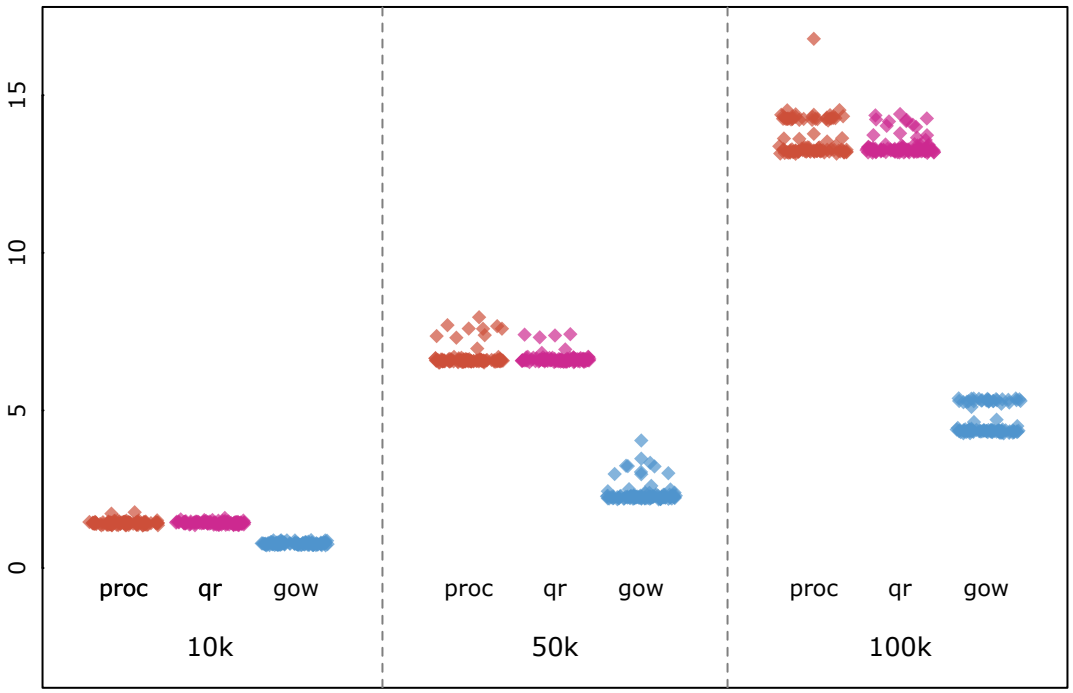


Figura 3.5: Tiempos de procesamiento (en segundos) para muestras de tamaño mayor.

Para tamaños de muestra muy grandes, como en este caso, no es factible realizar el escalado clásico completo ni obtener la totalidad de los autovalores de $\mathbf{B}_X$, por lo que no se pueden calcular las métricas anteriores. Por esta razón, se trabajará con `k=5` autovalores de la covarianza $\hat{\mathbf{\Sigma}}_Z = \frac{1}{n}\mathbf{Z}^t\mathbf{Z}$ de cada escalado en particular, que denotaremos $\hat{\lambda}_a$. Si la representación es buena, estos autovalores no deberían alejarse significativamente de 5, que son las varianzas de las coordenadas *verdaderas* de los puntos. Una primera inspección de los resultados consiste en calcular el promedio de los autovalores de cada solución

$$\frac{1}{5}\sum_{a=1}^{d}\hat{\lambda}_{at}, \quad \text{con} \quad t = \{1, 2, ..., \texttt{Nrep}\}. \tag{3.11}$$

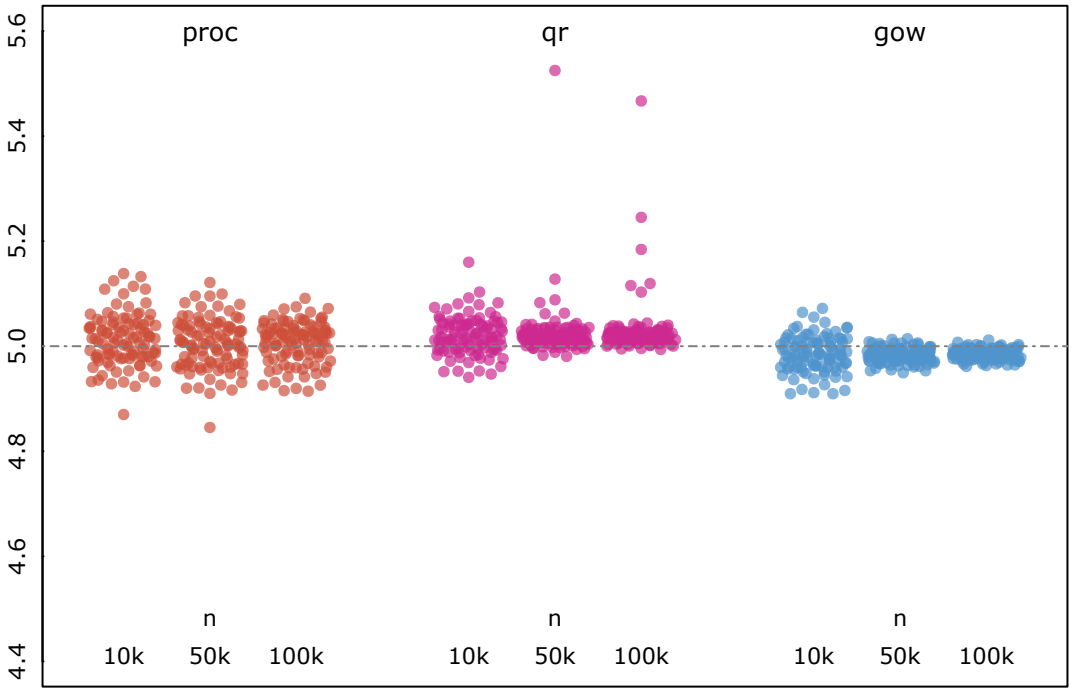


Figura 3.6: Autovalores promedio en escenario 1 (sin datos atípicos).

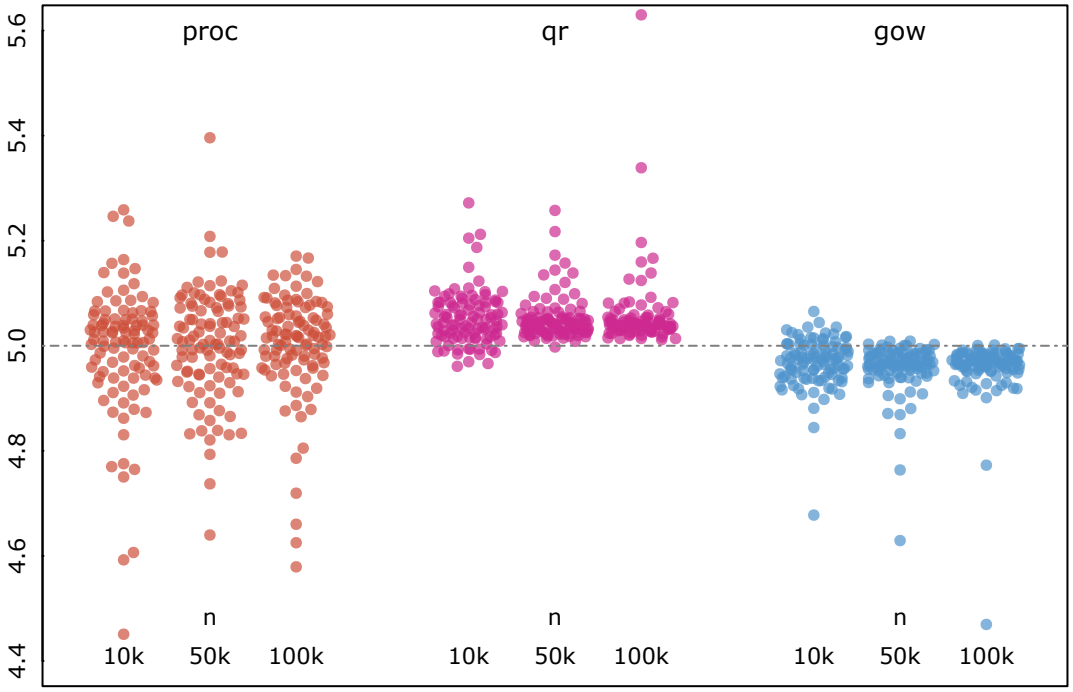


Figura 3.7: Autovalores promedio en escenario 2 (con datos atípicos).

Estos gráficos permiten ver el grado de dispersión de cada método, y el sesgo en relación a los autovalores verdaderos. El método `qr` arrojó algunas soluciones, en ambos escenarios, que se encuentran fuera de la escala del gráfico.

Sean $\boldsymbol{\lambda} = (5, 5, 5, 5, 5)^{\mathrm{t}}$ el vector de autovalores poblacionales y $\hat{\boldsymbol{\lambda}}$ el vector de autovalores muestrales de una solución cualquiera, tenemos

$$\mathsf{B}(\hat{\boldsymbol{\lambda}}) = \mathsf{E}(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}) \quad \text{y} \quad \mathsf{Cov}(\hat{\boldsymbol{\lambda}}) = \mathsf{E}\Big\{ \big(\hat{\boldsymbol{\lambda}} - \mathsf{E}(\hat{\boldsymbol{\lambda}})\big) \big(\hat{\boldsymbol{\lambda}} - \mathsf{E}(\hat{\boldsymbol{\lambda}})\big)^{\mathrm{t}} \Big\}. \tag{3.12}$$

el sesgo y la matriz de covarianzas, respectivamente. Podemos calcular una versión escalar del error cuadrático medio como

$$\begin{aligned}
\mathsf{MSE}(\hat{\boldsymbol{\lambda}}) &= \mathsf{E}\Big\{ \big(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\big)^{\mathrm{t}} \big(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\big) \Big\} \\
&= \mathrm{tr}\ \mathsf{E}\Big\{ \big(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\big) \big(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\big)^{\mathrm{t}} \Big\} \\
&= \mathrm{tr}\ \mathsf{E}\Big\{ \big(\hat{\boldsymbol{\lambda}} - \mathsf{E}(\hat{\boldsymbol{\lambda}}) + \mathsf{B}(\hat{\boldsymbol{\lambda}})\big) \big(\hat{\boldsymbol{\lambda}} - \mathsf{E}(\hat{\boldsymbol{\lambda}}) + \mathsf{B}(\hat{\boldsymbol{\lambda}})\big)^{\mathrm{t}} \Big\} \\
&= \mathrm{tr}\ \Big\{ \mathsf{Cov}(\hat{\boldsymbol{\lambda}}) + \mathsf{B}(\hat{\boldsymbol{\lambda}}) \mathsf{B}(\hat{\boldsymbol{\lambda}})^{\mathrm{t}} + \mathsf{B}(\hat{\boldsymbol{\lambda}}) \mathsf{E}\{\hat{\boldsymbol{\lambda}} - \mathsf{E}(\hat{\boldsymbol{\lambda}})\} + \mathsf{E}\{\hat{\boldsymbol{\lambda}} - \mathsf{E}(\hat{\boldsymbol{\lambda}})\} \mathsf{B}(\hat{\boldsymbol{\lambda}}) \Big\} \\
&= \mathrm{tr}\ \mathsf{Cov}(\hat{\boldsymbol{\lambda}}) + \mathsf{B}(\hat{\boldsymbol{\lambda}})^{\mathrm{t}} \mathsf{B}(\hat{\boldsymbol{\lambda}})
\end{aligned} \tag{3.13}$$

Esta expresión separa el sesgo respecto de la variabilidad neta de cada método. Llamemos $\hat{\boldsymbol{\lambda}}$ al vector de autovalores del t-ésimo escalado de la simulación, donde $\mathrm{t} = 1, 2, ...,$ `Nrep`, estimamos el error cuadrático como

$$\hat{\mathsf{MSE}}(\hat{\boldsymbol{\lambda}}_{\mathrm{t}}) = \ \mathrm{tr} \left(\frac{1}{\mathtt{Nrep} - 1} \sum_{\mathrm{t}=1}^{\mathtt{Nrep}} \big(\hat{\boldsymbol{\lambda}}_{\mathrm{t}} - \overline{\boldsymbol{\lambda}}\big) \big(\hat{\boldsymbol{\lambda}}_{\mathrm{t}} - \overline{\boldsymbol{\lambda}}\big)^{\mathrm{t}} \right) + \big\| \overline{\boldsymbol{\lambda}} - \boldsymbol{\lambda} \big\|^2. \tag{3.14}$$

Donde $\overline{\boldsymbol{\lambda}} = \frac{1}{\mathtt{Nrep}} \sum_{\mathrm{t}=1}^{\mathtt{Nrep}} \hat{\boldsymbol{\lambda}}_{\mathrm{t}}$ es la media muestral de los vectores de autovalores.

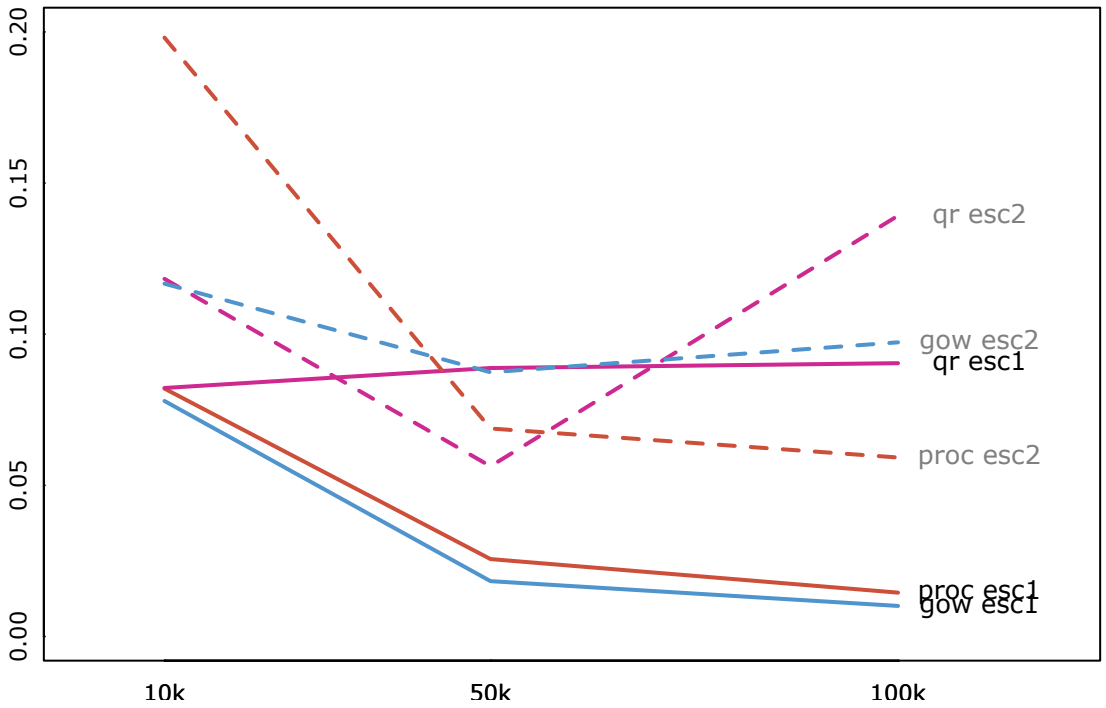


Figura 3.8: Error cuadrático para distintos tamaños de muestra, en los escenarios 1 y 2

		$\widehat{\text{MSE}}(\hat{\boldsymbol{\lambda}})$	tr $\widehat{\text{Cov}}(\hat{\boldsymbol{\lambda}})$	$\text{B}(\hat{\boldsymbol{\lambda}})^{\text{t}}\text{B}(\hat{\boldsymbol{\lambda}})$
proc	n=10k	0.0820	0.0170	0.0650
	n=50k	0.0256	0.0123	0.0133
	n=100k	0.0145	0.0081	0.0064
qr	n=10k	0.0822	0.0131	0.0691
	n=50k	0.0888	0.0661	0.0227
	n=100k	0.0904	0.0714	0.0190
gow	n=10k	0.0779	0.0109	0.0670
	n=50k	0.0183	0.0021	0.0162
	n=100k	0.0101	0.0013	0.0088

Tabla 3.3: Detalle del escenario 1 (sin datos atípicos).

		$\widehat{\text{MSE}}(\hat{\boldsymbol{\lambda}})$	tr $\widehat{\text{Cov}}(\hat{\boldsymbol{\lambda}})$	$\text{B}(\hat{\boldsymbol{\lambda}})^{\text{t}}\text{B}(\hat{\boldsymbol{\lambda}})$
proc	n=10k	0.1981	0.1346	0.0635
	n=50k	0.0688	0.0559	0.0129
	n=100k	0.0592	0.053	0.0062
qr	n=10k	0.1183	0.0307	0.0876
	n=50k	0.0563	0.0226	0.0337
	n=100k	0.1392	0.1055	0.0337
gow	n=10k	0.1167	0.0301	0.0866
	n=50k	0.0874	0.0465	0.0409
	n=100k	0.0973	0.066	0.0313

Tabla 3.4: Detalle del escenario 2 (con datos atípicos).

Algunas observaciones generales

Los resultados mostrados son sólo orientativos, ya que la variabilidad está influenciada en todos los casos por los errores numéricos del algoritmo y por la elección de los hiperparámetros. Estos temas merecen un análisis más detallado.

- El método de interpolación de Gower es más rápido que los otros, y parece ser el que arroja un resultado más similar a CMDS.
- El método de alineación con descomposición QR es el que produce un peor ajuste, debido a la falta de una transformación de cambio de escala, que sí tiene el método de Procrustes.
- El método de Procrustes tiene más varianza que los demás, pero a su vez menos sesgo.
- El método de Procrustes parece ser el más sensible a datos atípicos. Los otros dos métodos sufren alteraciones con sesgo hacia un lado.
- Hay una solución atípica muy alejada de las demás para el método QR, que no se visualiza en los gráficos 3.6 y 3.7.
- En el gráfico de tiempos, a medida que aumenta el tamaño de muestra, se acentúa la presencia de dos grupos de soluciones. Unas más rápidas y otras más lentas.
- Para el método de Gower, queda pendiente la proyección de las nuevas coordenadas al espacio de las coordenadas originales, esto podría hacer que el método mejore.

3.2.2. Trazabilidad de las soluciones

Cada nube de puntos se simula con semilla `n+i-1` donde `n` es el tamaño de la muestra e `i` es el número de repetición o ciclo. Entonces `set.seed(10001)` se usa en la segunda repetición de la muestra de tamaño 10000. Dentro de cada método se usa una aleatorización para elegir puntos: En los métodos `proc` y `qr`, se permutan al azar las observaciones con una semilla, y los puntos de referencia van a ser los `c` primeros. En el método `gow` se debe elegir un bloque al azar de puntos de tamaño `m` con una semilla. A la semilla del data set original, se le suma $w10^6$, donde `w` es el número de método: `w=1` es `proc`, `w=2` es `qr`, y `w=3` es `interp`. Por ejemplo, `set.seed(2010002)` es la semilla que se usa para elegir las particiones del método `qr` en la muestra número 2 de la corrida con muestras de tamaño 10000.

3.2.3. Repositorio

Listamos los archivos contenidos en el repositorio:

- `aux_functions.R` - Funciones auxiliares usadas a lo largo del trabajo.
- `Ejemplos Cap1.R` - Desarrollo de los ejemplos y casos del capítulo 1.
- `Ejemplos Cap2.R` - Desarrollo de los ejemplos y casos del capítulo 2.
- `cmdscaling.R` - Función principal para MDS usada en la simulación.
- `Simulacion Main.R` - Código con el que se realizó la simulación.
- `Simulacion Main Results.RData` - Resultados de la simulación.
- `Simulacion Analisis.R` - Cálculos y gráficos del capítulo 3.
- `Cap4.R` - Desarrollo del caso del capítulo 4 (próximo).
- `Cap4 Data.RData` - Resultados del capítulo 4 (próximo).

Capítulo 4

Caso de Aplicación

4.1. Introducción

En este capítulo trabajamos con un caso de aplicación de los métodos vistos para MDS con grandes datos. A través de un problema puntual, aplicamos MDS como técnica de reducción de dimensión previa a una separación de los datos en grupos. Para ello, hacemos uso de una función de implementación propia en `R`. Se busca mostrar la aplicación de MDS como forma de sortear la *maldición de la dimensión* en un problema de agrupamiento para una muestra de gran cantidad de datos. En casos como éste, los métodos usuales de MDS serían impracticables y debería ser necesario usar otras técnicas como Análisis de Componentes Principales o t-SNE.

4.1.1. Instalación de librería `mvtools`

En la librería `mvtools`, de desarrollo propio, se encuentra la función `procrustes_mdscal` que utilizaremos en este capítulo. El código se encuentra en `https://github.com/pcosatto/mvtools/blob/main/R/procrustes_mdscal.R` y para instalar la librería en R-Studio se debe ejecutar

```
devtools::install_github("pcosatto/mvtools")
```

en la consola, habiendo instalado previamente el paquete `devtools`. Luego se debe llamar a la librería de la forma usual con `library(mvtools)`. La página de ayuda de la función `procrustes_mdscal` se puede abrir con `?procrustes_mdscal`. Esta función implementa DCO-MDS con el método de Procrustes, usando escalado de distancias. Se realiza el ajuste del Stress con el algoritmo SMACOF, y se usan *splines* como mapeo de las disimilaridades.

Para obtener las disimilaridades entre las observaciones, es posible usar las métricas de Minkowski o la métrica de Gower para tipos de datos mixtos. La utilización de estas métricas puede causar que la implementación resulte más lenta que la de la función `cmdscaling_test` usada en el capítulo anterior. Esta última fue específicamente diseñada para el estudio de simulación y no tiene la funcionalidad necesaria para el uso práctico.

4.2. Caso de aplicación

Se tienen datos de `n=8621` clientes de un banco, para los que se registraron las siguientes variables sobre el uso de sus tarjetas de crédito. El objetivo es encontrar segmentos específicos entre los clientes, mediante análisis de *clusters*. Los datos originales se encuentran en `https://www.kaggle.com/datasets/arjunbhasin2013/ccdata`. Se trabajará con las siguientes variables:

- `oneoff_purchases` - Monto máximo de compra realizada en un pago.
- `install_purchases` - Monto de compras realizadas en cuotas.
- `cash_adv` - Adelantos de efectivo otorgados al cliente.
- `oneoff_purchases_freq` - Índice de la frecuencia de compras en un pago (1 = frecuente, 0 = no frecuente).
- `install_purchases_freq` - Índice de la frecuencia de compras en cuotas (1 = frecuente, 0 = no frecuente).
- `cash_adv_freq` - Índice de la frecuencia de adelantos de efectivo (1 = frecuente, 0 = no frecuente).
- `purchases_trx` - Cantidad de transacciones en el último período.
- `credit_limit` - Monto límite de la tarjeta de crédito del cliente.
- `prc_full_payment` - Proporción de las veces que el cliente realiza el pago completo del saldo.

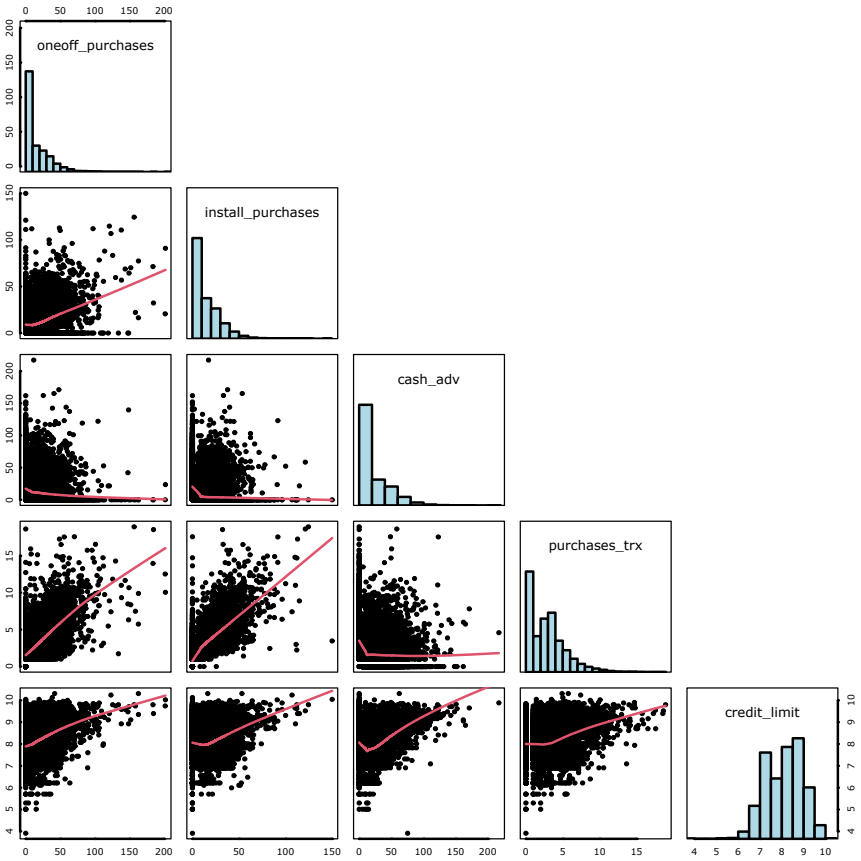


Figura 4.1: Gráficos de pares de cuatro de las variables, con datos transformados

Como disimilaridad para realizar MDS en este problema usamos la distancia de Gower para variables cuantitativas. Esta distancia toma valores cercanos a 1 para máxima disimilaridad, y 0 para máxima similaridad. Dados dos vectores de datos $\mathbf{x}_1$ y $\mathbf{x}_2$ compuestos de p variables cuantitativas, la distancia de Gower entre ellos se calcula como:

$$d_{ij} = 1 - \left\{ \frac{1}{p} \sum_{k=1}^{p} \left(1 - \frac{|X_{1k} - X_{2k}|}{R_k} \right) \right\}, \tag{4.1}$$

donde R_k es el rango de la k-ésima variable. Como la distancia depende directamente del rango, se comprobó previamente la inexistencia de datos atípicos significativos en la muestra. Para trabajar en escalas más reducidas, se tomó raíz cuadrada a `oneoff_purchases`, `install_purchases`, `purchases_trx` y `cash_adv`; y logaritmo a la variable `credit_limit`, como se muestran en el gráfico 4.1. Todas estas variables presentaban una asimetría positiva muy significativa en la muestra original, lo que puede afectar sensiblemente la métrica.

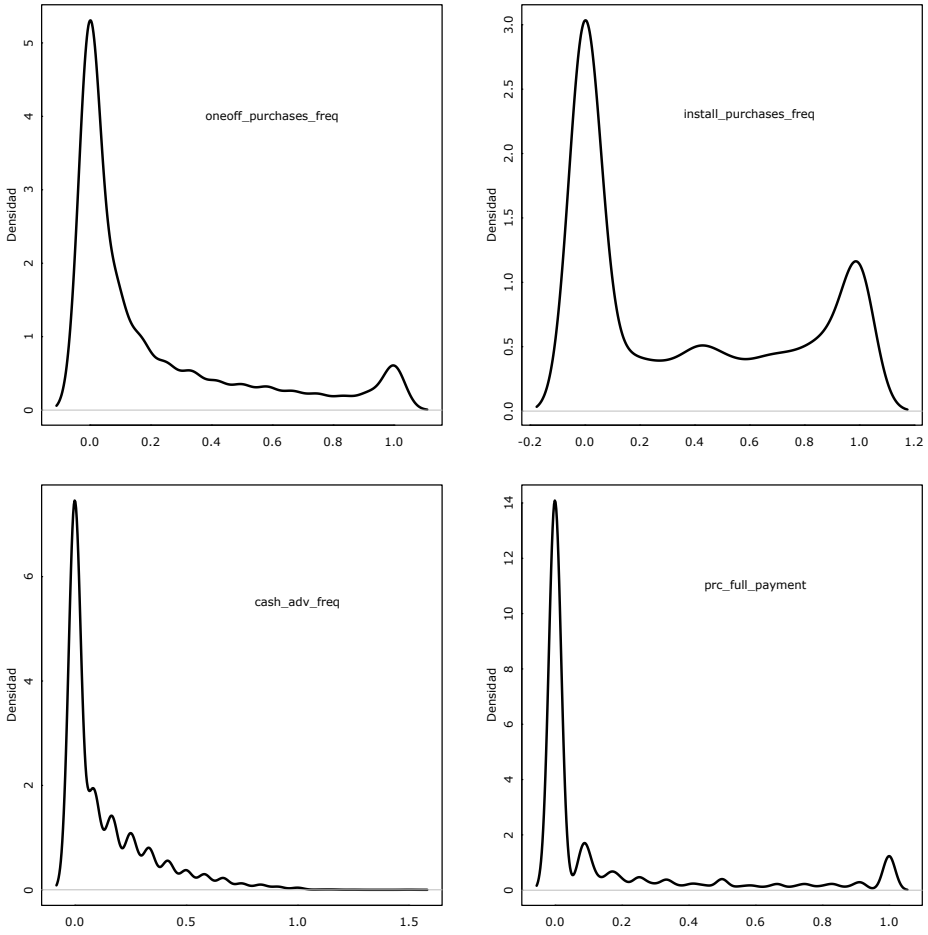


Figura 4.2: Gráficos de densidad de las variables restantes, sin transformar.

4.2.1. Reducción de la dimensión

Un primer acercamiento al problema consiste en reducir la dimensión a un espacio bidimensional con la función `procrustes_mdscal`. A continuación se muestra la nube obtenida:

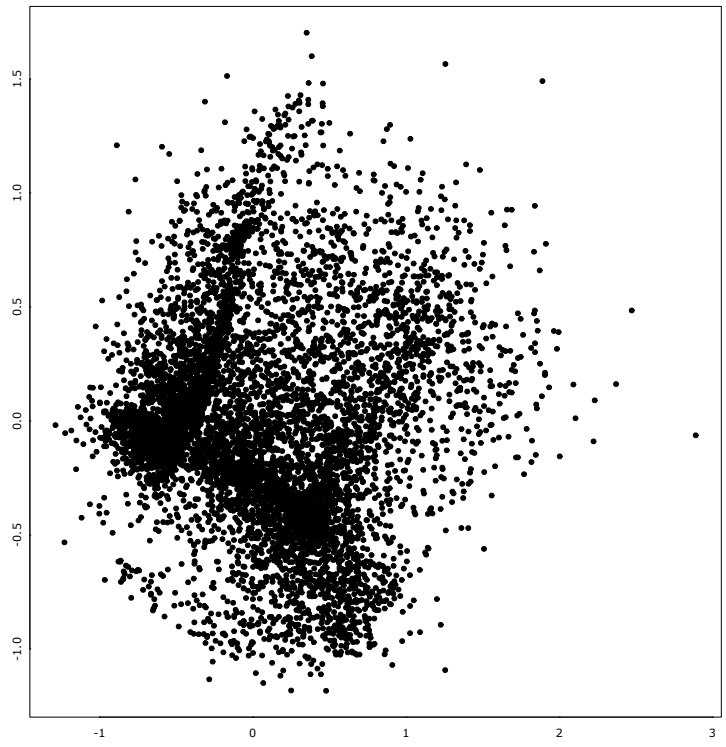


Figura 4.3: Escalado de distancias bidimensional (puntos en negro)

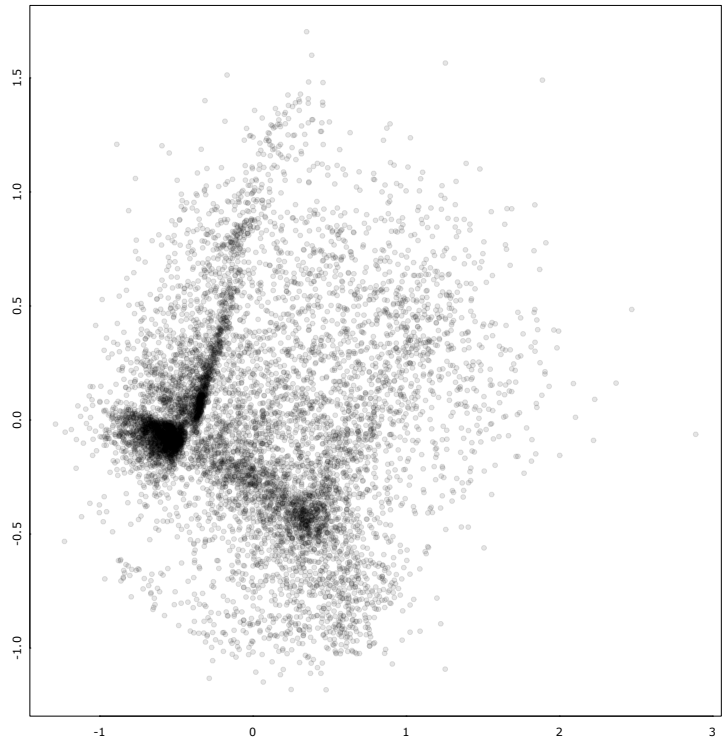


Figura 4.4: Escalado de distancias bidimensional (puntos con transparencia)

Al tratarse de una nube muy densa de puntos, es difícil realizar un análisis del resultado a simple vista. Situaciones como esta son esperables en reducciones con gran cantidad de datos. En el gráfico 4.4 se agrega transparencia a los puntos, lo que permite identificar algunas zonas de mayor concentración, donde podrían encontrarse agrupamientos. También se observan algunos puntos aislados con densidad muy alta en relación al resto.

4.2.2. Identificación de grupos

Aplicamos un método de agrupamiento basado en la densidad llamado DBSCAN (*Density-based spatial clusering of applications with noise*), usando la librería `dbscan`. Este método es adecuado para realizar agrupamientos en nubes de puntos con alta densidad ya que detecta zonas compactas y separa observaciones como *ruido*. No sólo no requiere la especificación previa de la cantidad de grupos sino que no forma grupos con la totalidad de las observaciones.

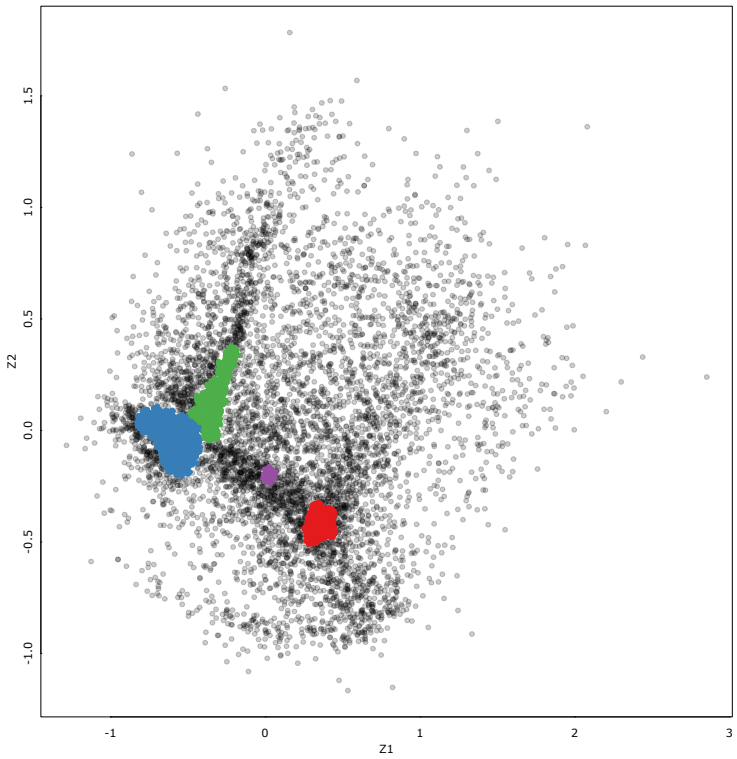


Figura 4.5: Agrupamientos con DBSCAN, hechos con los datos escalados a $\mathbb{R}^2$

En el gráfico se visualizan cuatro grupos distintos que pueden encontrarse en el escalado bidimensional. Como el método DBSCAN tiene dos hiper-parámetros, fue necesario determinarlos con una búsqueda de grilla. Se buscó la configuración de al menos 4 grupos que haga máximo el índice

$$\frac{\sum_{k=1}^{K} n_k ||\bar{\mathbf{z}}_k - \bar{\mathbf{z}}||^2}{\sum_{k=1}^{K}\sum_{i=1}^{n} ||\mathbf{z}_{ik} - \bar{\mathbf{z}}_k||^2} \frac{n-k}{K-1}, \tag{4.2}$$

donde $k \in \{1, 2, ..., K\}$ es el grupo, n_k es la cantidad de observaciones por grupo, $\bar{\mathbf{z}}$ es la media de todas las observaciones en el espacio reducido y $\bar{\mathbf{z}}_k$ es la media de las observaciones del k-ésimo grupo.

Este índice, conocido como métrica de Calinski-Harabasz, aumenta cuando hay grupos compactos y separados entre sí. La ventaja para esta aplicación es que no se requiere del cálculo de todos los pares de distancias entre puntos, sino las distancias a los centroides. En la figura 4.5 se ve un agrupamiento óptimo con la restricción $K \geq 4$. En el `aux_functions.R` del repositorio se encuentra la función `optimize_dbscan` utilizada para esta búsqueda, y en `Cap4.R` se encuentran los valores utilizados para armar la grilla.

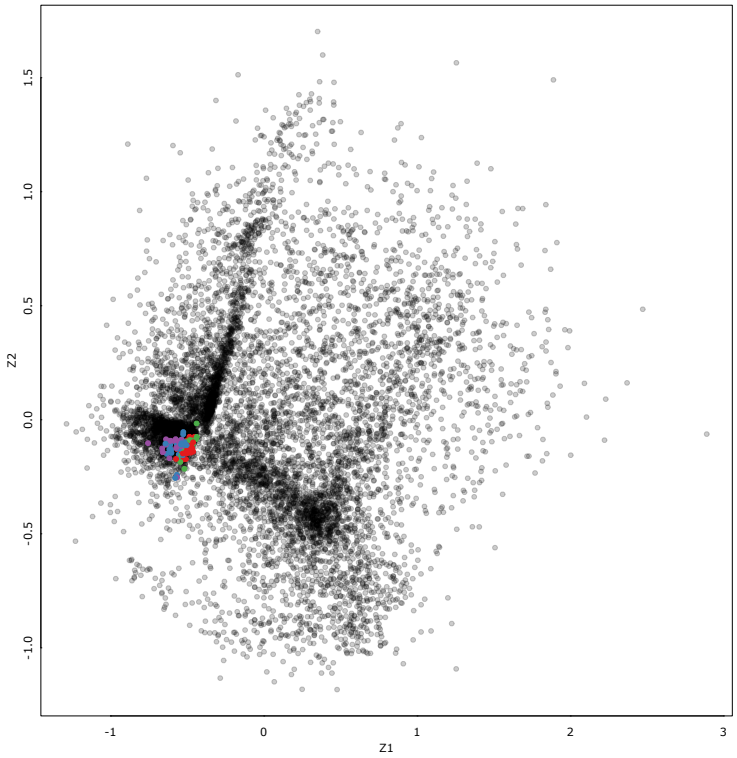


Figura 4.6: Agrupamientos con DBSCAN, hechos con los datos originales

Se realizó el mismo procedimiento para los datos originales, sin reducir la dimensión. En la figura 4.6 se representan los puntos con las coordenadas reducidas, pero coloreados según el agrupamiento que se realizó teniendo en cuenta todas las dimensiones, para una combinación óptima de hiperparámetros.

	ruido	**1**	**2**	**3**	**4**
ruido	6252	1	0	0	0
1	1540	11	16	12	12
2	202	0	0	0	0
3	528	0	0	0	0
4	50	0	0	0	0

Tabla 4.1: Grupos con MDS (filas) versus grupos con dimensiones originales (columnas)

La mala calidad del agrupamiento puede deberse a que la dimensión de los datos es muy alta como para lograr zonas de densidad suficiente. Esto se conoce como *maldición de la dimensión*. Para sortear este problema se usan habitualmente métodos de reducción de dimensión previos. El escalado multidimensional hubiera quedado excluido de las opciones para este problema debido a la gran cantidad de observaciones.

Una representación tridimensional

Podemos extender la representación a tres dimensiones, aumentando el espacio del escalado. En la figura se muestran las vistas canónicas de la nube de puntos, incluyendo los agrupamientos realizados con DBSCAN con el mismo procedimiento de optimización de hiper-parámetros que se hizo para dos dimensiones.

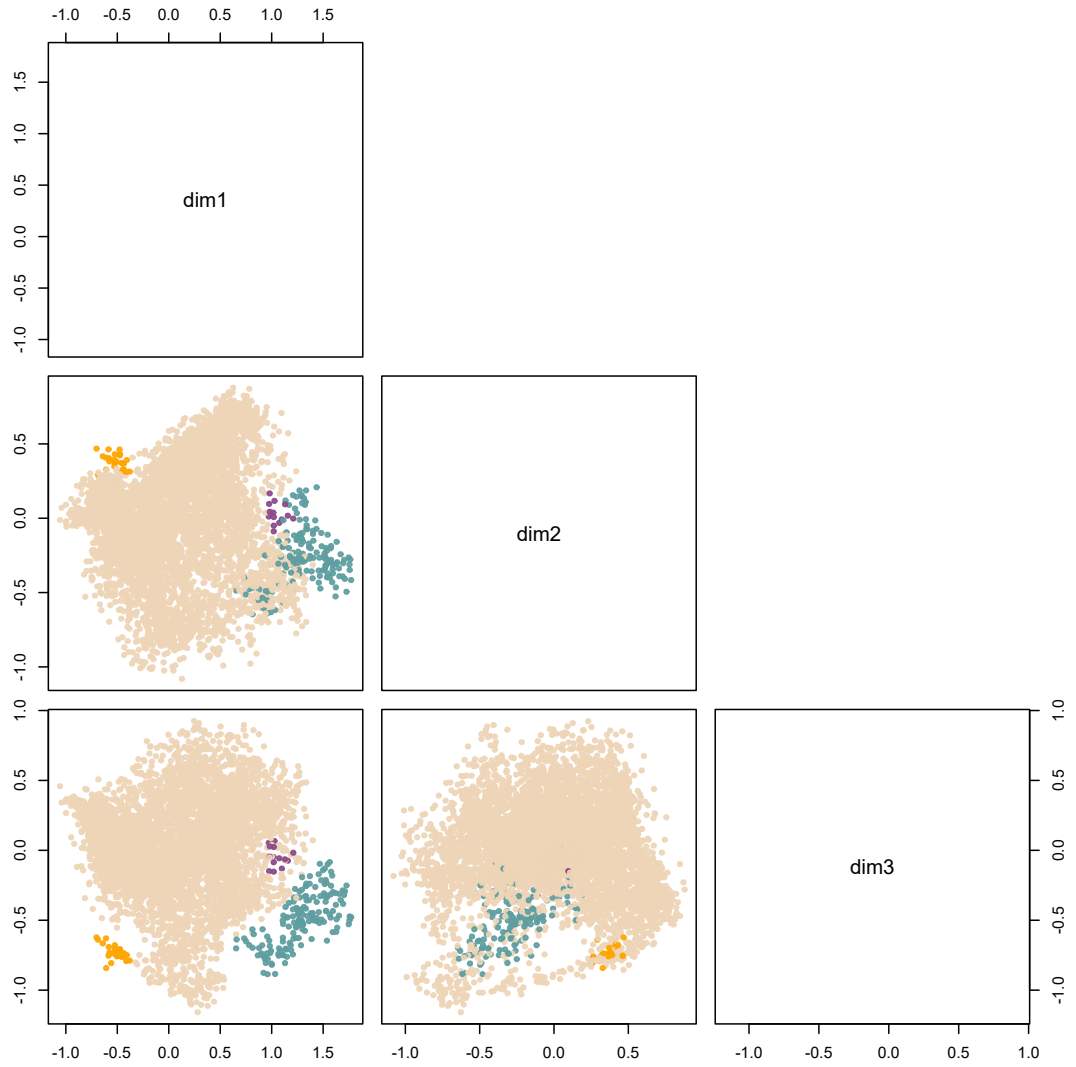


Figura 4.7: Agrupamientos en tres dimensiones, excluyendo los puntos clasificados como ruido.

Analizamos la composición de los grupos calculando

$$100\ \hat{\mathrm{F}}_{\mathrm{j}}\Big(\mathrm{M_e}(\mathrm{Z_{jk}})\Big), \tag{4.3}$$

donde $\mathrm{j} \in \{1, 2, ..., \mathrm{p}\}$ es la variable, $\hat{\mathrm{F}}_{\mathrm{j}}$ es la función de distribución empírica de la j-ésima variable obtenida con todas las observaciones y $\mathrm{M_e}(\mathrm{Z_{jk}})$ es la mediana de las observaciones de la j-ésima variable en el k-ésimo grupo. Esta medida permite saber cómo está ubicado el grupo en relación al total de las observaciones, incluyendo aquellas clasificadas como ruido. Valores cercanos a 100 indican que el grupo tiene valores altos para la j-ésima variable, caso contrario si están cerca de 0. Para valores cercanos a 50 significa que no se alejan del centro de la nube para esa coordenada.

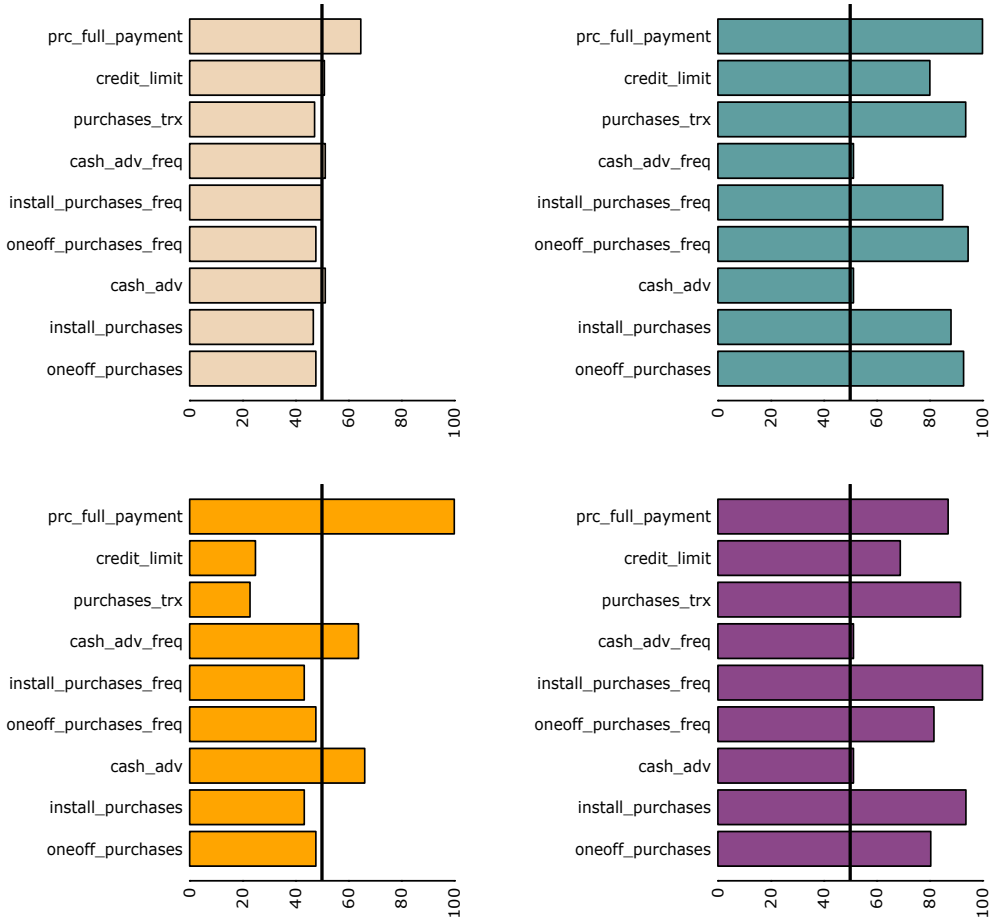


Figura 4.8: Valores de 100 $\hat{\mathrm{F}}_{\mathrm{j}}\left(\mathrm{M}_{\mathrm{e}}(\mathrm{Z}_{\mathrm{jk}})\right)$ para los cuatro grupos.

En los cuatro grupos encontrados, el primero (superior izquierda, color **durazno**) que es el que representa a la mayoría de los clientes del banco, posicionados cerca del centro de la nube y sin un comportamiento distintivo. El segundo grupo más representativo (superior derecha, color **azul**) está compuesto de clientes que hacen compras muy frecuentemente y en su mayoría son compras en un pago. El siguiente grupo (inferior izquierda, color **naranja**) es el de los clientes que usan su tarjeta casi únicamente para pedir adelantos de efectivo. Realizan muy pocas compras, tienen bajo límite de crédito y se caracterizan por tener el porcentaje de pago completo cerca del 100 %. El último grupo hallado (inferior derecha, **violeta**) es similar al segundo, solo que se destacan por realizar compras en cuotas en lugar de hacer compras en un pago.

Un análisis más detallado de esta composición podría incluir la comparación de las estructuras de correlación entre los distintos grupos. La representación visual de la figura 4.7 nos permite ver el peso que tiene cada grupo en la composición de clientes, y la cercanía entre los distintos grupos. Un problema adicional podría ser el de clasificar nuevos clientes en estos grupos para poder ofrecer productos o promociones acordes a su tipo de consumo.

4.2.3. Bondad de ajuste

Los métodos rápidos de MDS tienen la desventaja de que no es posible obtener el valor exacto de métricas como el *Stress-1* o el coeficiente de la pérdida *Strain*, debido a la cantidad de distancias y disimilaridades en muestras de gran tamaño. En este caso, deben definirse nuevas formas de estudiar la bondad de ajuste.

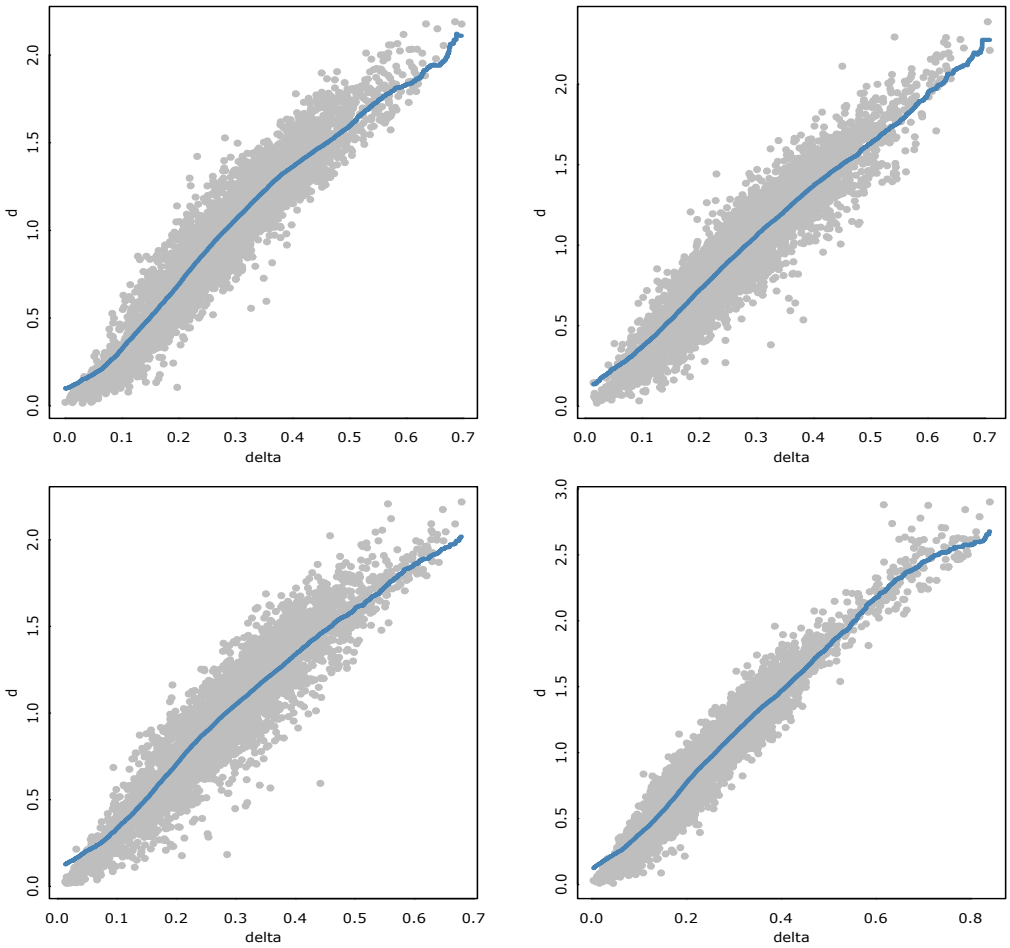


Figura 4.9: Disimilaridades versus distancias euclídeas en $\mathcal{Z}$, para cuatro muestras de 100 clientes.

Notemos que, al haber utilizado escalado de distancias con funciones de mapeo ajustadas dentro de cada grupo del DCO-MDS, se agrega un costo adicional si se desea recopilar las disparidades $\hat{d}_{ij}$ exactas de cada distancia. En la figura 4.9 se muestran los plots de disimilaridad δ_{ij} (en este caso calculada con la métrica de Gower) versus distancia euclídea d_{ij} en el espacio reducido, para cuatro muestras aleatorias de 100 observaciones. A estos gráficos se superpuso la función de regresión ajustada con Nadaraya-Watson. Una opción sencilla para analizar la bondad de ajuste global es calcular los residuos de sucesivas regresiones como éstas y obtener las sumas de cuadrados residuales. Con estas sumas pueden obtenerse estimaciones del *Stress-1* y así construir, por ejemplo, un intervalo de confianza mediante *bootstrap* no paramétrico.

Otro método, similar al implementado en [3], consiste en calcular el *Stress-1* dentro de cada bloque del DCO-MDS y luego hacer un promedio ponderado de los distintos valores, de acuerdo al tamaño de los grupos. Con esto se pierde la información de las distancias cruzadas entre distintos bloques.

4.2.4. Comparación con t-SNE

Un método muy usado para reducción de la dimensión orientada a *clustering* es t-SNE (*t-distributed stochastic neighbour embedding*), que se basa en ajustar una medida de similaridad basada en la probabilidad de que una observación elija a otras como vecinos:

$$p_{i|j} = \frac{e^{-|x_i - x_j|^2/2\sigma_j^2}}{\sum_{k \neq j} e^{-|x_k - x_j|^2/2\sigma_j^2}} \tag{4.4}$$

Donde σ es un parámetro que se determina en función de la *perplexity* elegida por el usuario. La *perplexity* es un coeficiente que típicamente se encuentra entre 5 y 50 y permite formar vecindarios de puntos más o menos densos. El método produce coordenadas en dimensión reducida de modo que la similaridad $p_{i|j}$ calculada con estos puntos se ajuste lo mejor posible a la que se calcula con los puntos de alta dimensión.

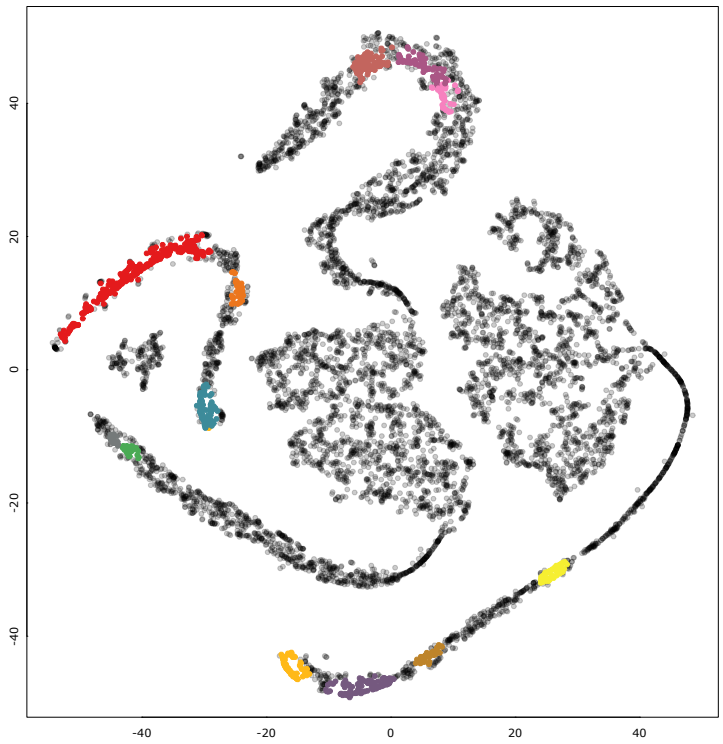


Figura 4.10: Agrupamientos de clientes con t-SNE, con *perplexity* igual a 50

En el gráfico se muestra el resultado de agrupar los clientes luego de realizar una reducción con t-SNE. Dado un valor de la *perplexity*, se realizó la misma búsqueda óptima de grupos según el índice de Calinski-Harabasz. Se probaron los resultados con distintos valores de *perplexity* y se encontró que la configuración con valores menores a 50 formaba gran cantidad de grupos pequeños con formas muy distorsionadas. Al parecer, este método tiende a exagerar las estructuras no lineales, lo que dificulta la interpretación de los grupos y su visualización espacial.

Temas a profundizar

A modo de conclusión del trabajo, enumeramos algunos temas que creemos que merece la pena tratar con profundidad:

En relación a los métodos rápidos en sí mismos

- Mejorar el método DCO-MDS con descomposición QR para incorporar un cambio de escala. Se pudo mostrar cómo este método tiene menos capacidad que los demás para reproducir la solución clásica. En línea con esto, buscar un método unificado de DCO-MDS que combine las dos estrategias vistas en este trabajo.
- Investigar la respuesta del método DCO-INTERP con la fórmula de Gower con otras disimilaridades, ya que sólo fue analizado con distancias euclídeas.
- Pudimos mostrar que los datos atípicos influyen sensiblemente en la solución. Puede ser interesante proponer variantes robustas de los métodos rápidos.
- En los métodos vistos se deben seleccionar puntos de referencia o *landmarks*. En [7] se propone un criterio de selección de estos puntos, en lugar de elegirlos al azar. Esto podría disminuir la influencia de datos atípicos o mejorar la calidad de la representación final.
- Estudiar con mayor detalle la influencia de los hiper-parámetros -tamaño máximo de los bloques, cantidad de *landmarks*- en la calidad de la solución. En este trabajo fijamos los hiper-parámetros de acuerdo a las recomendaciones generales de los autores, pero notamos que la calidad de la solución varía al modificarlos. No se realizó un análisis de sensibilidad exhaustivo.

En relación a la aplicación de los métodos

- Proponer medidas de bondad de ajuste que requieran poco tiempo de cómputo y que representen adecuadamente la calidad de la solución.
- Estudiar la construcción de biplots no lineales usando los métodos rápidos, para ayudar a comprender la representación en baja dimensión.
- Analizar con mayor detalle la conveniencia de usar distintas métricas para MDS en problemas de reducción de la dimensión.
- Buscar métodos de regularización de variables en problemas de reducción de la dimensión. Por ejemplo, dada una métrica ponderada por variables, de la forma de

$$\mathrm{d_{ij}} = \sum_{\mathrm{k=1}}^{\mathrm{p}} \mathrm{w_k}\ \mathrm{d_{ijk}},$$

donde $\mathrm{d_{ijk}}$ es el término de distancia entre $\mathrm{O_i}$ y $\mathrm{O_j}$ que corresponde a la k-ésima variable y

w_k es un coeficiente de peso, analizar cómo calibrar los coeficientes de peso para seleccionar variables en un escalado multidimensional. En otras palabras, detectar qué variables ayudan más a representar las distancias en baja dimensión y en qué grado. Para esto debe ser necesario implementar algún método de optimización o búsqueda.

- Investigar aplicaciones alternativas, como la representación simplificada de grafos o redes con gran cantidad de nodos.

Referencias

[1] Ingwer Borg y Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[2] Michael AA Cox y Trevor F Cox. *Multidimensional scaling*. Chapman y Hall/CRC, 2001.

[3] Pedro Delicado y Cristian Pachon-Garcia. "Multidimensional Scaling for Big Data". En: *arXiv preprint arXiv:2007.11919* (2020).

[4] John C Gower y David J Hand. *Biplots*. Vol. 54. CRC Press, 1995.

[5] Samudra Herath, Matthew Roughan y Gary Glonek. "High Performance Out-of-sample Embedding Techniques for Multidimensional Scaling". En: *arXiv preprint arXiv:2111.04067* (2021).

[6] KV Mardia, JT Kent y JM Bibby. *Multivariate analysis, 1979*. Academic Press Inc, 1979.

[7] Emmanuel Paradis. "Multidimensional scaling with very large datasets". En: *Journal of Computational and Graphical Statistics* 27.4 (2018), págs. 935-939.

[8] Nasir Saeed et al. "A survey on multidimensional scaling". En: *ACM Computing Surveys (CSUR)* 51.3 (2018), págs. 1-25.

[9] Jengnan Tzeng, Henry Horng-Shing Lu y Wen-Hsiung Li. "Multidimensional scaling for large genomic data sets". En: *BMC bioinformatics* 9.1 (2008), págs. 1-17.