Trabajo de Pasantía

Licenciatura en Estadística

Clustering Robusto en el Paradigma Basado en Modelos

Rodrigo Gadea, Leonardo Moreno

Orientador

Dr. Marco Scavino

Tribunal

Ing. Enrique Cabaña

Dr. Ricardo Fraiman

Dr. Juan Goyeneche

Instituto de Estadística
Facultad de Ciencias Econoómicas y Administración
Universidad de la República

Montevideo, 2011

Resumen

Una gran variedad de técnicas o procedimientos de clustering han surgido en los últimos años en distintas ramas de la investigación estadística.

Sin embargo, la mayoría se basa en conjeturas heurísticas que carecen del debido rigor científico, lo que conlleva a conclusiones "oscuras" y posiblemente ficticias. Brindar procedimientos consistentes, que sean eficientes en distintos paradigmas de datos - inclusive en presencia de ruido - es un problema de elevada complejidad.

Algunas principales líneas recientes de investigación, tales como McLachlan (2006) [21], Gallegos y Ritter (2005, 2009) [38] [39], Cuesta, Gordaliza y Matran (2008) [24], abordan el problema de clustering desde una óptica basada en modelos y en presencia de outliers, intentando brindar respuestas formales en este ámbito.

El objetivo de esta monografía es introducir, analizar, comparar e implementar distintas técnicas robustas de clustering basadas en modelos probabilísticos. La finalidad es de poder discernir que procedimiento es más eficiente y estable frente a distintas tipologías de datos, pudiendo así obtener conclusiones más precisas sobre los grupos obtenidos.

El análisis se realizó sobre distintos patrones de datos simulados y luego se aplicaron las técnicas a un conjunto de datos reales.

Palabras Claves: Cluster, Robustez, Outliers, k-Medias, Mezcla de distribuciones, Trimming.

Abstract

A wide range of techniques or procedures of clustering have been developed in different branches of the statistical research recently.

Nevertheless, most of them are based on heuristic conjectures that lack of due scientific rigour, which may lead to "obscure" and possibly false conclusions. Offering solid procedures, efficient in different data paradigms, even in presence of noise, is a problem of great complexity.

Some main research lines, such as McLachlan (2005) [21], Gallegos and Ritter (2005, 2009) [38] [39], Cuesta, Gordaliza, Matran (2008) [24], tackle the problem of clustering from a model-based approach and in presence of outliers, trying to give formal answers in the field.

The aim of this monograph is to introduce, analyze, compare and set up different robust clustering techniques based on probabilistic models. The goal is being able of discerning which is the most efficient as well as stable procedure against different types of data, getting more accurate conclusions over the resultant clusters.

The analysis has been done on different simulated data patterns and then applied the techniques to a real dataset.

Key words: Cluster, Robustness, Outliers, k-Means, Mixture of distributions, Trimming.

Índice general

Índice de figuras

Introducción

La *clasificiación*, o *categorización*, es uno de los procesos en los que se sustenta el aprendizaje en los seres humanos. Esta habilidad es tomada como sinónimo de inteligencia por algunas teorías psicológicas y cognitivas, incluso midiendo la misma a través de problemas de clasificación (como lo son las pruebas de Cociente Intelectual).

La primera definición considerada científica de *clasificación* se le reconoce a Aristóteles (384 AC - 322 AC), quien en sus *Tópicos* propone los *Cinco Predicables* para describir la lógica de la clasificación. Una definición más reciente del problema es dada por John Stuat Mill (1806 - 1873), quien la definía como:

> La clasificación es la conjunción, actual o ideal, de aquellos que son similares, y la separación de aquellos que no lo son; el propósito de esta conjunción es primariamente:
>
> 1. el facilitar las operaciones en la mente concibiendo claramente y reteniendo en la memoria el carácter de los objetos en cuestión,
> 2. el desentrañar leyes de correlación de propiedades de unión y circunstancias, y
> 3. el habilitar el registro de los mismos de forma de que sean referenciados convenientemente.

La clasificación estadística, o matemática, es la realización de este proceso mediante herramientas estadístico-matemáticas.

Existen dos escenarios bien diferenciados dentro de la clasificación estadística, para los cuales han sido desarrollados métodos y teorías casi independientes: cuando se tiene un conjunto de objetos de referencia de donde inferir la clase o categoría, y cuando no se tiene referencia alguna sobre las posibles categorías. En el primer escenario se suele llamar "Análisis de Regresión" o "Análisis de Clasificación" según la naturaleza de la variable del análisis en cuestión, mientras que en el segundo se suele hablar de "Análisis de Grupos" o "Clustering". Desde el ámbito del reconocimiento de

patrones y desde el de la computación, se les refiere a estos escenarios como *técnicas de clasificación o aprendizaje supervisado* o *técnicas de clasificación o aprendizaje no supervisado*, haciendo énfasis en tener o no referencias para realizar el mismo proceso.

Este trabajo se propone introducir, analizar, comparar e implementar distintas técnicas de Clustering.

La finalidad del mismo es poder discernir qué procedimiento es más eficiente y estable frente a distintas tipologías de datos, pudiendo así obtener conclusiones más precisas sobre los grupos obtenidos. En particular, se centrará en técnicas robustas basadas en modelos probabilísticos, evaluando cuando los algoritmos pueden seguir clasificando o reconociendo correctamente los grupos en las observaciones cuando éstas se encuentran contaminadas - fenómeno cada vez más presente en los conjuntos de datos actuales.

El Análisis de Grupos ha tenido un gran desarrollo en los últimos años, probablemente inducido en gran medida por el desarrollo de herramientas computacionales más potentes.

Muchas veces se observa al Clustering como una colección de técnicas mayoritariamente heurísticas para particionar datos multivariados. Este percepción se apoya en el hecho de que la mayoría de las técnicas de Clustering no son *explícitamente* basadas en un modelo probabilístico. Esto podría "llevar al investigador inocente a creer que él o ella no hicieron ningún supuesto en absoluto, y por ende los resultados son objetivos" (Flury, 1997 [7]). Sin embargo, esa objetividad está lejos de la realidad en tanto la mayoría de las veces los resultados del Clustering están fuertemente afectados por el método elegido y su performance es muy dependiente del modelo probabilístico subyacente asumido. Por ejemplo, cuando se usa *k-Medias*, se debe tener en cuenta que el método está diseñado para construir grupos esféricos de aproximadamente igual tamaño, y por lo tanto, este método no es confiable cuando los grupos que se buscan se alejan fuertemente de este supuesto.

Entonces, para comprender los métodos de clustering y decidir cual de estos métodos se deberían aplicar a un caso particular, es de interés el determinar modelos apropiados y desarrollar métodos especialmente diseñados para esos modelos.

Otra problemática de interés actual es el impacto negativo y distorsionante que tienen los *outliers* en los procedimientos de Clustering. Sin embargo, sin especificar un modelo no es claro qué se entiende por una observación siguiendo un comportamiento "anómalo". Por está razón, el presente trabajo toma como punto de partida al Clustering basados en modelos. Por ejemplo, no es claro cuando un conjunto de observaciones muy dispersas pueden ser visto como un grupo o como meramente ruido de fondo

a ser eliminado. Adicionalmente, no es obvio si un pequeño grupo de outliers muy "juntos" deberían ser considerados como un grupo propio en vez de un fenómeno de contaminación.

Por estas razones, se tratarán procedimientos de Clustering en un paradigma "*model-based*" bajo fenómenos de contaminación en la muestra, tanto desde el punto de vista de la consistencia como de la robustez.

En esta instancia, al comienzo de la investigación, caben las interrogantes de "¿Existe alguna razón para favorecer un algoritmo sobre otro?", "¿Existe algún procedimiento universal, que sea el "mejor", frente a cualquier estructura de los datos?", "¿Existe algún resultado fundamental que se cumpla sin importar la inteligencia del diseñador, el número y distribución de los patrones, y la naturaleza de la tarea de clasificación?" Y en caso afirmativo, "¿Porqué?".

Estas preguntas conciernen a las cimientos del reconocimiento de patrones estadístico.

Duda y Hart en su trabajo *Pattern classification* (2001) [15] concluye que *Independientemente del problema, sin realizar algún supuesto, ningún método de clasificación de patrones es inherentemente superior a algún otro, aún a la asignación aleatoria.*

Los pilares de ésta afirmación son los teoremas de *"No hay Almuerzo Gratis"* y el teorema del *"Patito feo"* que hablan sobre la carencia de superioridad inherente de cualquier clasificador y que frente la ausencia de supuestos no existe una "mejor" representación de las características, y que aún la noción de similaridad entre patrones depende implícitamente en los supuestos, los cuales pueden o no ser correctos.

El Teorema *"No hay Almuerzo Gratis"* justifica el escepticismo acerca de estudios que se proponen demostrar la superioridad de un algoritmo particular de aprendizaje o reconocimiento sobre el resto.

El teorema del *"Patito feo"*[1] fuerza a reconocer que aún la aparentemente simple noción de similaridad entre patrones esta fundamentalmente basada en supuestos implícitos acerca del dominio del problema.

Como consecuencia, en el momento de comparar las diversas técnicas presentadas **siempre** se debe tener claramente explicitados los supuestos sobre la tipología de los datos y que las conclusiones extraídas no se podrán extender a otras formas de modelado de datos (de aquí la elección de la perspectiva basada en modelos en el trabajo).

[1]Nombrado por la famosa historia de Hans Christian Andersen, "El Patito Feo". Se debe a que el teorema muestra que, si todas las cosas son iguales, un patito feo es tan similar a un cisne como dos cisnes son entre ellos.

Como se intenta proporcionar herramientas de Clustering en diversas situaciones que se comporten de forma estable frente a distintos tipos de perturbaciones sobre el modelo de los datos, es necesario presentar variadas técnicas y enfoques que funcionen de forma más eficiente una con respecto a la otra en diferentes paradigmas.

En el capítulo 1, se introducirán los conceptos generales de robustez cualitativa siguiendo la línea que comenzó desarrollando Hampel en sus trabajos de 1971 y 1974 [13] [14]. Cabe destacar que estadística robusta no se imparte actualmente en los cursos de grado de la Licenciatura en Estadística, por tanto detenta este capítulo un interés en sí mismo.

La distancia de Prokhorov, el punto de quiebre y la curva de influencia serán formalmente definidos, lo que permitirá en capítulos futuros manejar con mejor claridad estos conceptos, así como también evaluar los estimadores respecto a sus medidas de robustez. Trabajos recientes, como los de Yohai (2006) [23], Jureckova (2006) [20] y Huber (2009) [32] aportan mayor "luz" a este capítulo.

El capítulo 2 delinea qué se entenderá por datos atípicos, también llamados *outliers*, dando una definición formal de dicho concepto.

Se describen distintos métodos de detección o identificación de outliers intentando brindar sus fortalezas y debilidades. Por otro lado, se describen los efectos negativos que puede producir, en las técnicas de Clustering, la no detección de éstos o el trabajo con estimadores que no están diseñados para soportar perturbaciones en el modelo.

Puesto que en este trabajo se utiliza recurrentemente el estimador de *Determinante de Covarianza Mínima (MCD)* desarrollado Rousseeuw en 1985 [41] y una variante computacionalmente más eficiente, el *fast-MCD* introducido por Rousseeuw, Katrien y Van Driessen en 1999 [5], se tratarán éstos con mayor atención, precisando sus propiedades de consistencia y sus cualidades robustas.

Los capítulos siguientes, 3, 4 y 5, presentan tres técnicas de Clustering en presencia de outliers:

- k-Medias robustas.
- Mezcla de distribuciones t.
- Clusters podados.

Si bien cada método presenta sus ventajas en distintos tipos de modelado de datos, se intenta a medida que se avanza en el trabajo, ir utilizando menos supuestos sobre la distribución de éstos, la tipología de outliers, el porcentaje de outliers y sobre el

número de grupos. Pero, como ya se fundamentó, no es posible liberarse absolutamente de todos estos supuestos.

El método de k-Medias es tratado en el capítulo 3, desarrollado inicialmente por McQueen (1965) y es sustentado por los trabajos de Pollard (1981, 1982) [33] [34]. Presenta alta eficiencia cuando los datos presentan distribuciones esféricas (desarrollado en el Apéndice A), pero es altamente sensible a outliers. En el trabajo se estiman los centros de los grupos, no minimizando respecto a la distancia euclídea, sino respecto a otra métrica acotada, de forma de volverlo más estable frente a perturbaciones en el modelo.

En el capítulo 4 se intenta levantar el supuesto de distribuciones esféricas, modelando los datos a partir de una mezcla de distribuciones. Si bien la mezcla de distribuciones normales es el camino clásico, McLachlan y Peel (2000) [30] y McLachlan (2006) [28] proceden a través de una mezcla de distribuciones t, ya que las colas más pesadas de está distribución son capaces de soportar (débilmente) de forma más estable la presencia de outliers. Este método sustenta su consistencia en el algoritmo EM (desarrollado en el Apéndice B) y presenta alta performance en presencia de distribuciones elípticas (ver Apéndice A).

Si el porcentaje de contaminación es elevado o estos datos presentan "patologías" particulares, el método anterior claudica. Una alternativa para resolver este tipo de problemas es expuesta en el capítulo 5. Los pilares de este capítulo se encuentran en las investigaciones de Cuesta, Matrán, Mayo e Iscar (2008) [24] y Gallegos y Ritter (2009) [39], las cuales abordan la problemática de outliers mediante el "podado" (*trimming*). Ambos trabajos modelan distribuciones elípticas con dispersiones heterogéneas, pero difieren en las restricciones impuestas a la matrices de varianzas y covarianzas.

En el capítulo 6 se simulan distintas distribuciones para los datos así como también distintas formas y porcentajes de contaminación, se comparan las técnicas antes expuestas, obteniendo diversas conclusiones.

Finalmente, a partir de datos sobre precios de ventas de casas y metros de construcción, en el capítulo 7 se aplican los diferentes algoritmos para detectar clusters y outliers a un problema práctico concreto.

El capítulo 8 se resumen las conclusiones obtenidas y se realizan comentarios sobre posibles investigaciones futuras.

Capítulo 1

Robustez

Las primeras ideas sobre estadística robusta aparecieron en un trabajo de Tukey [44], pero es a partir de una serie de trabajos influyentes de Huber en los años 60 [17] donde comienza la formalización y el desarrollo sistemático de este enfoque estadístico. Posteriormente, Hampel introduce la noción de Robustez Cualitativa en el año 1971 [13] y la de Función de Influencia en 1974 [14], cimentado las bases de este paradigma.

El objetivo de este capítulo es introducir conceptos básicos de estadística robusta, siguiendo el enfoque cualitativo de Hampel [13], [14], con el fin de analizar posteriormente las propiedades robustas de los métodos de clustering a tratar. Trabajos recientes de Maronna, Douglas Martin, Yohai (2006) [23] , Jurcková, Picek (2006) [20] y Huber, Ronchetti (2009) [32], ayudaron a delinear de manera más precisa este capítulo.

1.1. Introducción

Es sabido que dada una población normal con media μ, el estimador IMVU (insesgado y de mínima varianza uniformemente) y MINIMAX, es la media de la muestra. Sin embargo en la mayoría de las aplicaciones prácticas uno puede - a lo sumo - asegurar que los datos provienen de una distribución aproximadamente normal. En este caso cabe preguntarse cuál es el comportamiento del estimador $\bar{x}$.

Una forma sencilla de abordar este problema es dotar a la distribución normal, Φ, de un entorno de contaminación, H, de tamaño ϵ, a través de una mezcla:

$$F = (1 - \epsilon)\Phi + \epsilon H. \tag{1.1}$$

Es fácil observar que si la varianza de la distribución H es alta, la media muestral se vuelve un estimador poco eficiente [32] [23].

Otra manera de ejemplificar el problema es imaginar que, de los n datos observados, se lleva uno al infinito (se lo convierte en un outlier), lo que trae como consecuencia que la media muestral se vaya al infinito. Esto no sucede con la mediana, por lo que diremos que la mediana es un estimador más robusto que la media. Por tanto, en estos casos se necesita estimadores lo suficientemente estables a estos datos anómalos sin perder demasiada eficiencia.

En términos generales, y desde un punto de vista informal, se puede decir que un estimador $\hat{\theta}(F(n))$ es robusto si su comportamiento es relativamente bueno y estable cuando F varía "cerca" del modelo teórico asumido (sobre el entorno $\mathcal{F}_\theta$ del modelo paramétrico F_θ). En otras palabras el estimador $\hat{\theta}(F(n))$ debe poseer las siguientes dos propiedades [47]:

Eficiencia $\hat{\theta}(F(n))$ se comporta bien cuando el modelo central $F = F_\theta$ se cumple.

Estabilidad El buen comportamiento de $\hat{\theta}(F(n))$ se preserva cuando F varia sobre $\mathcal{F}_\theta$.

La formalización de este último concepto a dado lugar a varios enfoques en la teoría de la Robustez: Robustez Cualitativa, Robustez Cuantitativa y Robustez Infinitesimal. Se presenta una breve descripción de la primer rama, la cual es un insumo sustancial de la monografía.

1.2. Robustez Cualitativa

El concepto fue iniciado por Hampel (1971) [13], para el cuál se han también presentado diferentes definiciones alternativas (i.e. Fraiman, Boente (1987) [1]).

Se dirá, a grandes rasgos, que el funcional $\hat{\theta}(F)$ es cualitativamente robusto si $\hat{\theta}$ es continuo en una cierta manera. Robustez Cualitativa es una propiedad muy básica, por tanto, estimadores que no posean dicha propiedad pueden ser descartados desde el punto de vista de la robustez.

Se asume que el proceso generador de observaciones bajo consideración, puede ser aproximadamente descrito por algún modelo paramétrico, y se quiere estimar los parámetros de dicho modelo. Pero se sabe que el modelo paramétrico no es exactamente el verdadero. Entonces, se requiere que la distribución del estimador cambie levemente si la distribución de las observaciones es tenuemente alterada del modelo paramétrico estricto.

Se pueden tener los siguientes tipo de desviaciones [13] del modelo:

1. Redondeo de las observaciones;
2. Ocurrencia de errores grandes;
3. El modelo en sí mismo puede ser sólo una aproximación al modelo subyacente.

Una distancia que cuantifica este tipo de desviaciones es la distancia de Prokhorov, que, además de poseer la propiedad de metrizar la convergencia en distribución, puede ser definida en espacios muy generales. Por lo tanto, se necesita que la distribución del estimador sea un funcional continuo - en la distancia de Prokhorov - de la distribución subyacente en el modelo paramétrico. Cuando se habla de un estimador, se tiene en mente una sucesión de estimadores, que cuando n tiende a infinito, la verdadera distribución subyacente tiene que estar muy cerca del modelo paramétrico. De ésta forma, se mantendría la distribución del estimador cerca de la cual habría sido bajo este modelo. Para ello se requiere la continuidad uniforme, la que se utilizará para la definición de estimador robusto.

1.2.1. Distancia de Prokhorov

Una manera de cuantificar los errores de redondeo - y a su vez los errores grandes - es a través de la métrica de Prokhorov.

Definición 1 (Métrica de Prokhorov) *Sea $(\Omega, \mathcal{A})$, siendo Ω un espacio métrico separable y completo, y $\mathcal{A}$ la σ-algebra generada de Borel; y sean P,Q probabilidades sobre $(\Omega, \mathcal{A})$. Dado $A \in \mathcal{A}$, $A^\epsilon = \{y \in \Omega : d(y, A) < \epsilon\} = \cup_{x \in A} B(x, \epsilon)$, la distancia de Prokhorov se define como*

$$\prod_d(P, Q) = \text{ínf}\{\epsilon > 0 : P(A) \leq Q(A^\epsilon) + \epsilon \quad \forall\, A \in \mathcal{A}\} \tag{1.2}$$

Se muestran algunas propiedades que revelan las bondades de ésta métrica.

Propiedades

Caracterización Metriza la convergencia débil. [36]

Acotación $0 \leq \Pi_d \leq 1$.

Invariancia Es invariante bajo transformaciones de escala.

Propiedad de Strasseen $\Pi_d(P,Q) \leq \delta \Leftrightarrow \exists\, R$ una medida de probabilidad en $\Omega \times \Omega$ con marginales P y Q tales que $R(D(\delta)) \geq 1-\delta$ siendo $D(\delta) = \{(w,w') : d(w,w') \leq \delta\}$. [43]

Distancia para una mezcla $(1-\delta)P + \delta H = Q \rightarrow \Pi_d(P,Q) \leq \delta$.

1.2.2. Sucesiones robustas

Se definirá a una sucesión de estimadores como robusta en una medida de probabilidad F_0 si, y sólo sí, $\forall \epsilon > 0$, $\exists\, \delta > 0 \,\forall\, G$ tal que si

$$\Pi_d(F_0, G) < \delta \rightarrow \Pi_d(\mathcal{L}_{F_0}(T_n), \mathcal{L}_G(T_n)) < \epsilon \,\forall\, n. \tag{1.3}$$

Una sucesión de estimadores $\{T_n\}$ es continua en F si, y sólo si,

$$\begin{aligned} \forall\, \epsilon > 0, \exists\, \delta > 0, \exists\, n_0 \forall\, n, m \geq n_0, \forall\, F_n, F_m : \\ F_n \in \mathcal{F}_n \text{ y } F_m \in \mathcal{F}_m \text{ y } \Pi(F, F_n) < \delta \\ \text{y } \Pi(F, F_m) < \delta \Rightarrow |T_n(F_n) - T_m(F_m)| < \epsilon \end{aligned}$$

Hampel (1971) [13] concluye los siguiente teoremas:

Teorema 1 *Sea la sucesión de estimadores $\{T_n\}$ que cumple:*

- *T_n es continua como función puntual de Ω^n para cualquier n*
- *$\{T_n\}$ es continua en F.*

Entonces $\{T_n\}$ es robusta en F.

Teorema 2 *Sea $T : \mathcal{F} \rightarrow \mathbb{R}^k$ y $\{T_n\}$ es definida por $T_n \equiv T|\mathcal{F}_n \,\forall\, n$. Entonces las siguientes condiciones son equivalentes:*

1. *T es continua para cada F*
2. *$\{T_n\}$ es robusta y consistente, tendiendo a $T(F)$ para cada F*
3. *$\forall K \subset \mathcal{F}$, K relativamente compacto, $\forall\, \epsilon > 0, \exists\, \delta > 0$, $\forall F \in \mathcal{F}$, $\forall G \in \mathcal{F}$:*

$$\{F \in K, \Pi(F,G) < \delta \Rightarrow |T(F) - T(G)| < \epsilon\}$$

Una definición alternativa de Robustez Cualitativa ha sido propuesta por Fraiman y Boente (1987) [1], basado en el concepto de resistencia introducido por Tukey (1975) [45] que parece capturar mejor su concepto intuitivo. En vez de considerar la insensibilidad de los estimadores con respecto a pequeños cambios el la distribución del proceso, se considera cuán insensible son en un punto de la muestra dada, $x \in X^{\infty}$, cuando

1. todas las observaciones tienen pequeños cambios,

2. o una pequeña fracción de observaciones sufre grandes cambios.

Huber (64) define el máximo sesgo asintótico de un estimador, pero este es en general complejo de calcular. Por tanto se necesitan otras medidas relacionadas que indiquen acerca de la robustez del estimador.

Algunas de ellas son el Punto de Quiebre, la Sensibilidad de Contaminación o la Curva de Influencia de un estimador.

1.3. Punto de Quiebre

El Punto de Quiebre es una medida popular de la robustez de un estimador frente a observaciones atípicas. Informalmente, indica la menor proporción de datos contaminantes en una muestra que causa que el estimador "se quiebre", esto es, que tome valores arbitrariamente malos o sin sentido. Es definido por Hampel en 1971 [13] para analizar la robustez de funcionales de locación de distribuciones univariadas. Este trabajo pionero se concentró en un enfoque funcional para analizar estimadores como una función de las distribuciones poblacionales. Diez años después, Donoho (1982) propuso versiones para muestras finitas, iniciando una extensa literatura en la estimación de alto punto de quiebre, la cual ha utilizado para la estimación de locación, dispersión y modelos de regresión.

Desde un punto de vista más formal, el punto de quiebre indica hasta que distancia de Prokhorov del modelo paramétrico el estimador todavía provee alguna información de la distribución original del modelo paramétrico, en el sentido de excluir una parte del espacio de los parámetros.

Definición 2 (Versión poblacional) *Sea $\{T_n\}$ una sucesión de estimadores. El Punto de Quiebre δ^* de $\{T_n\}$ en alguna medida de probabilidad F es definido de la siguiente manera:*

$\delta^* = \delta^*(\{T_n\}, F) = sup\{\delta \leq 1 : \exists$ *un conjunto compacto* $K = K(\delta)$*, que es un subconjunto propio del espacio de los parámetros, de tal manera que* $\Pi(F, G) < \delta \Rightarrow G\{T_n \in K\} \to 1$ *si* $n \to \infty\}$

Definición 3 (Versión muestral) *Sea* $X_n = (x_1, \ldots, x_n)$ *una muestra aleatoria y* T_n *un estimador que puede depender de la misma. Sea* $X_{n,m}$ *el conjunto obtenido por remplazar* m *puntos* $x_{i_1}, \ldots, x_{i_m}$ *de* X_n *por los valores arbitrarios* $y_1, \ldots, y_m$*. Para medir la diferencia entre el estimador aplicado a* X_n *y a* $X_{n,m}$*, se necesita una medida de distancia apropiada,* D*, en el espacio paramétrico* Θ*, al cual pertenece* $T_n(X_n)$*.*

El Punto de Quiebre para muestras finitas de T_n *en* X_n *cuenta la fracción más pequeña de contaminantes,* m/n*, para la cual la distancia entre* $T_n(X_n)$ *y* $T_n(X_{n,m})$ *puede convertirse arbitrariamente grande:*

$$\epsilon_n^*(T_n; X_n) = \frac{1}{n} \text{mín} \left\{ m \in \{1, \ldots, n\} : \sup_m D(T_n(X_n), T_n(X_{n,m})) = +\infty \right\}. \quad (1.4)$$

Estrictamente, el Punto de Quiebre también depende de D, pero para simplificar la notación, se escribe $\epsilon_n^*(T_n; X_n)$, teniendo en cuenta que la métrica D de cumplir que $\sup_{\theta_{1,\theta_2}} D(\theta_1, \theta_2) = +\infty$.

Aunque el Punto de Quiebre de un estimador es una herramienta que indica si un estimador es malo desde el punto de vista de la robustez, *no* nos dice que tan bueno es. Por tanto, otras medidas se tienen que tener en cuenta también, como por ejemplo, la Curva de Influencia.

1.4. La curva de influencia

Hampel (74)[14] introduce uno de los conceptos claves de la estadística robusta a nivel local: *la Curva de Influencia.*

La Curva de Influencia de un estimador es esencialmente la primer derivada de un estimador, visto como un funcional, en alguna distribución (en un espacio de dimensión infinita). La curva mide el efecto infinitesimal de los outliers en el estimador.

El estudio de Curvas de Influencia sirve para profundizar el entendimiento de estimadores, por ejemplo de la relación entre medias podadas, medias Winsorizadas y estimadores de Huber. También sirven para derivar nuevos estimadores robustos con determinadas propiedades deseadas.

Sea $\mathbb{R}$ la recta real, sea T un funcional de valores reales definido en algún subconjunto de todas las medidas de probabilidad en $\mathbb{R}$, y sea F la medida de probabilidad en $\mathbb{R}$

para la cual T está definido. Se denota por δ_x la medida de probabilidad determinada por el punto de masa 1 en cualquier punto dado $x \in \mathbb{R}$.

Definición 4 (Curva de Influencia) *Dado $0 < \epsilon < 1$, la Curva de Influencia $IC_{T,F}(\cdot)$ del estimador T en la distribución de probabilidad subyacente F esta definida punto a punto por*

$$IC_{T,F}(x) = \lim_{\epsilon \to 0} \frac{T((1-\epsilon)F + \epsilon\delta_x) - T(F)}{\epsilon},$$

si este límite está definido para cada punto $x \in \mathbb{R}$.

Por ejemplo, la media aritmética $T = \int x\, dF(x)$ está definida para todas las medidas de probabilidad con primer momento. Se supone que la media de F existe y es igual a μ. Entonces la Curva de Influencia de T está definida en F y está dada por

$$IC_{T,F}(x) = \lim_{\epsilon \to 0} \frac{(1-\epsilon)\mu + \epsilon x - \mu}{\epsilon} = x - \mu \quad (x \in \mathbb{R}).$$

La varianza $T = \int (x-\mu)^2 dF$ en una F con varianza finita σ^2 y media conocida μ tiene función de influencia dada por

$$IC_{T,F}(x) = \lim_{\epsilon \to 0} \frac{(1-\epsilon)\sigma^2 + \epsilon(x-\mu)^2 - \sigma^2}{\epsilon} = (x-\mu)^2 - \sigma^2 \quad (x \in \mathbb{R}).$$

En general, si tenemos Ω un espacio métrico polaco, sea T una función vectorial de un subconjunto del espacio de medida Ω en $\mathbb{R}^k$ y sea F en el dominio de T. Sea δ_ω la medida atómica de probabilidad concentrada en $\omega \in \Omega$. Entonces la Curva de Influencia de T en F es definida por

$$IC_{T,F}(\omega) = \lim_{\epsilon \downarrow 0} \frac{T[(1-\epsilon)F + \epsilon\delta_\omega] - T(F)}{\epsilon} \tag{1.5}$$

1.4.1. Ejemplos

¿Qué sucede con la media y al varianza de una muestra de observaciones reales de una variable si se agrega un valor a ellas?. Es fácil observar que el error cometido produce un efecto lineal en la media y un efecto de orden cuadrático en la varianza, por lo tanto, se tiene que plantear estimadores que sean menos influenciados por estas modificaciones en los datos. Algunos ejemplos de esto son:

La media α-recortada Es el clásico estimador sustituto de la media aritmética, que consiste en recortar las αn más pequeñas y αn más grandes de las observaciones.

Es sencillo observar que si yo agrego una observación el valor se afecta a lo sumo en una constante.

Si definimos la α-media recortada para cualquier distribución F es definida por

$$\bar{x}_\alpha(F) = \int_\alpha^{1-\alpha} F^{-1}(t)dt/(1-2\alpha),$$

obtenemos la siguiente Curva de Influencia para distribuciones unimodales simétricas respecto de 0:

$$IC_{\bar{x}_\alpha,F}(x) = \begin{cases} F^{-1}(\alpha)/(1-2\alpha) & \text{si } x < F^{-1}(\alpha) \\ x/(1-2\alpha) & \text{si } F^{-1}(\alpha) \leq x \leq F^{-1}(1-\alpha) \\ F^{-1}(1-\alpha)/(1-2\alpha) & \text{si } x > F^{-1}(1-\alpha) \end{cases}$$

La media winsorizada No descarta los outliers sino que les asignas el valor de los datos extremos no outliers. Es decir, reemplaza las $[\alpha n]$ observaciones extremas por $x^{(h+1)}$ y $x^{(n-h)}$ respectivamente. La definición asintótica equivalente es dada por

$$\int_\alpha^{1-\alpha} F^{-1}(t)dt + \alpha[F^{-1}(\alpha+0) + F^{-1}(\alpha+0)]$$

En el capítulo siguiente se comienza por definir que es un outliers en este trabajo y se desarrollarán métodos de detección de los mismos.

Capítulo 2

Outliers

> Aquel que conozca los pasos de la Naturaleza notará más fácilmente sus desviaciones; y por otro lado, quien conozca sus desviaciones podrá describir más precisamente sus pasos.
>
> *Sir Francis Bacon*

La definición de lo "típico" implica la definición de lo "atípico", y vice–versa.

Esto implica que el problema sea de naturaleza arbitraria, por lo que hay numerosas definiciones de outliers, tanto en la literatura estadística como en las de aprendizaje automático (Machine Learning).

Desde la estadística, una definición comúnmente utilizada es que los "outliers" son una minoría de observaciones en un conjunto de datos que tienen patrones diferentes al de la mayoría de las observaciones en el conjunto. A veces se los refiere como datos "anómalos" o "atípicos", haciendo énfasis en la posible fuente de los mismos.

2.1. Introducción

Una definición exacta de un outlier se formula a partir de los supuestos, generalmente ocultos o implícitos, con respecto a la estructura o el proceso generador de datos, y el método de detección aplicada.

Definiendo el proceso generador de los datos - elección que es arbitraria por parte del investigador - se elije un método que "distinga" las puntos provenientes del mismo de alguna forma. Es esta distinción la que define exactamente lo "típico" - y por ende lo "atípico" - para ese problema en particular.

Un modelo distinto y/o un método de detección distinto conllevan a diferentes definiciones concretas de outliers, por lo que en concreto, hay tantas definiciones de outliers posibles como combinaciones de modelos y métodos posibles: *infinitas*.

Aún así, algunas definiciones son consideradas lo suficientemente generales para enfrentarse con varios tipos de datos y métodos. Hawkings (1980) define un outlier como *una observación que se desvía mucho de otras observaciones, que resulta sospechosa de haber sido generada por otro mecanismo*. Barnett y Lewis (1994) indican que *una observación outlier es una que aparece desviarse marcadamente de otros miembros de la muestra en la cual ocurre*, similarmente, Johnson (1992) define un outlier como *una observación en un conjunto de datos que parece ser inconsistente con el resto del conjunto de datos*.

El supuesto que subyace en esta definición es que por lo menos el 50 % de las observaciones en el conjunto de datos es homogéneo (en el sentido de estar representados por el mismo patrón) y las restantes observaciones tienen patrones inconsistentes con el patrón "principal".

Conciencia sobre este tipo de datos ha existido por lo menos desde hace cientos de años. En 1620, Sir Francis Bacon, pionero en el desarrollo de lo que posteriormente se llamara "método científico", realizaba la frase con la que se empieza el capítulo. Adrien-Marie Legendre, explícitamente mencionaba el desestimar los outliers para mejorar la exactitud y reducir el error: "*Si entre esos errores hay algunos que parecen muy grandes para ser admisibles, entonces esas ecuaciones, que produjeron esos errores, serán rechazadas como viniendo de experimentos muy defectuosos, y las cantidades desconocidas determinada por las otras ecuaciones, darán un error mucho más pequeño*".

De esta forma, el rechazo o descarte de outliers previo a realizar un análisis estadístico clásico ha sido considerado como un paso esencial de pre-procesamiento para la mayoría desde los inicios de los métodos de inferencia. Si bien la inspección visual a través de gráficos puede ser efectiva cuando la dimensión de los datos es menor o igual a 3, este método no funciona para datos en dimensión mayor, debido a que la inspección de cada dimensión o pares de éstas puede conducir a conclusiones erróneas, ya que una observación puede ser outlier en el espacio multivariado pero no en cualquier subconjunto de dimensiones tomadas aisladamente. La detección automática de outliers puede ser dificultosa, debido a los conocidos efectos de enmascaramiento y hundimiento, que posteriormente se desarrollarán.

Con frecuencia la identificación de outliers suele ser el objetivo principal de la investigación. Los outliers en sí mismos son puntos de interés primario, revelando

aspectos desconocidos de los datos, o llevando a descubrimientos inesperados. Por ejemplo, Rayleigh descubrió el argón, un elemento de la tabla periódica, a partir de la observación de diferencias inesperadas o atípicas en medidas de nitrógeno. Las detección de intrusos en redes de computadoras pueden ser vistos como outliers, donde el intruso exhibe una combinación de características que, conjuntamente consideradas, difieren de los usuarios típicos de la red. Perpetradores de fraudes con tarjetas de crédito proveen otro ejemplo donde la identificación de outliers es crítica, siendo necesario el análisis de base de datos de transacciones con el propósito específico de detectar transacciones inusuales.

Estos ejemplos demuestran la importancia del análisis de outliers, teniendo en cuenta no sólo la precisión del método, sino también la eficiencia computacional del mismo.

2.2. Identificación de Outliers y Métodos Robustos

Muchos conjuntos de datos, especialmente los grandes, contienen outliers y subgrupos de datos "atípicos". Esto puede llevar a problemas ya que el análisis estadístico clásico asume que los datos son homogéneos y libre de outliers. Todos los métodos de estadística clásica (e.g. análisis discriminante, análisis factorial, análisis de componentes principales, modelos de regresión) pueden ser severamente distorsionados por la presencia de los mismos.

Esta distorsión, que puede llevar al punto de obtener conclusiones opuestas a las verdaderas, se debe fundamentalmente a que estos métodos residen sobre las estimaciones de la media y la matriz de covarianza "clásicas", las cuales son extremadamente sensibles a outliers. De esta forma, los intervalos o regiones de confianza, así como las estimaciones de los parámetros del modelo, vía los efectos de enmascaramiento y hundimiento, pueden llegar a ser "destruídos".

Efecto Enmascaramiento Se dice que un outlier enmascara un segundo outlier, si el segundo outlier puede ser considerado como un outlier sólo por sí mismo, pero no en la presencia del primer outlier. Entonces, después de eliminar el primer outlier, el segundo emerge como outlier. El enmascaramiento ocurre cuando observaciones outliers afectan las estimaciones de la media y la covarianza hacia él, y la distancia resultante del outlier de la media es pequeña. Se da un fenómeno de "falso negativo", donde las outliers pueden pasar como no identificados.

Efecto Hundimiento Se dice que un outlier *hunde* una segunda observación, si la segunda puede ser considerada como un outlier sólo bajo la presencia del primero. En otras palabras, después de eliminar el primer outlier, la segunda observación se convierte en no-outlier. El hundimiento ocurre cuando un grupo de outliers afectan la estimación de la media y la covarianza hacia ellos y lejos de otras observaciones no-outliers, y la distancia resultante de esas observaciones a la media es grande, haciendo parecerlas como outliers. Se trata de un fenómeno opuesto al de "enmascaramiento", donde se detectan "falsos positivos".

Estos efectos llevan a estimaciones incorrectas de los parámetros del modelo paramétrico pues no se detectan correctamente los outliers, lo que lleva a la re-estimación defectuosa de los parámetros.

Por tanto la identificación precisa de los outliers antes de realizar un análisis estadístico clásico es de vital importancia si se pretende sacar conclusiones confiables. A su vez, expresa el problema en términos análogos al dilema de "¿Primero el huevo o la gallina?", ya que un outlier siempre se define en relación a un modelo, no existen outiliers *per se* (para seguir ejemplificando, un dato puede ser atípico para una distribución normal estándar pero no para una distribución de Cauchy). Pero el modelo, desconocido a priori, se estima a partir de las observaciones.

Para dar un "paso base" a este problema recursivo, existen dos enfoques amplios para el problema de identificación de outliers: estimación robusta y métodos enfocados específicamente en la identificación de outliers

Víctor Yohai hace un paralelismo de estos enfoques con la seguridad informática en en su discurso de investidura Doctor Honoris Causa (2006) diciendo, "*En este caso hay dos estrategias posibles, tratar de detectar la presencia de hackers*[1] *en la red y encarcelarlos de manera que no puedan actuar. Este sería un enfoque similar a la detección de atípicos. El segundo enfoque sería construir sistemas operativos tan seguros que ningún hacker*[2] *podría dejarlos actuar libremente sabiendo que no representan un peligro porque sus intentos de penetrar la red fracasarán. Este sería un enfoque similar al de los procedimientos robustos cuando eliminan el efecto de los atípicos*".

Los métodos de estimación robusta están diseñados para dar resultados óptimos en la presencia de outliers, pero para esto, sacrifican la alta eficiencia estadística. Una vez estimados los parámetros por estos métodos, los mismos se utilizan para clasificar a las observaciones como outliers.

[1] La palabra "hacker" es utilizada aquí como sinónimo de intruso.
[2] Idem.

Los métodos específicos para la identificación de outliers, una vez que los identifican, se le quita peso a tales observaciones (o simplemente se las elimina) en el proceso de estimación de los parámetros. De esta forma, aplicando estimadores clásicos, se obtienen estimaciones robustas de los parámetros.

Se pueden también distinguir métodos basados en búsqueda de direcciones (projection pursuit) y métodos basados en distancias robustas.

La idea de la búsqueda de direcciones es encontrar proyecciones en un espacio de dimensión menor que revelen información útil o alguna estructura. En este espacio, al ser examinado por el investigador, se puede utilizar el "don del ser humando para el reconocimiento de patrones", en las palabras de Friedman y Tukey, quienes introdujeron el método en 1974 [8].

El enfoque presenta el inconveniente de que no siempre es fácil, o aún posible, encontrar una dirección que devele los outliers. El caso más conocido de búsqueda de direcciones es el Análisis de Componentes Principales.

El estimador de Stahel-Donoho [4] es uno de los primeros métodos robustos para identificar outliers y se encuentra en este enfoque.

Definido a partir de una media y una matriz de covarianzas ponderadas, donde cada observación es inversamente proporcional a "cuán oultiler es". La medida de "cuán outlier es" está basada en la idea de una "búsqueda" en todas las direcciones "posibles", ya que si una observación es un oultlier multivariada, entonces debe existir alguna proyección unidimensional en la cual el punto es revelado claramente como outlier. El estimador de Stahel-Donoho representa un buen enfoque intuitivo para identificar outliers, pero en la práctica es muy costoso computacionalmente, por lo que se han desarrollado otros métodos, como el de Kurtosis1 de Peña y Prieto para acelerar su cálculo [35]. Trabajos recientes han podido especificar la curva de influencia del estimador [18], [10].

2.2.1. Métodos basados en distancias

Una alternativa a la estrategia de búsqueda de direcciones es el enfoque basado en la distancia de Mahalanobis. Estos tienen fundamentalmente dos etapas:

1. Obtener estimadores robustos para la media y la matriz de covarianzas, T y C, de forma que la versión robusta de la distancia de Mahalanobis, $d_i(T, C)$, pueda ser calculada para punto x_i. Esto provee una medida de cuán lejos está cada punto del centro robusto T, de acuerdo con la escala robusta de los datos, C.

2. Determinar la cota de separación, D, de forma tal que los puntos con $d_i > D$ son declarados como outliers. Si los datos provienen de una distribución normal con media T y matriz de covarianza C, entonces la distancias al cuadrado, d_i^2, se aproximan a una distribución χ^2 con p grados de libertad. Entonces, una cota adecuada sería el quantil superior de una distribución χ^2. Sin embargo, T y C son elegidas usualmente vía estimadores robustos, por lo que la distribución χ^2 puede no ser apropiada, ni siquiera aproximada. Elegir una cota apropiada no es un tema trivial, pero en la ausencia de alternativas precisas y de fácil cálculo, muchos practicantes elijen (no siempre de manera adecuada) la distribución χ^2.

Ejemplos de estos métodos son el estimador de Elipsoide de Mínimo Volumen (MVE)[40] y el estimador de Determinante de Covarianza Mínimo (MCD)[41], este último se tratará en detalle.

2.3. Determinante de Covarianza Mínima (MCD)

2.3.1. Introducción

El estimador de Determinante de Covarianza Mínima (MCD en adelante) es un estimador robusto de locación y escala multivariada. Propuesto por Rousseeuw en 1985 [41], su uso se "popularizó" cuando junto a Van Driessen introdujeron el algoritmo Fast-MCD [5] para calcularlo eficientemente. Ya que estimar la matriz de covarianza es la piedra angular de muchos métodos estadísticos multivariados, el MCD ha sido usado para desarrollar técnicas multivariadas robustas y eficientes computacionalmente.

Se asume que se cuenta con n datos, donde cada uno tiene p variables asociadas. Éstos se encuentran en una matriz de dimensiones $n \times p$, $X = (x_1, \ldots, x_n)^t$, con $x_i = (x_{i1}, \ldots, x_{ip})$.

La elipsoide de tolerancia clásica es definida como el conjunto de puntos p-dimensionales, x, cuya *distancia de Mahalanobis*

$$MD(x) = \sqrt{(x - \bar{x})' S^{-1} (x - \bar{x})} \tag{2.1}$$

es igual a $\sqrt{\chi^2_{p,\alpha}}$, donde α representa el cuantil de la distribución.

El objetivo del MCD es poder elegir h observaciones de n posibles de forma tal que la matriz de covarianzas de estas tenga el menor determinante posible. Cuando el número de objetos n y de variables p son altos el algoritmo (MCD) es muy costoso computacionalmente hablando, es por eso que Rousseeuw (84) introduce un procedimiento de "iteración selectiva" llamado Fast-MCD. Este procedimiento no modifica

el MCD cuando el número de datos es pequeño sino que lo hace cuando n es grande aumentando la velocidad de convergencia del algoritmo.

La distancia de Mahalanobis es una manera sencilla de detectar outliers cuando estos aparecen individualmente, pero no es un método recomendado cuando hay múltiples outliers, porque se puede producir el "efecto de enmascaramiento" donde los outliers no presentan necesariamente distancias altas de Mahalanobis.

Rousseeuw (1984) [40] introduce el método elipsoide de mínimo volumen (MVE) que es aquella de mínimo volumen que contiene h datos, donde $n/2 \leq h \leq n$. El punto de quiebre de este estimador es $(n-h)/2$.

Un método que nos permite aumentar la velocidad de convergencia del MVE, teniendo la ventaja de normalidad asintótica de los estimadores, es el MCD y manteniendo además el mismo punto de quiebre del MVE.

2.3.2. Motivación y Definición

Si bien la distancia de Mahalanobis permite detectar outliers cuando estos aparecen de forma aislada, si éstos se presentan en grupos, la sensibilidad de la media muestral y de la matriz de covarianzas muestral a los mismos, hace que este método de detección de outliers sea distorsionado por los efectos mencionados. Una posible solución a este problema es modificar las estimaciones clásicas de los estimadores de posición y dispersión por estimadores robustos, a esto apunta el método MCD. Se obtiene así una elipse de tolerancia robusta determinada por la distancia robusta

$$RD(x) = \sqrt{(x - \hat{\mu}_{MCD})' \hat{\Sigma}_{MCD}^{-1} (x - \hat{\mu}_{MCD})} \tag{2.2}$$

Los estimadores de posición y dispersión robustos $\hat{\mu}_{MCD}$ y $\hat{\Sigma}_{MCD}$ a partir de un parámetro h, $[(n+p+1)/2] \leq h \leq n$ se definen de la siguiente manera:

1. Sea $\hat{\mu}_0$ es la media de aquellas h observaciones muestrales que minimizan el determinante de la matriz de covarianza muestral.

2. $\hat{\Sigma}_0$ es la correspondiente matriz de covarianzas multiplicada por un factor de consistencia c_0.

Si trabajamos en un espacio de dimensión p el estimador MCD sólo puede ser computado si $h > p$, sino la matriz de covarianzas sería singular. Como h es al menos $(n+2)/2$ es necesario pedir que n sea al menos $2p$. No obstante, para evitar la maldición de la dimensionalidad, es recomendable que n sea mayor que $5p$.

El método MCD esta diseñado para distribuciones elípticas unimodales, es decir para densidades de la forma

$$f(x) = \frac{1}{\sqrt{|\Sigma|}} g\left((x-\mu)'\Sigma^{-1}(x-\mu)\right) \tag{2.3}$$

Propiedades sobre la distribución asintótica de los estimadores MCD son tratadas por Davies (1991). Si bien los estimadores anteriormente mencionados son altamente robustos, presentan una eficiencia relativa baja en presencia de datos que provienen de una distribución normal.

Una alternativa de retener está eficiencia sin perder la alta robustez es reponderando los estimadores,

$$\hat{\mu}_{MCD} = \frac{\sum_{i=1}^{n} W(d_i^2) x_i}{\sum_{i=1}^{n} W(d_i^2)} \tag{2.4}$$

$$\hat{\Sigma}_{MCD} = c_1 \frac{1}{n} \sum_{i=1}^{n} W(d_i^2)(x_i - \hat{\mu}_{MCD})(x_i - \hat{\mu}_{MCD})^t \tag{2.5}$$

con $d_i = \sqrt{(x-\hat{\mu}_0)^t \hat{\Sigma}_0^{-1}(x-\hat{\mu}_0)}$ y c_1 un factor de consistencia. Una elección simple y efectiva de los pesos es $W(d^2) = I(d^2 \leq q_\alpha)$ siendo $q_\alpha = G^{-1}(1-\alpha)$, con $G(u) = P_F(X^t X \leq u)$. En el caso de el modelo normal $q_\alpha = \chi^2_{p,1-\alpha}$

2.3.3. ¿Cómo Funciona el Algoritmo MCD?

El siguiente teorema muestra la esencia del algoritmo, que permite afirmar su convergencia.

Teorema 3 (Monotonía del MCD) *Consideremos el conjunto de datos* $X_n = \{x_1, x_2, \ldots, x_n\}$ *, observaciones de p-variables. Sea* $H_1 \subset \{1, 2, \ldots, n\}$ *con* $|H_1| = h$*, y sea* $S_1 = \frac{1}{h}\sum_{i \in H_1}(x_i - T_1)(x_i - T_1)^t$*, con* $T_1 = \frac{1}{h}\sum_{i \in H_1} x_i$

Si $det(S_1) \neq 0$ *definimos la distancia relativa como:*

$$d_1(i) = \sqrt{(x_i - T_1) S_1^{-1} (x_i - T_1)^t} \qquad para \quad i = 1, 2, \ldots, n.$$

Si ahora consideramos H_2 *tal que* $\{d_1(i) : i \in H_2\} := \{(d_1)_{1:n}, (d_1)_{2:n}, \ldots, (d_1)_{h:n}\}$ *donde* $(d_1)_{1:n} \leq (d_1)_{2:n} \leq \ldots \leq (d_1)_{n:n}$ *son las distancias ordenadas, y se calcula* T_2 *y* S_2 *en base a* H_2*. Entonces*

$$det(S_2) \leq det(S_1)$$

con igualdad si y solo si $T_2 = T_1$ *y* $S_2 = S_1$*.*

Corolario 1 *El subconjunto H de X_n del MCD es separado de $X_n \setminus H$ por una elipsoide.*

Se puede observar que esta elipsoide no tiene porque ser la de mínimo volumen.

2.3.4. Punto de Quiebre del MCD

En esta sección se calcula el Punto de Quiebre del algoritmo MCD tomando como referencia el trabajo realizado por Lophaa y Rousseeuw (1991) [42]. Si denotamos $X_{n,m}$ al conjunto obtenido de remplazar m datos $x_{i1}, \ldots, x_{im}$ de X_n por valores arbitrarios. Para el estimador de posición multivariado T_n el punto de quiebre es definido como:

$$\epsilon_n^*(T_n; X_n) = \frac{1}{n} \text{mín} \left\{ m \in \{1, \ldots, n\} : \sup_m \|T_n(X_n) - T_n(X_{n,m})\| = +\infty \right\}.$$

Y para el estimador multivariado de dispersión se tiene:

$$\begin{aligned} \epsilon_n^*(C_n; X_n) &= \frac{1}{n} \text{mín}\{m \in \{1, \ldots, n\} : \\ &\quad \sup_m \max_i \{|\log(\lambda_i(C_n(X_n))) - \log(\lambda_i(C_n(X_{n,m})))|\}\}, \end{aligned}$$

con $0 < \lambda_p(C_n) \leq \ldots \leq \lambda_1(C_n)$ los valores propios de C_n.

Si $k(X_n)$ denota el número máximo de observaciones en los datos que están sobre un hiperplano de $\mathbb{R}^p$. Se asume que $k(X_n) < h$ entonces el punto de quiebre de los estimadores MCD de dispersión y escala es

$$\epsilon_n^*(\hat{\mu}_0; X_n) = \epsilon_n^*(\hat{\Sigma}_0; X_n) = \frac{min(n - h + 1, h - k(X_n))}{n}.$$

Si los datos provienen de una distribución continua, entonces $k(X_n) = p$ casi seguramente y por tanto

$$\epsilon_n^*(\hat{\mu}_0; X_n) = \epsilon_n^*(\hat{\Sigma}_0; X_n) = \frac{min(n - h + 1, h - p)}{n}.$$

Notemos que $\lim_{n\to\infty} \epsilon_n^* = min(1 - \alpha, \alpha)$ es maximal cuando $\alpha = 0,5$.

2.3.5. La Curva de Influencia de MCD

Un estudio formal del mismo, es brindado en el trabajo de Croux y Haesbroeck (1999) [12].

Se denotará para $0 < \alpha < 1$

$$\mathcal{D}_G(\alpha) = \{A | A \subset \mathbb{R}^p \text{ medibles y acotados con } P_G(A) = 1 - \alpha\},$$

para cada $A \in \mathcal{D}_G(\alpha)$ la media y la matriz de covarianza computada bajo este conjunto son denotadas por

$$T_A(G) = \frac{\int_A y dG(y)}{1-\alpha} \quad y \quad \Sigma_A(G) = \frac{\int_A (y - T_A(G))(y - T_A(G))^t dG(y)}{1-\alpha}$$

y el conjunto A es llamado la solución MCD si

$$det(\Sigma_A(G)) \leq det(\Sigma_{\tilde{A}}(G)),$$

para cualquier $\tilde{A} \in \mathcal{D}_G(\alpha)$.

Se definirán entonces los estimadores MCD por

$$T(G) = T_G(A) \quad y \quad \Sigma(G) = c_\alpha \Sigma_A(G)$$

donde c_α es una constante de consistencia. Si G no es una distribución continua $\mathcal{D}_G(\alpha)$ podría ser vacío. En ese caso, se define entonces

$$\tilde{\mathcal{D}}_G(\alpha) = \{(A,x) | A \subset \mathbb{R}^p \text{ med. y acot., } x \in \mathbb{R}^p \setminus A, \\ \exists 0 \leq \delta \leq P_G(\{x\}) : P_G(A) + \delta = 1 - \alpha\},$$

Para cada $(A,x) \in \tilde{\mathcal{D}}_G(\alpha)$ se define:

$$T_{(A,x)}(G) = \frac{\int_A y dG(y) + \delta x}{1-\alpha}$$

y a $\Sigma_{(A,x)}(G)$ como

$$\frac{\int_A (y - T_{(A,x)}(G))(y - T_{(A,x)}(G))^t dG(y) + \delta (x - T_{(A,x)}(G))(x - T_{(A,x)}(G))^t}{1-\alpha}$$

Si se consideran las distribuciones elípticas y simétricas, $F_{\mu,\Sigma}$, el conjunto solución del problema MCD es única y dada por la elipsoide

$$A(F_{\mu,\Sigma}) = \{z \in \mathbb{R}^p | (z-\mu)^t \Sigma^{-1} (z-\mu) \leq q_\alpha\},$$

donde $G(t) = P_{F_{0,I}}(Z^t Z \leq t)$ y $q_\alpha = G^{-1}(1-\alpha)$.

Se anota $F_{\epsilon,x} = (1-\epsilon)F + \epsilon \Delta_x$ siendo Δ_x la delta de Dirac.

Teorema 4 *Dado $0 < \epsilon < min(\alpha, 1-\alpha)$, $x \in \mathbb{R}$ y se considera la distribución contaminada $F_{\epsilon,x}$. Para cualquier solución del MCD $(A,y) \in \tilde{\mathcal{D}}_{F_{\epsilon,x}}(\alpha)$, entonces existe una elipsoide abierta $\mathcal{E}$ tal que:*

$$\mathcal{E} \in \mathcal{D}_{F_{\epsilon,x}}(\alpha), T_{\mathcal{E}}(F_{\epsilon,x}) = T_{(A,y)}(F_{\epsilon,x}) \quad y \quad \Sigma_{\mathcal{E}}(F_{\epsilon,x}) = \Sigma_{(A,y)}(F_{\epsilon,x})$$

o, en el caso de que x se encuentre en el borde de $\mathcal{E}$,

$$(\mathcal{E}, x) \in \tilde{\mathcal{D}}_{F_{\epsilon,x}}(\alpha), T_{(\mathcal{E},x)}(F_{\epsilon,x}) = T_{(A,y)}(F_{\epsilon,x}) \quad y \quad \Sigma_{(\mathcal{E},x)}(F_{\epsilon,x}) = \Sigma_{(A,y)}(F_{\epsilon,x})$$

El siguiente teorema muestra la función de influencia de la matriz de dispersión de el MCD en su versión funcional. Para una mejor interpretación se escriben por separado los términos de la diagonal y los que no pertenecen a ésta.

Teorema 5 *Con la notación ya descrita anteriormente, se tiene que*

$$IF(x, \Sigma_{ii}, F) = \frac{1}{b_1}\left\{ \frac{c_\alpha}{1-\alpha} x_i^2 I(\|x\|^2 \leq q_\alpha) + \frac{b_2}{b_1 + pb_2}\frac{c_\alpha}{1-\alpha}\|x\|^2 I(\|x\|^2 \leq q_\alpha) \right.$$

$$\left. + \frac{b_1}{b_1 + pb_2}\left[\frac{c_\alpha}{1-\alpha}\frac{q_\alpha}{p}(1 - \alpha - I(\|x\|^2 \leq q_\alpha) - 1\right] \right\}$$

$$IF(x, \Sigma_{ij}, F) = \frac{x_i x_j}{-2c_3} I(\|x\|^2 \leq q_\alpha) \quad si \quad i \neq j,$$

donde las constantes b_1, b_2, c_2, c_3 y c_4 son determinadas por las relaciones

$$c_2 = \frac{\pi^{\frac{p}{2}}}{\Gamma(\frac{p}{2}+1)} \int_0^{\sqrt{q_\alpha}} r^{p+1} g'(r^2) dr$$

$$c_3 = \begin{cases} \frac{\pi^{\frac{p}{2}}}{(p+2)\Gamma(\frac{p}{2}+1)} \int_0^{\sqrt{q_\alpha}} r^{p+3} g'(r^2) dr & si\ p \geq 2 \\ 0 & en\ otro\ caso \end{cases}$$

$$c_4 = \frac{3\pi^{\frac{p}{2}}}{(p+2)\Gamma(\frac{p}{2}+1)} \int_0^{\sqrt{q_\alpha}} r^{p+3} g'(r^2) dr$$

$$b_1 = \frac{c_\alpha(c_3 - c_4)}{1-\alpha}$$

$$b_2 = \frac{1}{2} + \frac{c_\alpha}{1-\alpha}[c_3 - \frac{q_\alpha}{p}(c_2 + \frac{1-\alpha}{2})].$$

Observación 1 Si $p > 1$ la función de influencia puede ser escrita de forma más compacta

$$IF(x, \Sigma, F) = \frac{-1}{2c_3} x x^t I(\|x\|^2 \leq q_\alpha) + w(\|x\|) I_p,$$

donde w es cierta función real.

Observación 2 El estimador clásico de la matriz de dispersión, $C(G)$, es definido por

$$C(G) = c_0 \int_{\mathbb{R}^p} (x - \mu(G))(x - \mu(G))^t dG(x)$$

donde $\mu(G) = \int_{\mathbb{R}^p} x dG(x)$, $c_0 = E_F(X_i^2)^{-1}$ cierto factor de consistencia y X_i una componente del vector X. Notar que C está sólo definido para distribuciones con segundo momento mientras que el MCD funcional está definido para cualquier distribución arbitraria. Se puede verificar además que

$$\lim_{\alpha \downarrow 0} IF(x, \Sigma, F) = c_0 x x^t - I = IF(x, C, F)$$

Observación 3 También se puede calcular de manera mas sencilla la función de influencia para el estimador de posición del MCD,

$$IF(x, T, F) = \left(\frac{-2}{1-\alpha} \int_{z^t z \leq q_\alpha} z z^t g'(z^t z) dz \right)^{-1} \frac{x}{1-\alpha} I(\|x\|^2 \leq q_\alpha).$$

Resulta aquí importante destacar que outliers elevados no tienen influencia en la curva de posición como si lo tienen en la diagonal principal de la matriz de dispersión.

Observación 4 T_0 es equivariante bajo transformaciones ortogonales y Σ_o es equivariante afín. Un estimador de posición que también sea equivariante afín es fácil de obtener:

$$T_1(x_1, x_2, \ldots, x_n) = \Sigma_0^{1/2} T_0(\Sigma_0^{-1/2} x_1, \ldots, \Sigma_0^{-1/2} x_n)$$

T_1 hereda las propiedades deseables de robustez de T_0 y Σ_0.

2.3.6. Fast-MCD

La complejidad computacional de MCD crece de manera abrupta cuando el número de datos aumenta. Además, Adrover y Yohai (2002) muestran que al crecer la dimensionalidad aumenta el sesgo máximo asintótico del estimador[3]. Una solución al primer problema es brindada por Rousseeuw en 1999 [5], quien introdujo el algoritmo Fast-MCD. Básicamente si el número de observaciones es relativamente pequeño el algoritmo es idéntico al del MCD. Si el número de observaciones es grande ($n > 600$)

[3]Problema que se intenta solucionar mediante reducciones en la dimensión del espacio, usando proyecciones al azar, si la dimensión fuese elevada.

se divide en grupos de al menos 300 observaciones y se realiza el algoritmo MCD en cada uno de ellos una cantidad elevada de veces (500) y se elijen las 10 mejores soluciones (de determinante menor) y luego se fusionan con las otras soluciones y ahí se reitera el MCD.

Capítulo 3

k-Medias

Uno de los procedimientos más populares para determinar clusters en un conjunto de datos es el método de *k-Medias* (k-Means). Si bien los precursores de este algoritmo fueron MacQueen en el año 1967 [25] y Hartigan en el año 1978 [16], Pollard en sus trabajos de 1981 [33] y 1982 [34] prueba la consistencia fuerte del método y su distribución asintótica respectivamente.

Se comienza primero por describir el procedimiento a través de estos trabajos y luego para calcular el centro de cada cluster se implementará una métrica robusta de la forma $\frac{d}{1+d}$, siendo d la distancia euclideana, lo cuál convertirá al algoritmo en estable frente a la presencia de outliers. En lugar de tomar como centro de cluster la media - punto que minimiza la suma cuadrática de las distancias euclídeas - se toma el punto que minimiza la suma de las distancias robustas $\frac{d}{1+d}$. Además la misma prueba de Pollard servirá para probar su consistencia.

3.1. Introducción

El procedimiento de clustering por k-medias prescribe un criterio para particionar un conjunto de puntos en k grupos: para dividir los puntos $x_1, x_2, \ldots, x_n$ en $\mathbb{R}^s$ se debe elegir los centros de los clusters, $a_1, a_2, \ldots, a_k$, de forma de minimizar

$$W_n = \frac{1}{n} \sum_{l=1}^{n} \min_{1 \leq j \leq k} \|x_l - a_j\|^2,$$

donde $\|\cdot\|$ denota la norma euclídea, para entonces asignar cada x_l a su centro de cluster más cercano. De esta forma, cada centro a_l adquiere un subconjunto C_l de puntos x como su cluster asociado. La media de los puntos en C_l debe ser igual a a_l, de otra forma, W_n podría ser disminuida mediante el remplazo de a_l por la media del cluster, en primera instancia, y entonces reasignar algunos de los x's a sus nuevos

centros. Este criterio es, entonces, equivalente al de minimizar la suma de los cuadrados entre los clusters.

Se asume que $\{x_1, x_2, \ldots, x_n\}$ es una muestra de observaciones independientes de alguna distribución P. Pollard brinda condiciones que aseguran la convergencia casi segura de los centros de los clusters cuando el tamaño de la muestra aumenta, generalizando uno de los resultados de Hartigan (1978) [16], quién lo probó para dos cluster.

MacQueen (1967) [25] obtuvo resultados de consistencia débil para el algoritmo de k-Medias que distribuye puntos secuencialmente entre k clusters. Con este algoritmo, los centros no son escogidos para minimizar W_n; en su lugar, cada x_n es asignado a el cluster con el centro más cercano, entonces ese centro es movido a la media del cluster modificado.

Debido a las dificultades que pueden surgir de las ambigüedades en el etiquetado de los puntos $x_1, x_2, \ldots, x_n$ y los centros $a_1, a_2, \ldots, a_k$, es ventajoso el considerar W_n como una función del conjunto de centros de los clusters y de la medida empírica P_n obtenida de la muestra por asignarle masa n^{-1} a cada uno de los $x_1, x_2, \ldots, x_n$.El problema es entonces, el de minimizar

$$W(A, P_n) := \int \min_{a \in A} \|x - a\|^2 P_n(dx)$$

sobre todas las posibles elecciones del conjunto A conteniendo k (o menos) puntos. Para cada A fijo, una ley fuerte de los grandes números (LFGN) muestra que

$$W(A, P_n) \to W(A, P) := \int \min_{a \in A} \|x - a\|^2 P(dx), \quad \text{c.s.}$$

Puede esperarse que A_n, el conjunto de centros de clusters óptimo para la muestras de tamaño n, debería estar cerca de $\bar{A}$, el conjunto de centros que minimizan $W(\cdot, P)$, siempre que $\bar{A}$ este determinado únicamente. Por tanto, existe un etiquetado $a_{n1}, a_{n2}, \ldots, a_{nk}$ de puntos en A_n, y un etiquetado $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_k$ de puntos en $\bar{A}$, de forma que $a_{nl} \to \bar{a}_l$ c.s. Este enfoque también evita problemas con la posible coincidencia de dos de los centros de clusters.

En la práctica, encontrar un A en el cual $W(\cdot, P_n)$ alcanza su mínimo global involucra una cantidad prohibitiva de cálculos. Sin embargo, existen algoritmos eficientes para encontrar particiones localmente óptimas de los puntos muestrales en k clusters.

El método de prueba de la convergencia de los centros está basado en la aplicación repetida de la LFGN; el argumento se aplica a casi todos los puntos muestrales ω.

La prueba se aplicará a un criterio de clustering más general. Por ejemplo, los centros de los clusters pueden ser escogidos para minimizar una cantidad basada en desviaciones absolutas,

$$\int min_{a\in A}\|x-a\|P_n(dx),$$

o aún un criterio con atractivo de robustez,

$$\int min_{a\in A}\|x-a\|\wedge 1 P_n(dx).$$

El teorema es probado en una generalidad que incluye tales posibilidades: una función monótona creciente $\phi(\|x-a\|)$ de las desviaciones $\|x-a\|$ puede ser usada en definir una suma de desviaciones entre clusters, lo cuál será utilizado en este trabajo.

3.2. El Teorema de Consistencia

Sean $x_1, x_2, \ldots, x_n$ variables aleatorias independientes en $\mathbb{R}^s$ con distribución común P. Sea P_n la correspondiente medida empírica. La muestra $\{x_1, x_2, \ldots, x_n\}$ va a ser dividida en k clusters mediante minimizar una suma de desviaciones entre clusters, y se puede probar un resultado de consistencia sobre los centros de los cluster.

Para cada medida de probabilidad Q en $\mathbb{R}^s$ y cada subconjunto (finito) A de $\mathbb{R}^s$ se define

$$\Phi(A,Q) := \int min_{a\in A}\phi(\|x-a\|)Q(dx).$$

y

$$m_k(Q) := \text{ínf}\{\Phi(A,Q) : A \text{ contiene } k \text{ o menos puntos}\}.$$

Para un k dado, el conjunto $A_n = A_n(k)$ de centros de clusters muestrales óptimos será elegido para satisfacer $\Phi(A_n, P_n) = m_k(P_n)$; los centros de clusters poblacionales $\bar{A} = \bar{A}(k)$ satisfacen $\Phi(\bar{A}, P) = m_k(P)$. El objetivo es mostrar que $A_n \to \bar{A}$, c.s.

La convergencia de conjuntos debería ser tomada como la convergencia determinada por la métrica de Hausdorff $H(\cdot,\cdot)$, la cual está definida para subconjuntos compactos A, B de $\mathbb{R}^s$ por $H(A,B) < \delta$ si y solo si todo punto de A está entre una distancia (euclídea) δ de al menos un punto de B, y viceversa. Suponer que A contiene exactamente k puntos distintos, y que δ es elegido menor a la mitad de la distancia mínima entre puntos de A. Entonces si B es cualquier conjunto de k o menos puntos para el cual $H(A,B) < \delta$, él debe contener exactamente k puntos distintos, cada uno de ellos se encuentra a una distancia *no mayor* δ de un punto únicamente determinado en A. La convergencia casi segura de A_n en el sentido de Hausdorff podría, entonces,

ser traducida a una convergencia casi segura de los centros de los clusters bajo un adecuado etiquetamiento.

Para que los procedimientos aquí descritos tengan sentido, la función ϕ debe satisfacer algunas condiciones de regularidad. Se precisa tener una ϕ continua y no decreciente, con $\phi(0) = 0$. De forma de controlar el crecimiento de ϕ en las colas, asumir que existe alguna constante λ tal que $\phi(2r) \leq \lambda\phi(r)$ para todo $r > 0$. En tanto $\int \phi(\|x\|)P(dx)$ sea finito, esto asegura que $\Phi(A, P)$ es finito para cada A: para cada $a \in \mathbb{R}^s$,

$$\int \phi(\|x - a\|)P(dx) \leq \int \phi(\|x\| + \|a\|)P(dx) \leq$$

$$\leq \phi(2\|a\|) + \int_{\|x\| \geq \|a\|} \phi(2\|x\|)P(dx) \leq \phi(2\|a\|) + \lambda \int \phi(\|x\|)P(dx).$$

Estos supuestos sobre ϕ se mantendrán en el resto de este capítulo.

Teorema 6 (Consistencia de los centros) *Suponer que $\int \phi(\|x\|)P(dx) < \infty$ y que para $j = 1, 2, \ldots, k$ existe un único conjunto $\bar{A}(j)$ para el cual $\Phi(\bar{A}(j), P) = m_j(P)$. Entonces $A_n \to \bar{A}(k)$ c.s., y $\Phi(A_n, P_n) \to m_k(P)$ c.s.*

La condición de unicidad en los $\bar{A}(j)$'s acarrea mucha información: no sólo es necesaria para el argumento inductivo, sino también implica que $m_1(P) > m_2(P) > \cdots > m_k(P)$. Suponer que $m_{j-1}(P) = m_j(P)$ para algún j. Entonces $\bar{A}(j-1)$ podría ser aumentado por cualquier punto arbitrario para dar un conjunto A (no único), de no más de j puntos distintos, para el cual $\Phi(A, P) = m_j(P)$. La condición similar implica que $\bar{A}(j)$ contiene exactamente j puntos distintos.

Ya que las conclusiones del teorema son en términos de convergencia casi segura, puede haber conjuntos nulos de ω's para los cuáles la convergencia no se cumple.

3.2.1. LFGN Uniforme y la Continuidad de $\Phi(\cdot, P)$

Teorema 7 (Ley Fuerte de los Grandes Números) *Sea $\mathcal{G}$ la familia de funciones P-integrables en $\mathbb{R}^2$ de la forma $g_A(x) := \min_{a \in A} \phi(\|x - a\|)$, donde A varía sobre todos los subconjuntos de $\mathcal{E}_k$ conteniendo k o menos puntos.*

$$\sup_{g \in \mathcal{G}} \left| \int g \, dP_n - \int g \, dP \right| \to 0 \ \ c.s. \tag{3.1}$$

Una condición suficiente para que (3.1) se cumpla es: para cada $\epsilon > 0$ existe una clase finita $\mathcal{G}_\epsilon$ de funciones tales que para cada $g \in \mathcal{G}$ existen funciones $\dot{g}$, $\bar{g} \in \mathcal{G}_\epsilon$ con $\dot{g} \leq g \leq \bar{g}$ y $\int (\bar{g} - \dot{g}) \, dP < \epsilon$. (La prueba usa la LFGN aplicada a cada función en la

clase numerable $\mathcal{G}_{1/2} \cup \mathcal{G}_{1/3} \cup \mathcal{G}_{1/4} \cup \cdots$, junto con la cota $\int (\bar{g} - g_0)\, dP + \max\{|\int \bar{g}\, dP_n - \int \bar{g}\, dP|, |\int \dot{g}\, dP_n - \int \dot{g}\, dP|\}$ para $|\int g\, dP_n - \int g\, dP|$.)

Notar que los argumentos no dependen realmente del espacio muestral subyacente, estando $\mathbb{R}^s$ equipado con su norma usual, cualquier espacio métrico para el cual todas las bolas cerradas son compactas serviría. Por ejemplo, si se toma la métrica $d'(x, y) = \frac{d(x,y)}{1+d(x,y)}$ siendo d la métrica euclideana, la convergencia se cumple.

3.3. Teorema Central del Límite para k-Medias

Se realiza la descomposición de $W_n(a) = n^{-1} \sum_{i=1}^{n} \min_{1 \le j \le k} \|x_i - a_j\|^2$ en dos componentes que pueden ser expresadas en función de la medida empírica P_n y del proceso empírico asociado $X_n(.) = n^{1/2}(P_n(.) - P(.))$. Para cualquier vector $a = [a_1, a_2, \ldots, a_k] \in \mathbb{R}^{kd}$ y para cualquier $x \in \mathbb{R}^d$ se define

$$\phi(x, a) = \min_{1 \le j \le k} \|x - a_j\|^2,$$

entonces

$$W_n(a) = P_n \phi(\cdot, a) = P\phi(\cdot, a) + n^{-1/2} X_n \phi(\cdot, a).$$

La componente $P\phi(\cdot, a)$ conocida popularmente como suma de cuadrados entre cluster, se denotará por $W(a)$.

Pollard, en su trabajo de 1982, demuestra propiedades asintóticas de normalidad para la sucesión de centros.

Teorema 8 (Teorema Central para k-Medias) *Sea b_n el vector de centros óptimos del algoritmo k-means para muestras independientes con distribución P en $\mathbb{R}^d$. Suponer que*

1. *El vector μ que minimiza suma de las distancias al cuadrado $W(\cdot)$ entre clusters es única, hasta volver a etiquetar sus coordenadas.*

2. *$P\|x\|^2 < \infty$.*

3. *La medida de probabilidad P tiene una función continua de densidad f con respecto a la medida de Lebesgue λ en $\mathbb{R}^d$.*

4. *Existe una función dominante $g(\cdot)$ con $f(x) \le g(\|x\|), \forall x \in \mathbb{R}^d$ y $r^d g(r)$ integrable con respecto a la medida de Lebesgue en $[0, \infty)$.*

5. *La matriz Γ evaluada en $a = \mu$ es definida positiva.*

Entonces $n^{1/2}(b_n - \mu) \xrightarrow{d} N(0, \Gamma^{-1}V\Gamma^{-1})$*, donde* V *es una matriz conformada por*

$$V_i = 4P[M_i(x - \mu_i)(x - \mu_i)'],$$

donde V_i *es el* i*-ésmino bloque de la diagonal y* M_i *denota el conjunto de puntos en* $\mathbb{R}^n$ *más cerca de* μ_i *que respecto a otro* μ_j*.*

3.4. Una variante robusta de K-Medias

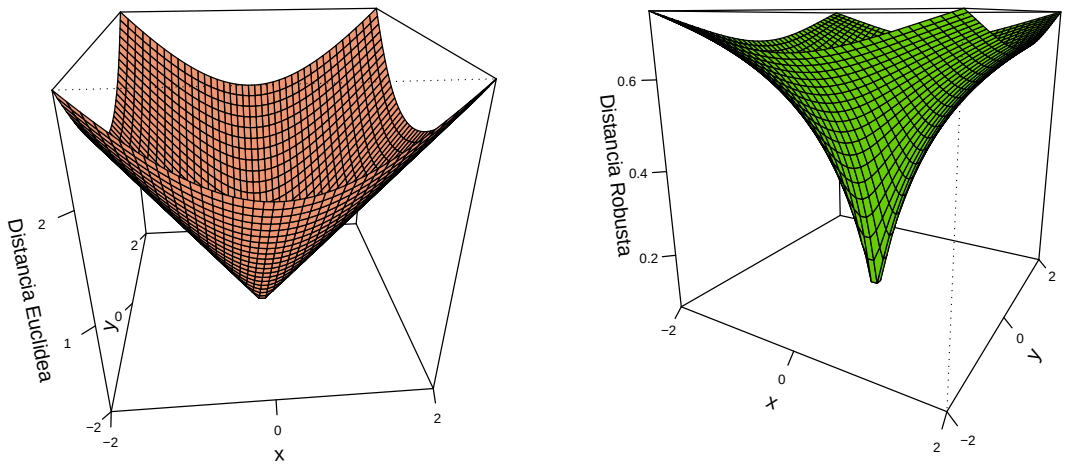

Figura 3.1: Funciones de distancias Euclídea (izquierda) y Robusta (derecha) al orígen

Es sencillo probar que $\phi = \frac{\psi}{1+\psi}$ es una función monótona creciente si $\psi \geq 0$ y una distancia si ψ lo es. Tomando a ψ como la distancia euclídea, la distancia ϕ verifica las hipótesis necesarias para los trabajo de Pollard [81] y [82]. La elección de esta distancia **acotada** se debe a su sencillez y que crece a tasa decreciente con la distancia euclídea: observaciones muy alejadas de la restantes verán reducido su impacto en el cálculo de los centros de los clusters, mientras que mantiene el orden inducido por la distancia euclídea con respecto al centro.

Para ejemplificar el efecto de la contaminación en el algoritmo k-Medias original, se implementó el mismo independiente de la distancia a utilizar en la determinación del centro de cada cluster. Se efectuó la misma con las distancias Euclídea y Robusta ψ, sobre simulaciones con distintos tipos de contaminación: local y global. Debido a que no se tiene una expresión analítica para obtener el centro óptimo de cada cluster bajo una distancia cualquiera, se optó por realizar la optimización mediante métodos numéricos.

3.5. Estudio de Simulación

En esta sección se estudiará la performance de las distintas variantes del algoritmo k-Medias y su estabilidad frente a la presencia de contaminación. Para ello se simularán distintos escenarios, en los cuales se considera ruido local entre los cluster y fuera de ellos, así como también ruido global.

En todos los casos se consideran 2 grupos conformados con 100 datos cada uno y 20 observaciones atípicas. Se comparará en estos escenarios tres técnicas de clustering ya descritas:

- k-Medias con 3 grupos, donde el grupo más pequeño determina los outliers.
- k-Medias con 2 grupos, donde luego de conformar los grupos se determinan cuales son los outliers tomando como criterio la distancia de Mahalanobis a el centro al que fue asociado, podando el 10 % de las observaciones.
- La variante robusta de k-Medias, luego de determinar los grupos se procede a podar igual que en k-Medias con 2 grupos.

A efectos de analizar la performance se realizan 150 repeticiones de cada técnica, y en cada una de estas se calcula el porcentaje de observaciones bien clasificadas. Se entiende que una observación es bien clasificado si es asociada al grupo del que fue simulado (Grupo A, Grupo B o Grupo R / Outlier).

Para estos 150 porcentajes se realiza los diagramas de caja correspondiente a cada técnica.

Se comienza por analizar el problema en los distintos tipos de escenarios y se derivan conclusiones de los mismos.

3.5.1. Primer Escenario: Ruido Global

A los efectos de analizar este primer escenario se simulan 220 datos en $\mathbb{R}^2$.

El primer grupo de 100 observaciones proviene de una distribución Normal bivariante con vector de medias $(4, 0)$ y matriz de varianzas y covarianzas identidad, mientras que el otro grupo también está conformado por 100 datos de una distribución Normal bivariante con vector de medias $(0, 4)$ y matriz de varianzas y covarianzas identidad. El papel de ruido global lo jugarán 20 datos que se simulan uniformemente en el cuadrado $[-10, 10]^2$.

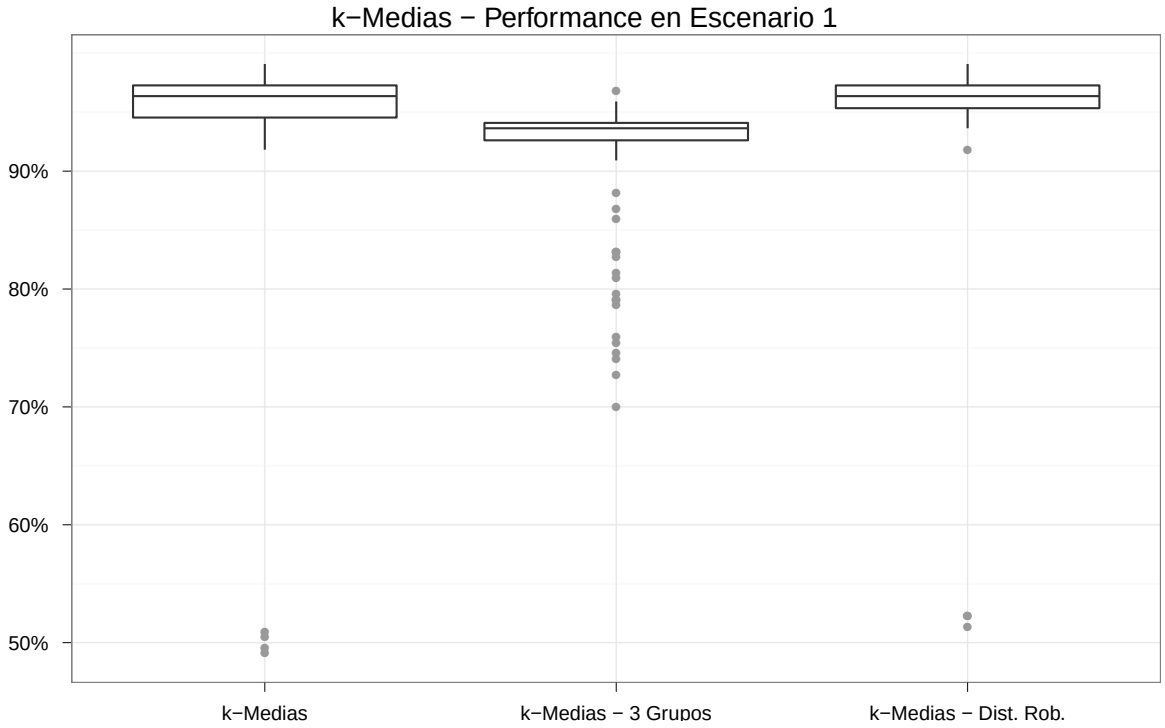

Figura 3.2: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo

Como se puede apreciar en 3.2, los tres algoritmos tienen una alta eficiencia. Este ruido, al ser global y uniforme no produce sesgos considerables en la estimación de los centros de los cluster.

De todas formas, la variante robusta produce en general mejores resultados con una variabilidad similar.

Como en este escenario el grupo de los outliers no se presenta en un grupo claramente definido k-Medias con 3 grupos es claramente deficiente.

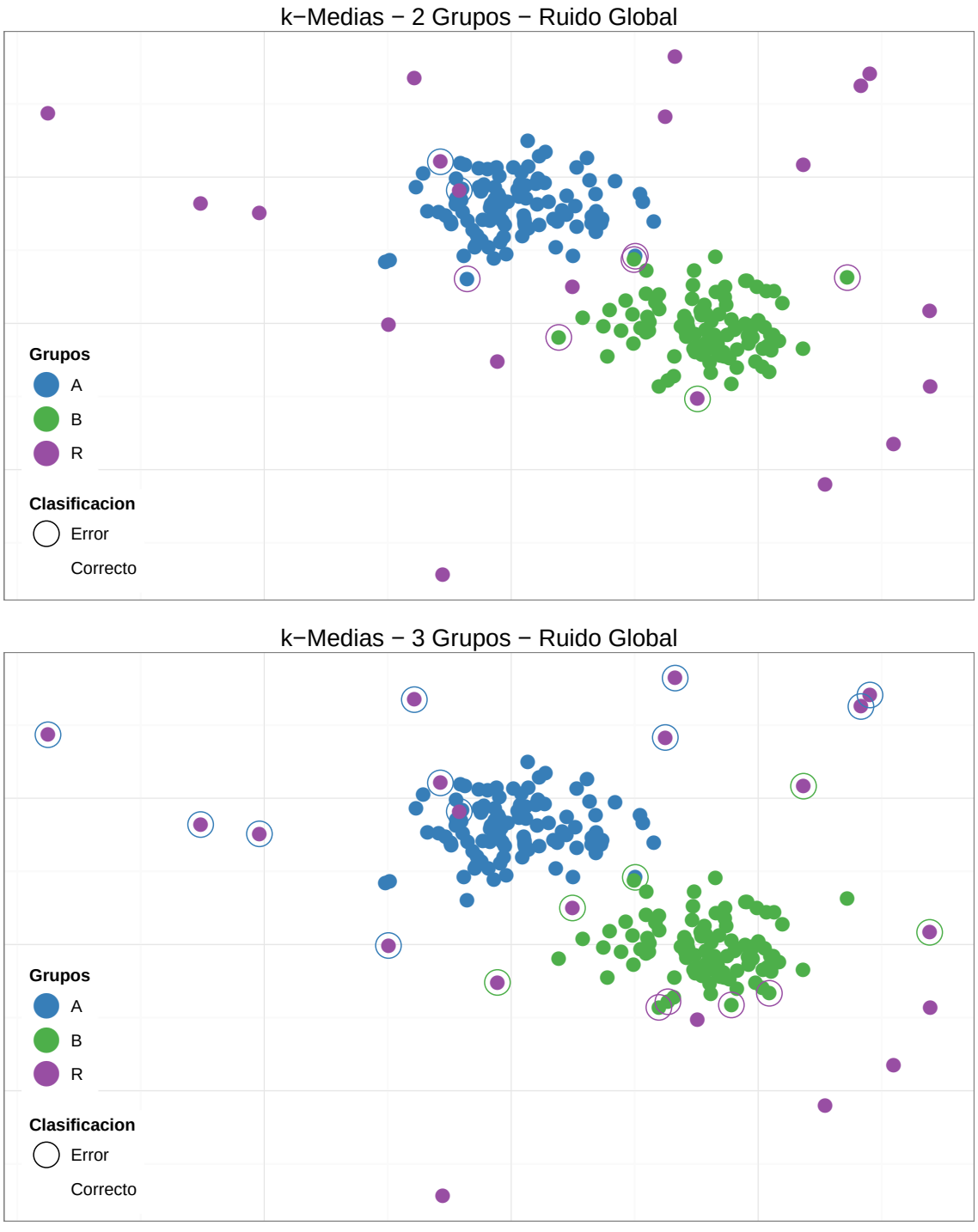

Figura 3.3: Clasificación de variantes de k-Medias en 2 grupos con ruido global

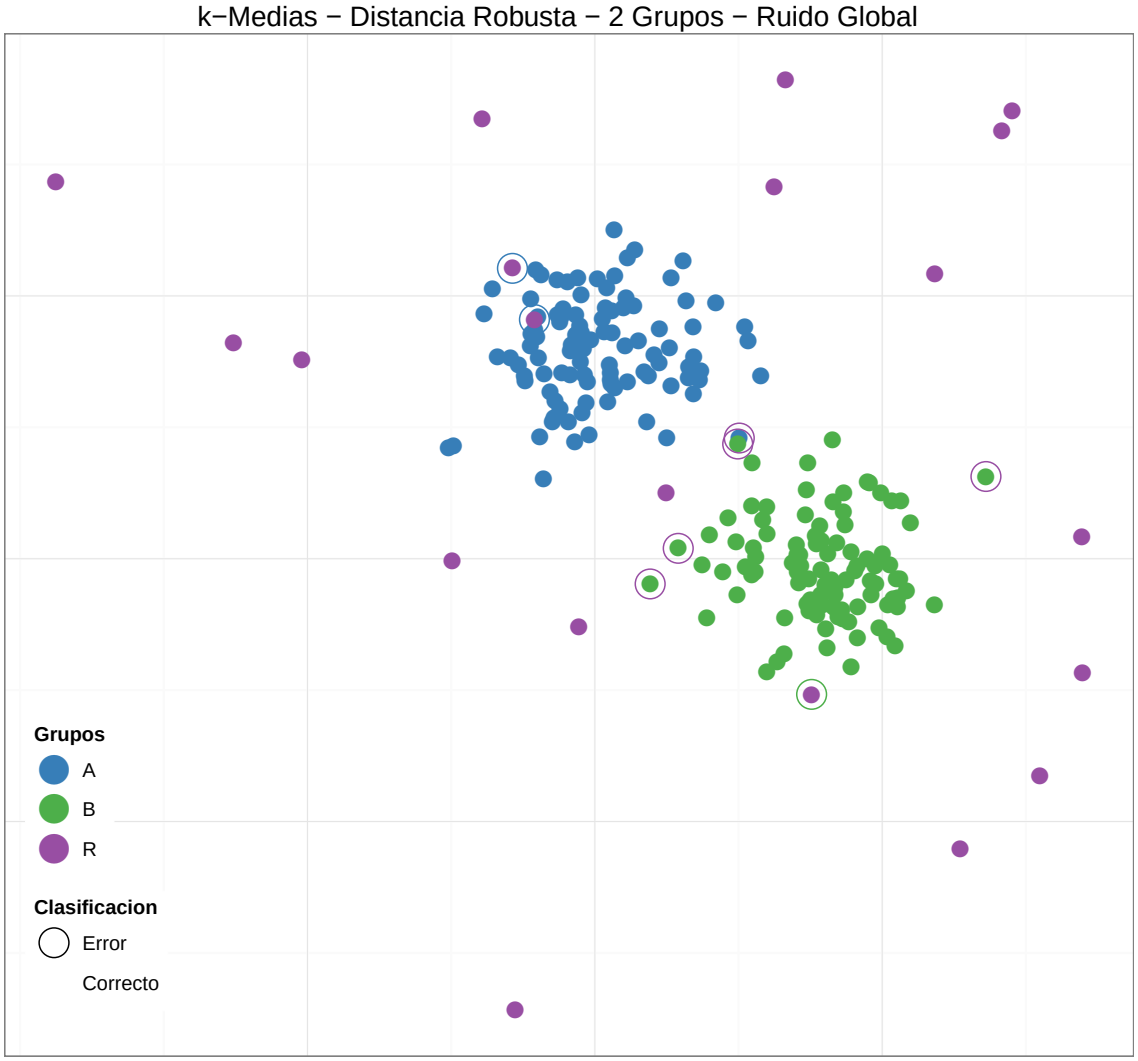

Figura 3.4: Clasificación de variantes de k-Medias en 2 grupos con ruido global - 2

3.5.2. Segundo Escenario: Ruido Local Alejado

Para este segundo escenario se mantienen las simulaciones respecto a los grupos, 220 datos en $\mathbb{R}^2$, 100 de estos provenientes de una distribución Normal bivariante con vector de medias $(4, 0)$ y matriz de varianzas y covarianzas identidad, 100 datos de una distribución Normal bivariante con vector de medias $(0, 4)$ y matriz de varianzas y covarianzas identidad. Pero los 20 datos que jugarán al papel de ruido local se simulan uniformemente en el cuadrado $[-5, 0]^2$, ubicándose en el cuadrante inferior izquierdo del escenario.

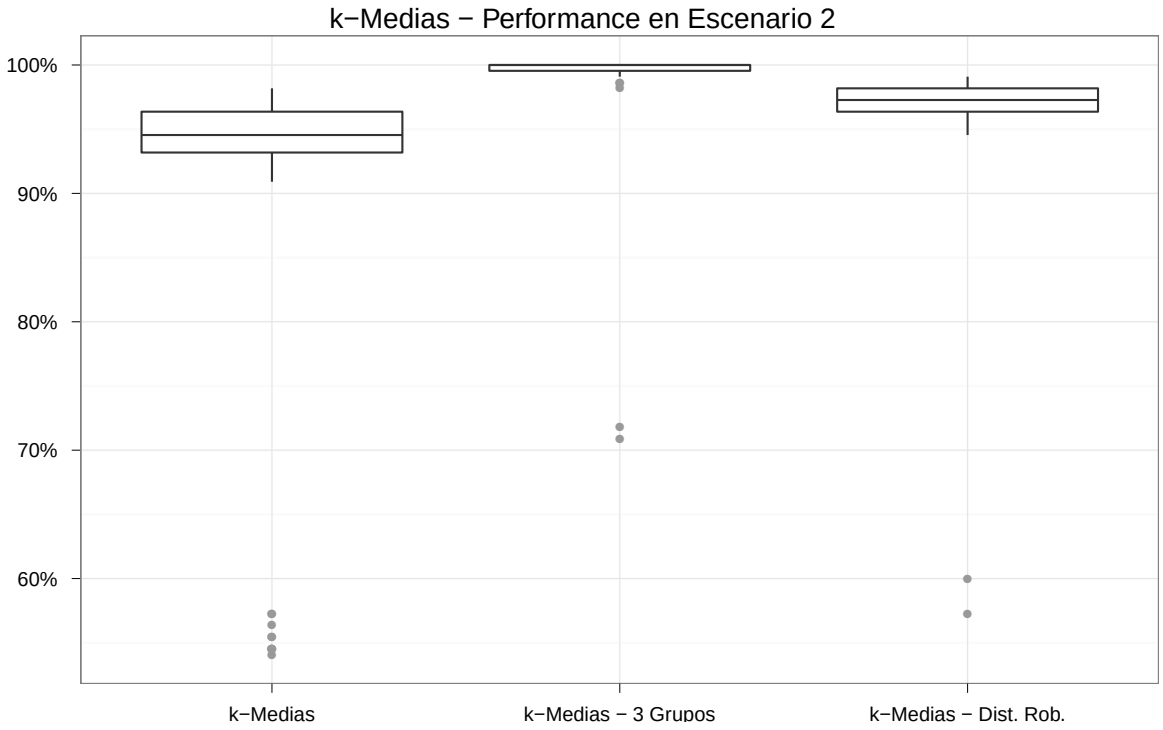

Figura 3.5: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo

Se puede observar que - como era de esperar en este caso - al estar el grupo de outliers claramente diferenciado de los grupos, k-Medias con 3 grupos es el algoritmo que mejor clasifica.

Sin embargo, la variante robusta de k-Medias se mantiene estable frente a estas tipologías, manteniendo en el 50 % central de las repeticiones un porcentaje de aciertos entre el 96 % y el 98 % (ver 3.5).

La incidencia de este grupo local de outliers es alta sobre los centros de los cluster del algoritmo de k-Medias con 2 grupos, problema que se logra amortiguar con la variante robusta.

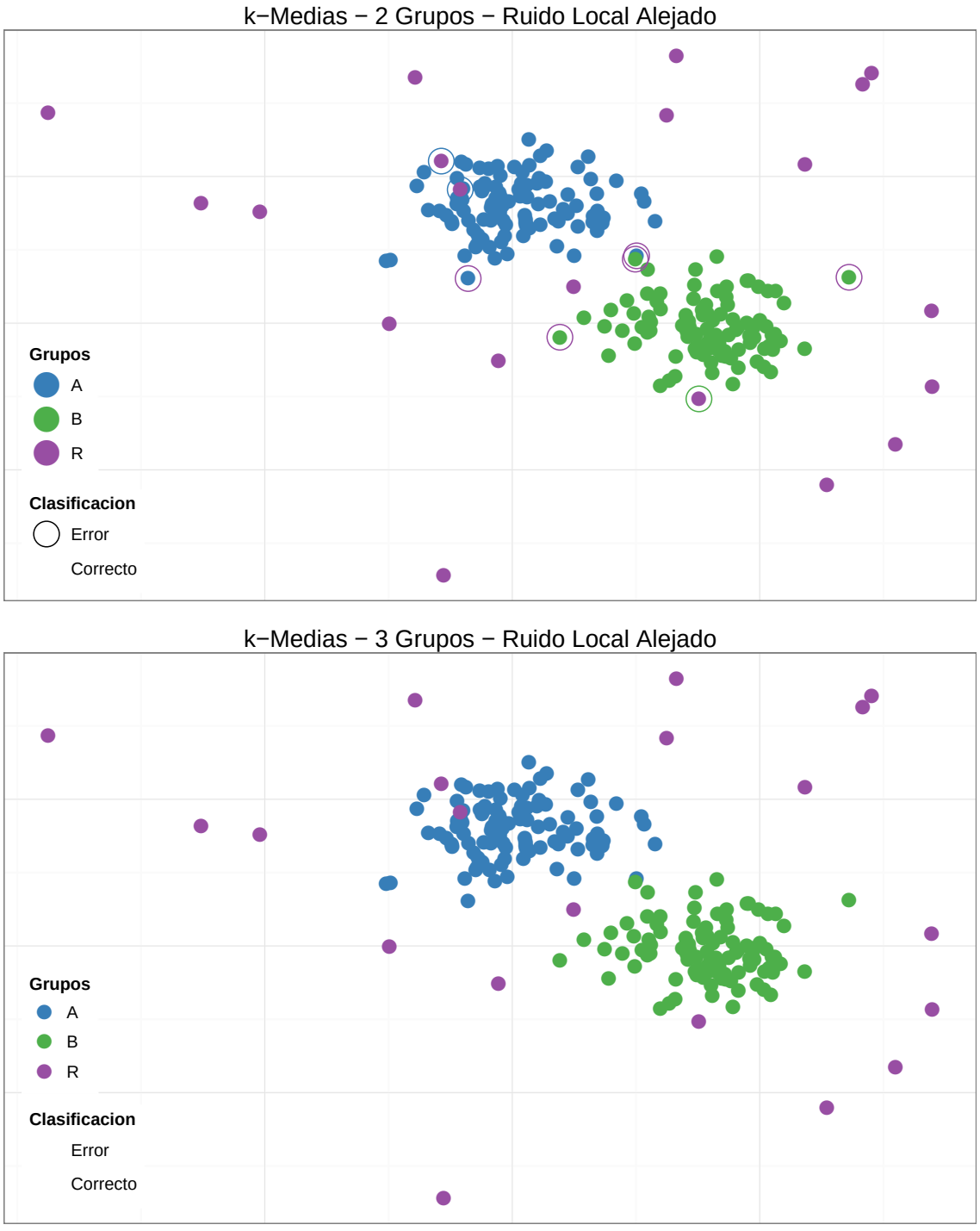

Figura 3.6: Clasificación de Algoritmo de k-Medias variante robusta en 2 grupos con ruido local alejado

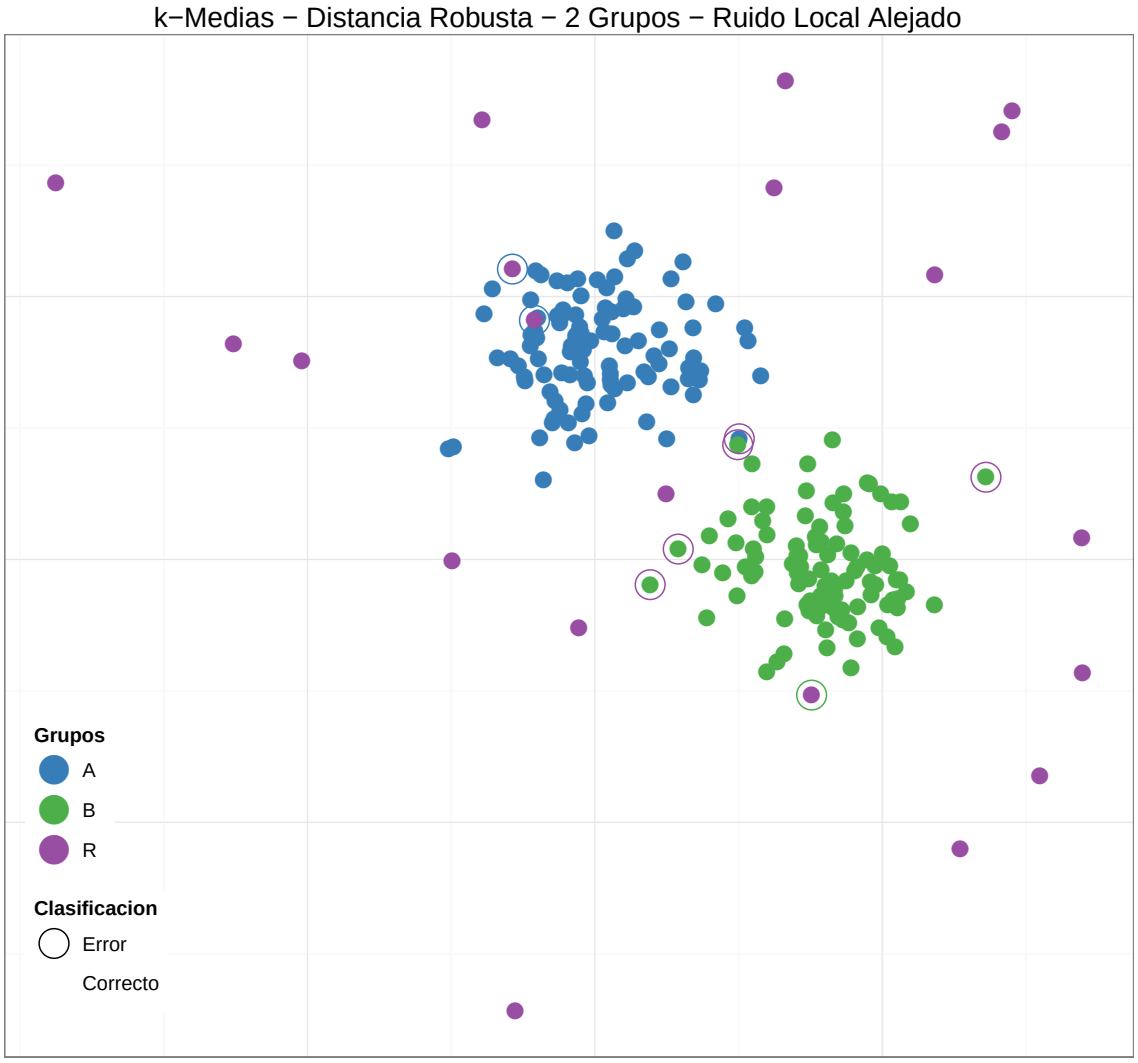

Figura 3.7: Clasificación de Algoritmo de k-Medias variante robusta en 2 grupos con ruido local alejado - 2

3.5.3. Tercer Escenario: Ruido Local entre Grupos

En este tercer y último escenario a analizar se mantienen la cantidad de simulaciones respecto a los grupos, 220 datos en $\mathbb{R}^2$, 100 de estos provenientes de una distribución Normal bivariante con vector de medias $(4, 0)$ y matriz de varianzas y covarianzas identidad, 100 datos de una distribución Normal bivariante con vector de medias $(0, 4)$ y matriz de varianzas y covarianzas identidad. Pero el ruido local se simula uniformemente en el cuadrado $[0{,}5, 3{,}5]^2$, ubicándose entre los centros de ambas normales.

Se genera así un escenario donde no es nada trivial delimitar los grupos, así como tampoco es sencilla la identificaciones de los outliers.

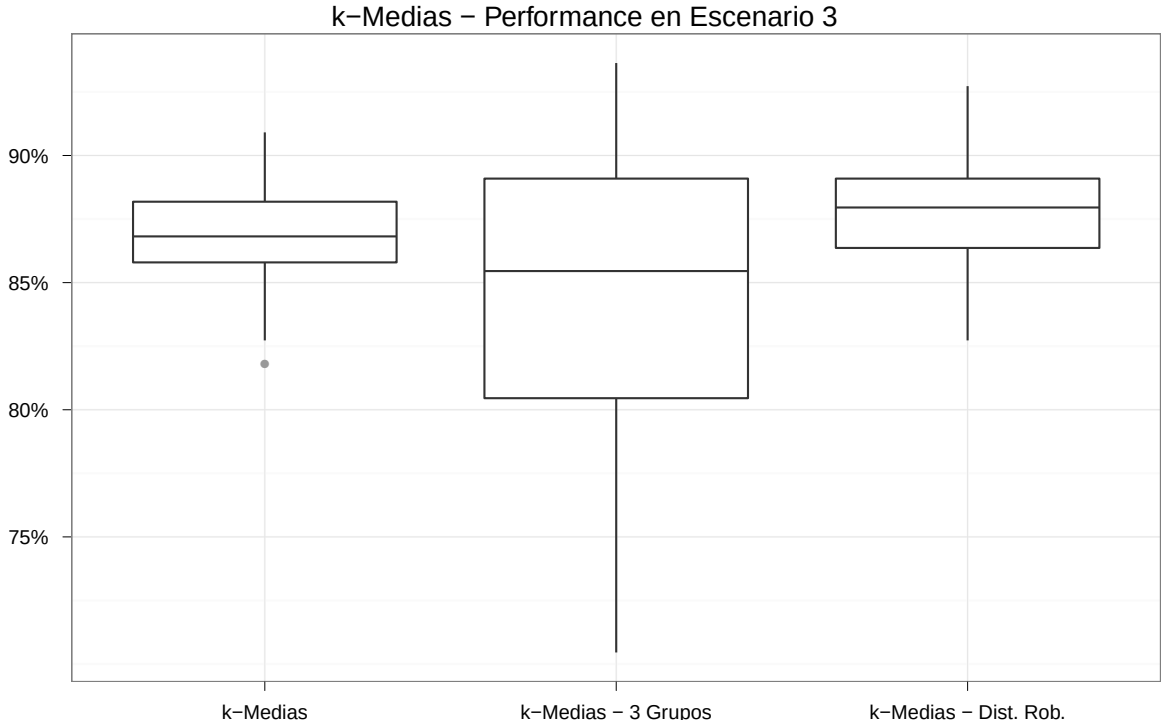

Figura 3.8: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo

Se esperaba que el algoritmo de k-Medias con tres grupo siguiera brindando la mejor clasificación. Sin embargo, esta tipología de outliers atrae los centros hacia la parte central del escenario, distorsionando de esta forma el algoritmo mencionado.

Observando la Figura 3.8 se puede ver como el porcentaje de aciertos disminuye considerablemente en los tres métodos, y la mayor eficiencia la presenta la variante robusta del k-Medias.

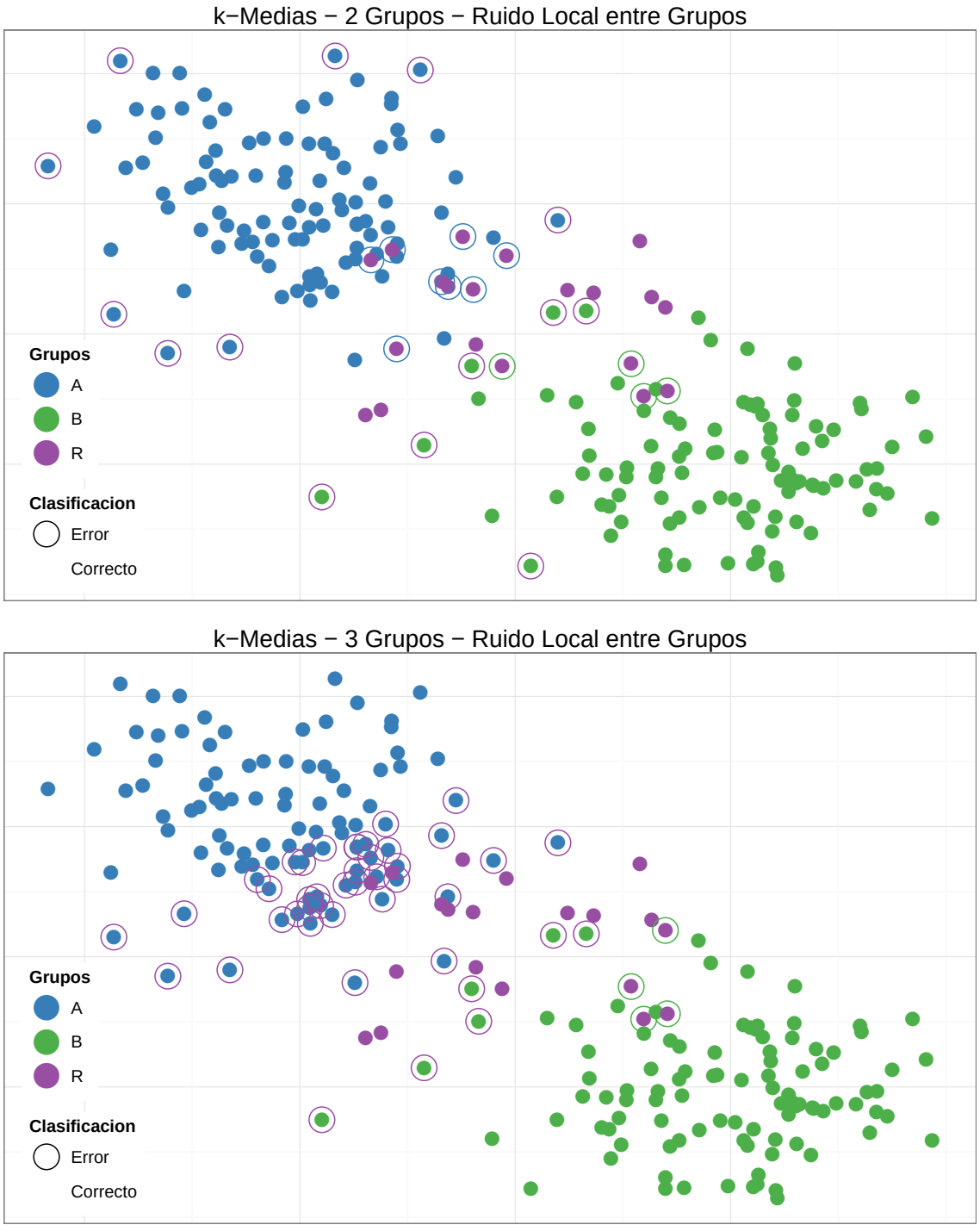

Figura 3.9: Clasificación de variantes de k-Medias en 2 grupos con Ruido Local entre Grupos

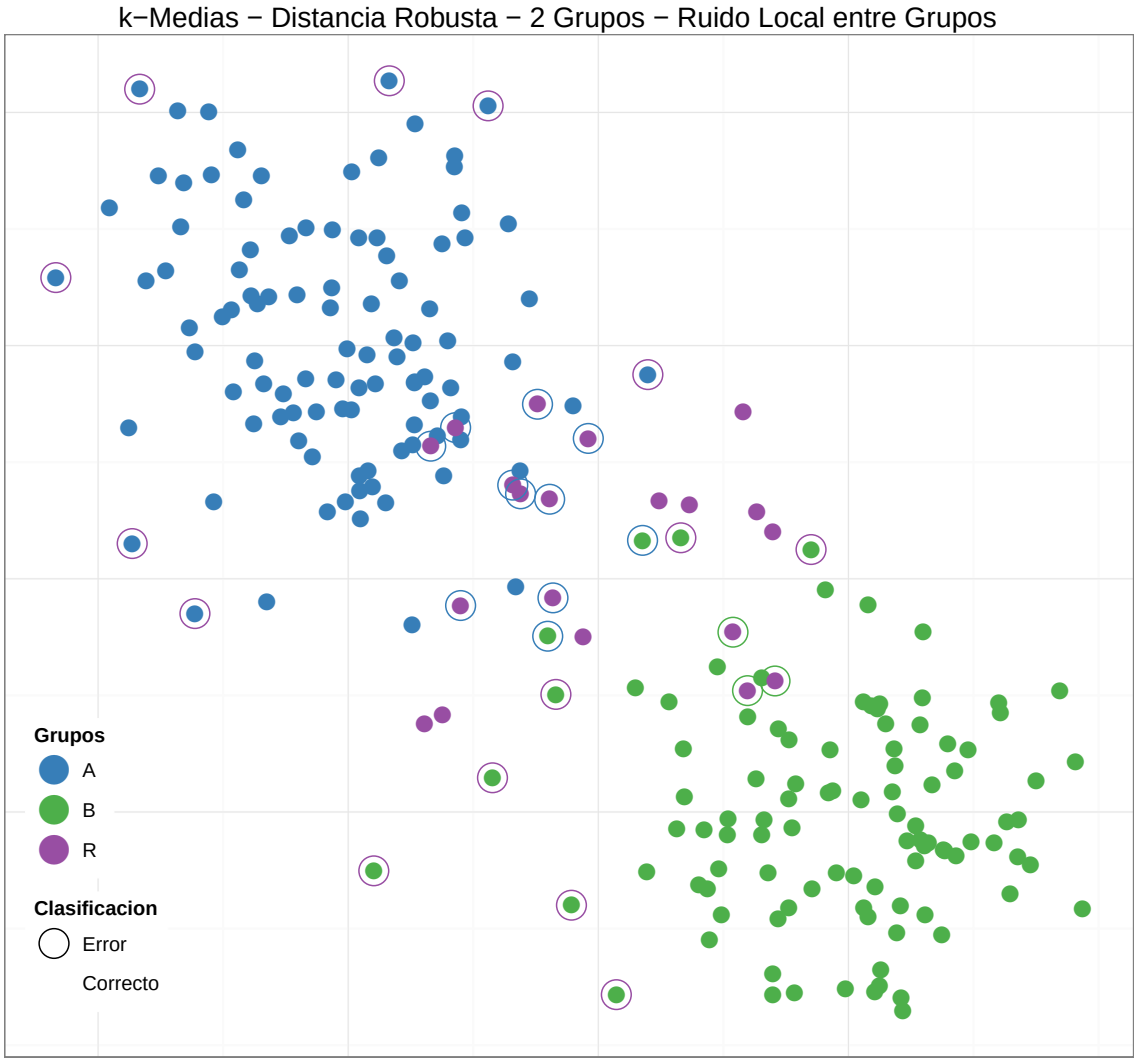

Figura 3.10: Clasificación de variantes de k-Medias en 2 grupos con Ruido Local entre Grupos - 2

3.5.4. Conclusiones

Los 3 métodos presentan ventajas y desventajas según el escenario de observaciones.

No obstante, la variante robusta es la que presenta mayor estabilidad frente a distintas tipologías de contaminación, sin ver su eficiencia comprometida en cuanto a lo que a clasificación se refiere.

Por tanto, como en general no se sabe de qué forma o múltiple formas se van a presentar los datos anómalos en la práctica, es aconsejable clasificar mediante un algoritmo que no sea altamente distorsionado por las diferentes variedades de grupos de outliers.

Si se cuenta con información acerca de que el ruido es local, es útil modelar estos como un nuevo grupo de menor tamaño y usar k-Medias sin tener que robustecer la métrica.

Capítulo 4

Mezcla de Distribuciones t

La mezcla de distribuciones para el modelado de datos tiene sus años en la historia de la estadística, ya Pearson (1894) [29] utilizaba mezclas de gaussianas univariadas para la modelación de datos. Wolfe [46] y Day [2] en 1969 comenzaron el estudio de las estimaciones de los parámetros de la mezcla de forma eficiente. Actualmente el modelado mediante una mezcla finita de distribuciones tiene aplicaciones en varias ramas de la estadística. Un trabajo actual de Melnykov (2010) [26] es un buen compendio de estos procedimientos.

Si bien la mezcla de normales tiene un extenso uso en modelaciones estadísticas, en particular para procedimientos de cluster, los parámetros de la mezcla de normales son muy sensibles a outliers.

Una alternativa para enfrentar este problema es dotar a las distribuciones de la mezcla de colas más pesadas, como son las distribuciones t, que soporten a estos outliers sin distorsionar en forma severa la estimación de los parámetros. Está propuesta es realizada por McLachlan y Peel en el 2000 [31].

4.1. Introducción

Una manera de modelar los potenciales outliers es a través de una mezcla de dos densidades normales:

$$(1-\epsilon)\phi(y_j;\mu,\Sigma)+\epsilon\phi(y_j;\mu,k\Sigma).$$

Este modelo de mezclas lo podemos escribir de la siguiente manera:

$$\int \phi(y_j;\mu,\Sigma/u)dH(u).$$

Siendo H la distribución de una probabilidad con masa $(1-\epsilon)$ en el punto $u=1$ y con masa ϵ en el punto $u=\frac{1}{k}$. Si se sustituye la distribución de H por una χ^2 con ν obtenemos una distribución de Student con parámetro de posición μ, con una matriz definida positiva Σ y ν grados de libertad,

$$f(y_j;\mu,\Sigma,\nu)=\frac{\Gamma(\frac{\nu+p}{2})|\Sigma|^{-1/2}}{(\pi\nu)^{\frac{p}{2}}\Gamma(\frac{\nu}{2})\{1+\delta(y_j;\mu,\Sigma)/\nu\}^{\frac{\nu+p}{2}}}$$

donde

$$\delta(y_j;\mu,\Sigma)=(y_j-\mu)^T\Sigma^{-1}(y_j-\mu),$$

denota el cuadrado de la distancia de Mahalanobis entre y_j y μ. Si $\nu>1$, μ es la media de Y_j, y si $\nu>2$ entonces $\nu(\nu-2)^{-1}\Sigma$ es la matriz de covarianzas.

La sección siguiente, asumiendo la presencia de outliers, se modela mediante la mezcla de un número g de distribuciones t y se estiman los parámetros a través del algoritmo EM.

4.2. Estimación Máximo Verosímil de una Mezcla de Distribuciones t

4.2.1. Aplicación del Algoritmo EM

Se considera la estimación máximo verosímil para una mezcla de g-componentes de distribuciones t dada por

$$f(y_j;\Psi)=\sum_{i=1}^{g}\pi_i f(y_j;\mu_i,\Sigma_i,\nu_i),$$

siendo

$$\Psi=(\pi_1,\ldots,\pi_{g-1},\xi^T,\nu^T)^T,$$

donde $\nu=(\nu_1,\ldots,\nu_g)^T$ y $\boldsymbol{\xi}$ es el vector que contiene las g medias y los elementos de las g matrices de covarianzas.

El vector de datos completos está dado por

$$y_c=(y^T,Z_1^T,\ldots,Z_n^T,u_1,\ldots,u_n)^T,$$

donde $y=(y_1^T,\ldots,y_n^T)^T$ denota el vector de datos observados, $(z_1,\ldots,z_n)$ son los vectores que etiquetan el origen de $(y_1,\ldots,y_n)$ respectivamente, $z_{ij}=(z_j)_i$ es uno o cero, acorde si y_j pertenece o no a la i-ésima componente. En la caracterización a

partir de las t distribuciones es también conveniente introducir en el vector de datos completos otros datos faltantes $(u_1, \ldots, u_n)$ definidos para $z_{ij} = 1$ dado,

$$Y_j|u_j, z_{ij} = 1 \sim N(\mu_i, \Sigma_i/u_j),$$
$$U_j|z_{ij} = 1 \sim Gamma(\frac{1}{2}\nu_i, \frac{1}{2}\nu_i).$$

Dados $(z_1, \ldots, z_n)$ las v.a $(U_1, \ldots, U_n)$ son independientes. La verosimilitud de datos completos $L_c(\Psi)$ puede ser factorizada como el producto de las densidades de Z_j, por las densidades condicionales de U_j dadas las z_j y por las condicionales de Y_j dadas u_j y las z_j. Por tanto se puede escribir

$$\log L_c(\Psi) = \log L_{1c}(\pi) + \log L_{2c}(\nu) + \log L_{3c}(\xi),$$

donde

$$\log L_{1c}(\pi) = \sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij} \log \pi_i,$$

$$\log L_{2c}(\nu) = \sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}\{-\log\Gamma(\frac{1}{2}\nu_i) + \frac{1}{2}\nu_i \log(\frac{1}{2}\nu_i) + \frac{1}{2}\nu_i(\log(u_j) - u_j) - \log(u_j)\},$$

$$\log L_{3c}(\xi) = \sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}\{-\frac{1}{2}p\log 2\pi - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}u_j(y_j - \mu_i)^T\Sigma_i^{-1}(y_j - \mu_i)\},$$

siendo $\pi = (\pi_1, \ldots, \pi_g)^T$ y $\xi = (\theta_1^T, \ldots, \theta_g^T)^T$ donde θ_i contiene a μ_i y a todos los elementos de Σ_i

4.2.2. Paso E

El paso E en la $(k+1)$-ésima iteración del algoritmo EM requiere el calculo de $Q(\Psi; \Psi^(k))$, la actual esperanza condicional del logaritmo de la función de verosimilitud completa $\log L_c(\Psi)$. Para esto se necesita computar

$$E_{\Psi^{(k)}}(Z_{ij}|y_j), \quad E_{\Psi^{(k)}}(U_j|y_j, z_j) \quad y \quad E_{\Psi^{(k)}}(\log U_j|y_j, z_j).$$

Los cálculos son los siguientes,

$$E_{\Psi^{(k)}}(Z_{ij}|y_j) = \tau_{ij}^{(k)},$$

donde

$$\tau_{ij}^{(k)} = \frac{\pi_i^{(k)}.f(y_j;\mu_i^{(k)},\Sigma_i^{(k)},\nu_i^{(k)})}{f(y_j;\Psi^{(k)})},$$

es la probabilidad a posteriori de que y_j pertenezca a la i-ésima componente de la mezcla ajustando $\mathbf{\Psi}$ por $\Psi^{(k)}$. Como la distribución Gamma es la distribución conjugada a priori para U_j, se puede probar que la distribución condicional de U_j dado $Y_j = y_j$ y $Z_{ij} = 1$ es

$$U_j|y_j, zij = 1 \sim Gamma(m_{1i}, m_{2i}),$$

donde

$$m_{1i} = \frac{1}{2}(\nu_i + p),$$

y

$$m_{2i} = \frac{1}{2}\{\nu_i + \delta(y_j, \mu_i; \Sigma_i)\}.$$

Por tanto

$$E(U_j|y_j, z_{ij} = 1) = \frac{\nu_i + p}{\nu_i + \delta(y_j, \mu_i; \Sigma_i)},$$

y de esta ecuación se deduce que

$$E_{\Psi^{(k)}}(U_j|y_j, z_{ij} = 1) = u_{ij}^{(k)},$$

siendo

$$u_{ij}^{(k)} = \frac{\nu_i^{(k)} + p}{\nu_i^{(k)} + \delta(y_j, \mu_i^k; \Sigma_i^{(k)})}.$$

Ahora se aplica el conocido resultado que si R tiene una distribución $Gamma(\alpha, \beta)$ entonces

$$E(logR) = \psi(\alpha) - \log(\beta),$$

donde

$$\psi(s) = \frac{\partial\Gamma(s)/\partial s}{\Gamma(s)},$$

es la función Digamma. Si se combina este resultado con lo ante expuesto,

$$E_{\Psi^{(k)}}(logU_j|y_j, z_{ij} = 1) = \psi\left(\frac{\nu_i^{(k)} + p}{2}\right) - \log[\frac{1}{2}\{\nu_i^{(k)} + \delta(y_j, \mu_i^{(k)}; \Sigma_i^{(k)})\}] =$$

$$= \log u_{ij}^{(k)} + \left\{ \psi\left(\frac{\nu_i^{(k)} + p}{2} \right) + log\left(\frac{\nu_i^{(k)} + p}{2} \right) \right\},$$

el término derecho de está igualdad

$$\psi\left(\frac{\nu_i^{(k)} + p}{2} \right) + \log\left(\frac{\nu_i^{(k)} + p}{2} \right)$$

puede ser interpretado como el factor de corrección inputado al valor de la media condicional $u_{ij}^{(k)}$ para u_j en el $log\, u_j$.

A partir de los resultados obtenidos se deduce una expresión para la esperanza condicional de la log-verosimilitud de los datos completos

$$Q(\Psi; \Psi^{(k)}) = Q_1(\pi; \Psi^{(k)}) + Q_2(\nu; \Psi^{(k)}) + Q_3(\xi; \Psi^{(k)}),$$

donde

$$Q_1(\pi; \Psi^{(k)}) = \sum_{i=1}^{g} \sum_{j=1}^{n} \hat{\tau}_{ij}^{(k)} \log \pi_i,$$

$$Q_2(\nu; \Psi^{(k)}) = \sum_{i=1}^{g} \sum_{j=1}^{n} \hat{\tau}_{ij}^{(k)} Q_{2j}(\nu_i; \Psi^{(k)}),$$

$$Q_3(\xi; \Psi^{(k)}) = \sum_{i=1}^{g} \sum_{j=1}^{n} \hat{\tau}_{ij}^{(k)} Q_{3j}(\theta_i; \Psi^{(k)}),$$

donde si se omite los términos donde no participa ν_i,

$$Q_{2j}(\nu_i; \Psi^{(k)}) = -\log\Gamma(\frac{1}{2}\nu_i) + \frac{1}{2}\nu_i log(\frac{1}{2}\nu_i) +$$

$$+\frac{1}{2}\nu_i \left\{ \sum_{j=1}^{n} (log u_{ij}^{(k)} - u_{ij}^{(k)}) + \psi\left(\frac{\nu_i^{(k)} + p}{2} \right) - log\left(\frac{\nu_i^{(k)} + p}{2} \right) \right\},$$

entonces

$$Q_{3j}(\theta_i; \Psi^{(k)}) = -\frac{1}{2} p \log(2\pi) - \frac{1}{2} \log|\Sigma_i| + \frac{1}{2} p \log u_{ij}^{(k)} - \frac{1}{2} u_{ij} (y_j - u_i)^T \Sigma_i^{-1} (y_j - u_i).$$

Finalizado el primer paso del algoritmo EM se pasa a maximización de la verosimilitud.

4.2.3. Paso M

El paso M en la $(k+1)$-ésima iteración de el algoritmo EM, $\pi^{(k+1)}$, $\xi^{(k+1)}$ y $\nu^{(k+1)}$ son computados independientemente uno de otros.

La solución para $\pi^{(k+1)}$ y $\theta^{(k+1)}$ existen en forma cerrada.

Sólo la actualización $\nu_i^{(k+1)}$ para los grados de libertad ν_i debe ser necesariamente computada iterativamente. π_i^{k+1} esta dada por el promedio de las probabilidades a posteriori de las componentes miembros de la mezcla.

$$\pi_i^{(k+1)} = \sum_{j=1}^{n} \tau_i j^{(k)}/n.$$

Para actualizar las estimaciones de μ_i y Σ_i se necesita considerar

$$Q_3(\theta_i; \Psi^{(k)}),$$

$$\mu_i^{(k+1)} = \sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)} y_j / \sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)},$$

$$\Sigma_1^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)} (y_j - \mu_i^{(k+1)})(y_j - \mu_i^{(k+1)})^T}{\sum_{j=1}^{n} \tau_{ij}^{(k)}}.$$

4.3. Estudio de Simulación

Se evaluará la performance de la clasificación de mezclas de Normales y mezclas de t de Student con distintos grados de libertad (4 y 12 grados de libertad respectivamente) mediante la simulación de un escenario 150 veces.

En todos los casos se determina el grupo de outliers podando el 15 % de las observaciones con menor verosimilitud en la distribución de la mezcla.

Para cada escenario se simulan 350 observaciones. Estas provienen de:

- 100 observaciones de una $N\left[\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 & 0,5 \\ 0,5 & 0,5 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & -0,5 \\ -0,5 & 0,5 \end{pmatrix}\right]$
- 50 observaciones uniformes en el cuadrado $[-10, 10]^2$

Las observaciones provenientes de las distribuciones normales determinan 3 grupos de 100 observaciones cada uno, mientras que las 50 observaciones uniformes conforman el ruido global.

Para poder evaluar de manera correcta qué algoritmo clasifica de forma más efectiva, al igual que en el capítulo anterior, en cada simulación del escenario se computa el porcentaje de datos bien clasificados por cada técnica. Se realizan los diagramas de caja para estos porcentajes en cada caso (ver Figura 4.1).

Analizando estos diagramas se puede observar como las colas pesadas de la distribución de Student con 4 grados de libertad soporta de forma mas estable a los outliers. Al aumentar los grados de libertad en la distribución de Student, ésta pierde peso en sus colas y se asemeja a una normal, bajando el porcentaje de datos bien clasificados de forma brusca.

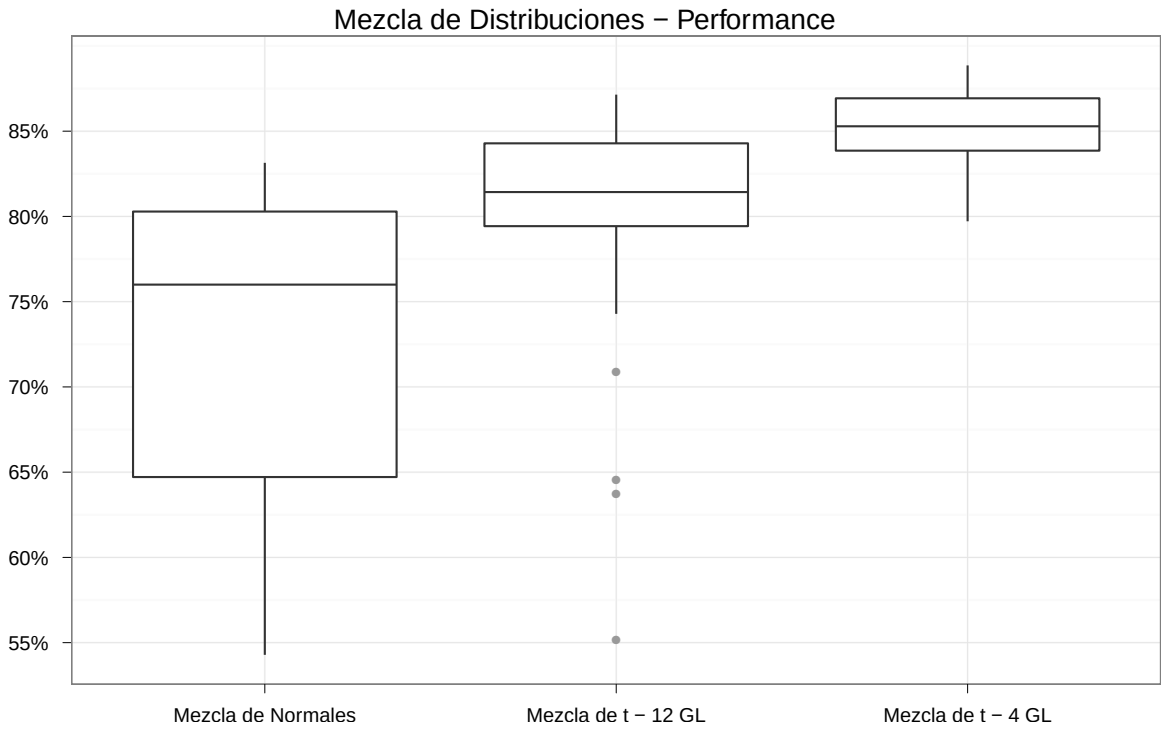

Figura 4.1: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo

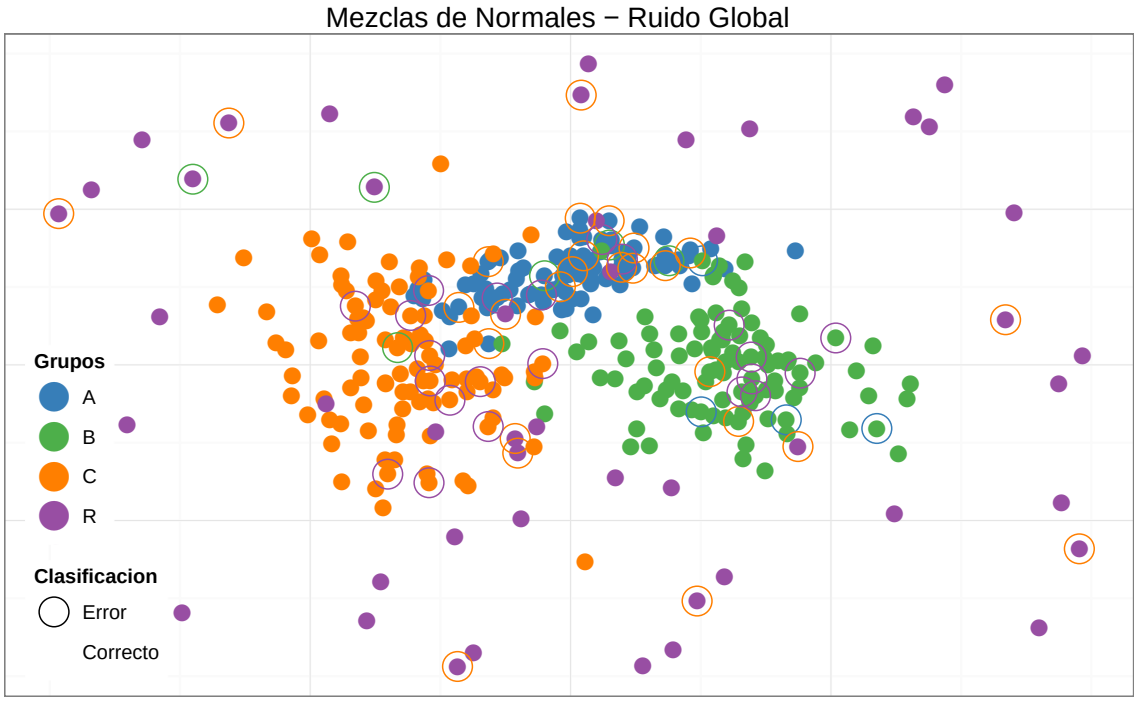

Figura 4.2: Clasificación mediante Mezcla de Normales - Ruido Glogal

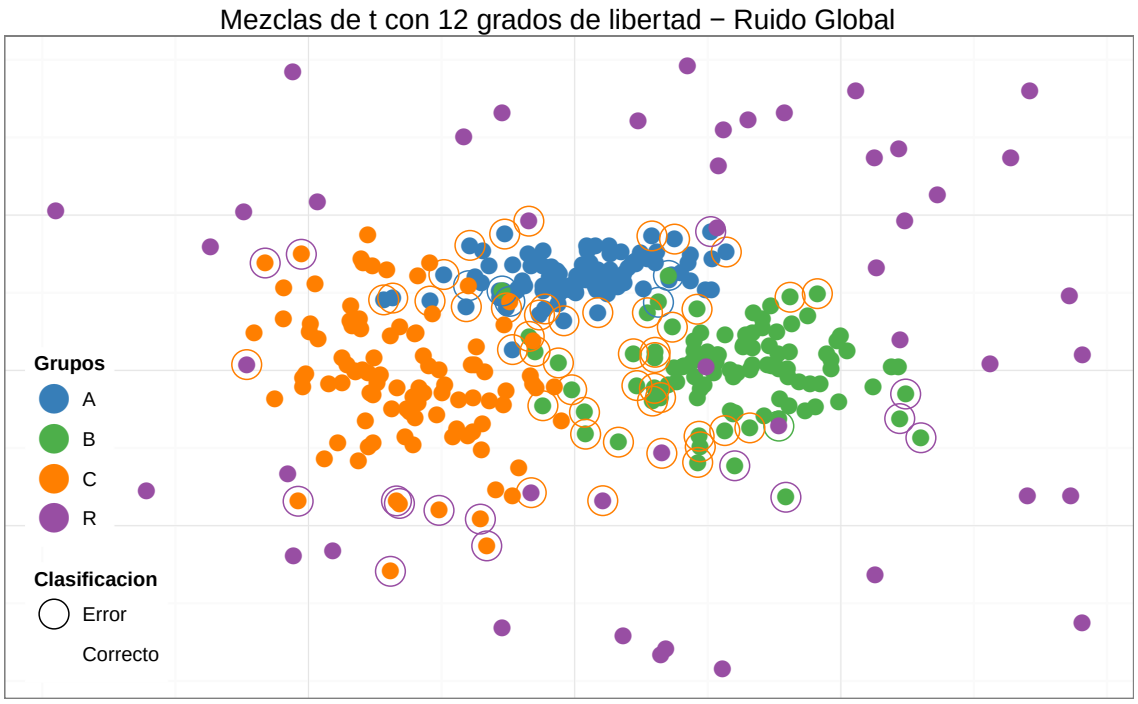

Figura 4.3: Clasificación mediante Mezcla de t de Student con 12 grados de libertad

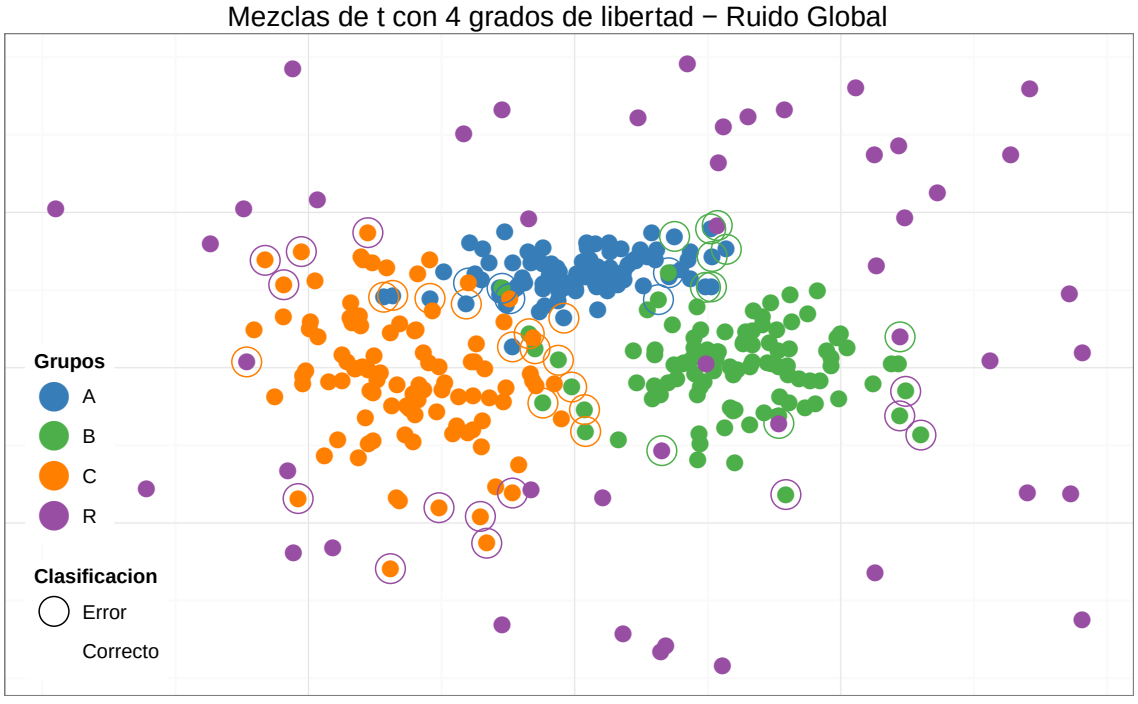

Figura 4.4: Clasificación mediante Mezcla de t de Student con 4 grados de libertad - Ruido Global

Capítulo 5

Trimming en Clustering

En este capítulo introduce en general el método propuesto por Garcia-Escudero, Gordaliza, Matran y Mayo-Iscar [24] [11], donde se trabaja con clusters de distinto peso y dispersión, admitiendo un proporción α de outliers.

El análisis que se desarrolla es el de podar las observaciones menos confiables, el cuál no es para nada trivial, debido a que no existen direcciones privilegiadas en la búsqueda y que muchas veces es necesario eliminar observaciones "puente" entre los cluster. Si bien se han introducido métodos de penalización y poda en el método de k-Medias ellos muestran poseer mejores resultados en términos de robustez, además se levanta el supuesto implícito de que la matriz de covarianza es la misma y esférica para los grupos en el algoritmo de k-Medias.

Se afronta el problema desde una perspectiva diferente al capítulo anterior, en lugar de dotar a las distribuciones de colas más pesadas para que los outliers tengan un menor impacto sobre las estimaciones, los poda, partiendo de que son valores anómalos, que no provienen del modelo.

5.1. Introducción

Se considera la presencia de una proporción α de outliers. La función de verosimilitud para el conjunto de datos $x_1, \ldots, x_n$ en este caso es

$$\left[\prod_{j=1}^{k}\prod_{i\in R_j} f(x_i;\mu_j,\Sigma)\right]\left[\prod_{i\notin R} g_{\Psi_i}(x_i)\right], \tag{5.1}$$

con $R = \cup_{j=1}^{k} R_j$ y $\#R = [n(1-\alpha)]$.

El parámetro k denota el número total de grupos, R_j contiene los índices de las observaciones "regulares" asignadas al grupo j y $f(\cdot;\mu;\Sigma)$ es la función de densidad

de una distribución normal p-variada con media μ y matriz de covarianza Σ, mientras que las g_{Ψ_i} son alguna función de densidad en $\mathbb{R}^p$.

Si se elige $\Sigma = \sigma^2 I$, entonces se está realizando el método de k-Medias podadas. Gallegos y Ritter (2005) [38] mostraron que la maximización se reduce a la consideración de la parte regular de las observaciones bajo algunos supuestos razonables para las g_{Ψ_i}s siempre y cuando las observaciones "no regulares" pueden ser vistas meramente como "ruido". El problema de maximizar esta verosimilitud es costoso computacionalmente. Para alivianar este problema es donde participa el algoritmo ECM[1].

El supuesto de igualdad de matrices de covarianza para los grupos puede ser restrictivo en muchos contextos, y seria un supuesto a levantar. Desafortunadamente, este problema de clustering robusto es notablemente complejo. Gallegos y Ritter (2005) mantienen el supuesto de igualdad de matrices de covarianza y levantan el supuesto de esfericidad de las matrices de covarianzas, modelo que llaman *"spurious-outliers model"* (supuesto de igualdad que eliminan en el 2009).

Actualmente se han encontrado respuestas parciales, en general, imponiendo restricciones sobre las distintas matrices de covarianzas (se admite una moderada diferencia en las dispersiones). Es fácil ver la no acotación de la funcion objetivo perseguida, como cada punto de los datos hace surgir una singularidad en el borde del espacio paramétrico. Si se utilizan métodos no restringidos frecuentemente se encuentran clusters conteniendo unos pocos puntos, ya sea muy juntos o casi estando en un espacio de menor dimensión, y la aplicación de algún tipo de restricción permitiría obtener particiones más interesantes o informativas.

Como forma de poner restricciones al problema, Gallegos (2002) [9] propone normalizar las covarianzas para que tengan un determinante de unidad cuando se computan las distancias de Mahalanobis en el paso de "concentración". Esto sirve para evitar el efecto pernicioso de las diferentes escalas y beneficiarse de la lógica detrás del algoritmo Fast-MCD.

El procedimiento de Gallegos funciona adecuadamente cuando los grupos tiene escalas similares, pero claudica cuando escalas de grupos muy diferentes están involucradas. Normalizar las covarianzas para tener un determinante de unidad puede ser muy restrictivo y, seguramente, tales restricciones fuertes no se necesitan siempre. Aún más, parece también adecuado el incorporar restricciones directamente en la definición del problema en vez de aparecer (artificialmente) en el algoritmo.

[1] ver apéndice B

Por tanto serán planteadas restricciones para el problema de clustering robusto heterogéneo, incorporado a través de una restricción en el cociente de los valores propios, donde c será una constante que controlará la fuerza de la restricción planteada.

La introducción de algunos términos π_js de pesos serán considerados para tratar con grupos de distintos pesos, lo que hace el problema más general pero más duro de ser trabajado.

Gallegos y Ritter [39] propone restricciones sobre las matrices de varianzas y covarianzas a partir del orden de *Löwner* las cuáles son llamadas restricciones HDBT (en referencia histórica a sus creadores Hathaway, Dennis, Beale y Thompson).

Si denotamos $V_1, V_2, \ldots, V_g$ matrices de varianzas y covarianzas que cumplen:

$$V_j \succeq cV_l, \quad 1 \leq j, l \geq g, \tag{5.2}$$

para alguna constante $c > 0$. Donde el símbolo $\succeq$ establece un orden entre las matrices semidefinidas positivas y c es necesariamente acotada entre $(0, 1]$. Si $c = 1$ estamos en el caso de homosedasticidad. Se define la proporción HDBT de la g-upla $V = (V_1, V_2, \ldots, V_g)$ al máximo valor de c que verifiquen las restricciones (5.2). Se puede observar que

$$r_{HDBT}(V) = \text{máx}\,\{c/V_j \succeq cV_l \quad j, l\} = \min_{j,l,k} \lambda_k(V_l^{-1/2} V_j V_l^{-1/2}) \tag{5.3}$$

donde $\lambda_1(A), \ldots, \lambda_d(A)$ denotan los valores propios de la matriz de A.

Gallegos y Ritter [39] demuestran que las restricciones HDBT son suficientes para asegurar la existencia del máximo de lo que llaman criterio del determinante trimeado (TDC).

5.2. Clustering Robusto con Restricciones del Cociente de Valores Propios

Sean $x_1, \ldots, x_n$ los datos disponibles en algún espacio p-dimensional. Sea $f(x; \mu; \Sigma)$ la densidad de una distribución normal de la forma

$$f(x; \mu; \Sigma) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp(-(x - \mu)\prime \Sigma^{-1}(x - \mu)/2),$$

Denotamos una medida de probabilidad P actuando sobre una función f por $Pf(\cdot) = \int f(x) dP(x)$.

Se comienza modificando el modelo de "autlaier espuriuos" considerado en Gallegos y Ritter (2005) [38].

Primero, como se menciono antes, se consideran diferentes matrices de dispersión Σ_is como en Gallegos (2001, 2002). Se asume la presencia de algunos pesos subyacentes, π_js con $\sum_{j=1}^{k} \pi_j = 1$ asociados a las distribuciones del conjunto de observaciones "regulares". Esto lleva a la maximización de

$$\left[\prod_{j=1}^{k}\prod_{i\in R_j} \pi_j f(x_i;\mu_j,\Sigma_j)\right]\left[\prod_{i\notin R} g_{\Psi_i}(x_i)\right], \tag{5.4}$$

con $R = \cup_{j=1}^{k} R_j$ y $\#R = n - [n\alpha]$. Adicionalmente, las restricciones sobre los valores propios de las matrices Σ_j serán introducidos mas tarde para evitar singularidades. Si las g_Ψ's satisfacen la condición

$$\arg\max_{\mathcal{R}} \max_{\mu_j,\Sigma_j} \prod_{j=1}^{k}\prod_{i\in R_j} \pi_j f(x_i;\mu_j,\Sigma_j) \subseteq \arg\max_{\mathcal{R}} \prod_{i\notin \cup_{j=1}{}^{k} R_j} \max_{\Psi_i} g_{\Psi_i}(x_i),$$

donde $\mathcal{R}$ denota el conjunto de todas las particiones de índices $1,\ldots,n$ en k grupos de observaciones regulares, R, y un grupo conteniendo las no regulares, con $\#R = n-[n\alpha]$. Esta condición se cumple bajo algunos supuestos razonables para las g_{Ψ_i}s siempre y cuando las observaciones "irregulares" sean vistas como mero "ruido".

Se usarán algunas *funciones de asignación*, z_js, diciendo a cuál clase *todo* punto x en $\mathbb{R}^p$ es asignado (no sólo las observaciones de la muestra, x_i's son clasificadas). Se utiliza un enfoque 0-1 "seco" donde x es asignado a la clase j si $z_j(x) = 1$ o es podada si $z_0(x) = 1$.

Con estas funciones, asumiendo que las g_{Ψ_i}'s pueden ser omitidas, podemos ver nuevamente el problema en 5.4 a la maximización de

$$\prod_{i=1}^{n}\left[\prod_{j=1}^{k} \pi_j^{z_j(x_i)} f(x_i;\mu_j,\Sigma_j)^{z_j(x_i)}\right],$$

siendo z_j las funciones 0-1 definidas en todo el espacio de la muestra verificando $\sum_{j=0}^{k} z_j(x_i) = 1$ y $\sum_{i=1}^{n} z_0(x_i) = [n\alpha]$.

Tomando logaritmos para simplificar la expresión se obtiene la formulación del problema.

Problema de Clustering Robusto Dada una medida de probabilidad P, se busca la maximización de

$$P\left[\sum_{j=1}^{k} z_j(\cdot)(\log\pi_j + \log f(\cdot,\mu_j,\Sigma_j))\right], \tag{5.5}$$

realizada en términos de las funciones de asignación:

$$z_j : \mathbb{R}^p \to \{0, 1\}, \text{ de forma que } \sum_{j=0}^{k} z_j = 1 \text{ y } Pz_0(\cdot) = \alpha,$$

y los parámetros $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \Sigma_1, \ldots, \Sigma_k)$ correspondientes a los pesos $\pi_j \in [0, 1]$ con $\sum_{j=1}^{k} \pi_j = 1$, vectores de medias $\mu_j \in \mathbb{R}^p$ y matrices de $p \times p$ simétricas semidefinidas positivas Σ_j, con $j = 1, \ldots, k$.

Si P_n denota la medida empírica, $P_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$. Si se remplaza P por P_n en el problema previo para recuperar el problema original de la muestra (notar que, quizás, $P_n z_0(\cdot) = \alpha$ no puede ser exactamente alcanzado).

Se introducen restricciones de valores propios a las matrices de covarianzas que permite evitar las singularidades introducidas por la posibilidad de Σ_js muy diferentes, mediante el control del cociente entre el máximo y el mínimo de los valores propios de esas matrices:

(ER) Restricciones sobre el Cociente de Valores Propios Se fija una constante $c \geq 1$ de forma que

$$M_n / m_n \leq c$$

para

$$M_n = \max_{j=1,\ldots,k} \max_{l=1,\ldots,p} \lambda_l(\Sigma_j) \text{ y } m_n = \min_{j=1,\ldots,k} \min_{l=1,\ldots,p} \lambda_l(\Sigma_j)$$

donde $\lambda_l(\Sigma_j)$ son los valores propios de las matrices Σ_j, $j = 1, \ldots, k$ y $l = 1, \ldots, p$.

Se denota por Θ_c al conjunto constituido por los θs que cumplen la condición ER para un c dado.

Notar que, la restricción más fuerte posible surge de establecer $c = 1$. En este caso en particular, el método propuesto puede ser visto como un procedimiento de k-Medias podadas con pesos. Sin embargo, la ventaja principal de este enfoque yace en el hecho de que el parámetro c permite alcanzar cierta (controlada) libertad en como se puede manejar las diferentes dispersiones de los grupos.

La condición se cumple trivialmente si la distribución P subyacente es continua o si es una medida empírica P_n correspondiente a una muestra de una distribución absolutamente continua (para n suficientemente grande): la distribución P no está concentrada en k puntos después de remover una masa de probabilidad igual a α.

Dado $\theta \in \Theta_c$, se considera alguna *función discriminante* definida como

$$D_j(x; \theta) = \pi f(x; \mu_j, \Sigma_j),$$

y

$$D(x;\theta) = \text{máx}\{D_1(x;\theta), \ldots, D_k(x;\theta)\}.$$

Estas funciones sirven para determinar las observaciones más "outliers". Para una elección fija de θ, cuanto $D(x;\theta)$ más chica sea para un x dado, más se lo considerara "outlier".

Usando las definiciones previas, para un θ dado y una medida de probabilidad P, se define,

$$G(\cdot;\theta,P) : u \in \mathbb{R} \to P\left[I_{[0,u]}(D(\cdot;\theta))\right], \tag{5.6}$$

y

$$R(\theta,P) := G^{-1}(\alpha;\theta,P) = \inf_u\{G(u;\theta,P) \geq \alpha\}$$

(notar que si X es una variable aleatoria con distribución dada por P entonces $R(\theta,P)$ es el α-cuantil de la variable aleatoria $D(X;\theta)$).

Con esta notación, se tiene la siguiente caracterización para las funciones de los z_js:

Lema 1 (Caracterización de z_j) *Para la medida de probabilidad P, usando funciones discriminantes $D_j(x;\theta)$, el problema de Clustering Robusto puede ser simplificado a la maximización sólo en términos de θ de*

$$\theta \to L(\theta,P) := P\left[\sum_{j=1}^{k} z_j(\cdot;\theta)\log D_j(\cdot,\theta)\right], \tag{5.7}$$

donde las funciones de asignación son obtenidas de θ como

$$z_j(x;\theta) = I\{x : \{D(x;\theta) = D_j(x;\theta)\} \cap \{D_j(x;\theta) \geq R(\theta,P)\}\},$$

y

$$z_0(x;\theta) = 1 - \sum_{j=1}^{k} z_j(x;\theta).$$

En otras palabras, se asigna x a la clase j con el valor más alto de la función discriminante, $D_j(x;\theta)$, o x es podado cuando todos los $D_j(x;\theta)$s (y consecuentemente $D(x;\theta)$) son mas chicos que $R(\theta,P)$. Una regla para romper empates en los valores de las funciones discriminantes también es también necesaria. Por ejemplo, se podría aplicar el orden lexicógrafo.

5.3. Existencia

Considerar una sucesión $\{\theta_n\}_{n=1}^{\infty} = \{(\pi_1^n, \ldots, \pi_k^n, \mu_1^n, \ldots, \mu_k^n, \Sigma_1^n, \ldots, \Sigma_k^n)\}_{n=1}^{\infty}$ de forma tal que

$$\lim_{n\to\infty} L(\theta_n, P) = \sum_{\theta\in\Theta_c} L(\theta, P) = M > -\infty, \tag{5.8}$$

(la acotación 5.8 puede ser fácilmente obtenida solo considerando $\pi_1 = 1$, $\mu_1 = 0$, $\Sigma_1 = I$, y estableciendo los otros pesos a 0 con elecciones arbitrarias de medias y varianzas).

Ya que $[0,1]^k$ es un conjunto compacto, podemos extraer una subsucesión de θ_{nn}^{∞} (que sera denotada como la original) de forma que

$$\pi_j^n \to \theta_j \in [0,1] \text{ para } 1 \leq j \leq k, \tag{5.9}$$

y también cumpliendo para alguna $g \in \{0, 1, \ldots, k\}$ (reetiquetar puede ser necesario) que

$$\mu_j^n \to \mu_j \in \mathbb{R}^p \text{ para } 0 \leq j \leq g \text{ y } \min_{j>g} \|\mu_j^n\| \to \infty. \tag{5.10}$$

Con respecto a las matrices de dispersiones, si se asume la restricción ER, también podemos considerar subsucesiones adicionales verificando una (y sólo una) de estas posibilidades:

$$\Sigma_j^n \to \Sigma_n \text{ para } 1 \leq j \leq k, \tag{5.11}$$

$$M_n = \max_{j+1,\ldots,k} \max_{l=1,\ldots,p} \lambda_l(\Sigma_j) \to \infty, \tag{5.12}$$

o

$$m_n = \min_{j=1,\ldots,k} \min_{l=1,\ldots,p} \lambda_l(\Sigma_j) \to 0. \tag{5.13}$$

Lema 2 (Convergencia de una Subsucesión) *Si ER se cumple y si P satisface (2.3), entonces las convergencias (5.12) o (5.13) para las matrices de covarianzas no es posible. Entonces, se puede encontrar subsucesión de Σ_j^n convergiendo hacia algunas matrices $\Sigma_j, j = 1, \ldots, k$.*

Lema 3 (Condición suficiente de convergencia) *Cuando ER y (2.3) se cumplen, si todo π_j en (2.7) verifica $\pi_j > 0$, $j = 1, \ldots, k$, entonces $g = k$ en (2.8) (e.g., los centros μ_j^n no están permitidos de crecer arbitrariamente en norma).*

Teorema 9 (Existencia) *Si (2.3) se cumple para la medida de probabilidad P, entonces existe algún $\theta \in \Theta_c$ de forma tal que el máximo de (2.6) bajo las restricción ER es alcanzado.*

La existencia de tal θ surge fácilmente de los Lemas 2 y 3:

1. Si $\pi_j^n \to \pi_j$ para $1 \leq j \leq k$, entonces la elección de θ es obvia.

2. Ahora asume que $\pi_j^n \to \pi_j > 0$ con $\pi_j > 0$ para $j \geq g$ y $\pi_j = 0$ para $g < j\theta \leq k$. Se definen los pesos π_j como

$$\pi_j = \lim_{n\to\infty} \pi_j^n \text{ para } j = 1, \ldots, g \text{ y } \pi_{g+1} = \cdots = \pi_k = 0.$$

   Análogamente, tomar $\mu_h = \lim_{n\to\infty} \mu_j^n$ y $\Sigma_j = \lim_{n\to\infty} \Sigma_j^n$ para $j \leq k$. Las otras μ_js y Σ_js pueden ser elegidas arbitrariamente (pero con los valores propios de las Σ_js verificando la restricción impuesta por ER). $\square$

Aunque se tomen pesos $\pi_j = 0$, esto no es un impedimento cuando se toma $\log \pi_j$ debido a que en este caso $z_j(\cdot;\theta) \equiv 0$ y entonces el conjunto $\{x : z_j(x;\theta) = 1\}$ es vacío. Notar que la presencia de grupos con peso cero aparecen en la práctica. Por ejemplo, cuando $k = 2$, $c = 1$, $\alpha = 0$ y P es la distribución $N(0,1)$ en la recta real, se puede ver que $\theta = (\pi_i, \pi_2, \mu_1, \mu_2, \sigma_1, \sigma_2) = (1, 0, 0, \mu_2, 1, 1)$ es la solución óptima para cada $\mu_s \in \mathbb{R}$.

5.4. Consistencia

Dada $\{x_n\}_{n=1}^{\infty}$ una muestra aleatoria i.i.d. de la distribución de probabilidad subyacente (desconocida) P, sea $\{\theta_n\}_{n=1}^{\infty} = \{(\pi_1, \ldots, \pi_k^n, \mu_1^n, \ldots, \mu_k^n, \Sigma_1^n, \ldots, \Sigma_k^n)\}_{n=1}^{\infty} \subset \Theta_c$ la sucesión de los estimadores de la muestra obtenidos por resolver el problema para las medidas empíricas $\{P_n\}_{n=1}^{\infty}$ con la restricción de valores propios definida por ER para una constante fija $c \geq 1$.

La sección 2.2 muestra que tal secuencia siempre existe, para un n suficientemente grande siempre y cuando P es una distribución absolutamente continua verificando (2.3). Notar que aunque notación similar a la aplicada en la sección previa será usada, aquí el indice n indicara la dependencia en una muestra aleatoria de tamaño n para B.

5.4.1. Acotación de los estimadores muestrales

Se observará primero que existe un conjunto compacto $K \subset \Theta_c$ tal que $\theta_n \in K$ para n suficientemente grande con probabilidad 1.

Lema 4 *Si P es una distribución absolutamente continua, los elementos de las matrices Σ_j^n están uniformemente acotadas con probabilidad 1.*

Lema 5 *Si P es una distribución absolutamente continua, entonces se pueden elegir los centros empíricos μ_j^n, $j = 1, \ldots, k$, de forma tal que sus normas estén uniformemente acotadas con probabilidad 1.*

Lema 6 *Dado un conjunto compacto K, las clases de funciones:*

$$\mathcal{H}_1 := \{I_{[u,\infty)}(D(\cdot;\theta)) : \theta \in K, u \geq 0\} \tag{5.14}$$

y

$$\mathcal{H}_\in := \{I_{[u,\infty)}(D(\cdot;\theta)) \sum_{j=1}^{k} z_j^*(\cdot;\theta) \log D_j(\cdot;\theta) : \theta \in K, u \geq 0\}, \tag{5.15}$$

son clases de Glivenko–Cantelli, donde $z_j^*(x;\theta) = I\{x : D(x;\theta) = D_j(x;\theta)\}$ (todas las observaciones en $\mathbb{R}^p$ son asignadas a alguna clase sin ser podadas usando las $z_j^* s$).

Lema 7 *Sea P una distribución absolutamente continua con una función de densidad estrictamente positiva. Entonces, para cada subconjunto compacto K, se tiene que*

$$\sup_{\theta \in K} |R(\theta; P_n) - R(\theta; P)| \to 0, P - a.e.. \tag{5.16}$$

Ahora se puede enunciar el resultado principal en este capítulo, el cual es el resultado de convergencia.

Teorema 10 (Consistencia) *Asumir que P tiene una función de densidad estrictamente positiva y que θ_0 es el único máximo, bajo la restricción ER. Si $\theta_n \in \Theta_c$ denota la versión de la muestra del estimador basado en la medida empírica P_n, entonces $\theta_n \to \theta_0$ casi seguramente.*

Notar que la condición de unicidad es necesaria para establecer el resultado de consistencia.

Desafortunadamente, esta propiedad no siempre se cumple. Por ejemplo, pensar en una mixtura simétrica P en la recta real con dos modas bien separadas, un nivel alto de podado y $k = 1$. La propiedad de unicidad era ya era necesaria para establecer el mismo resultado de consistencia para k-Medias podadas y, aun en este caso mas simple, el enunciado de los resultados generales de unicidad eran difíciles (ver Observación 4.1 en García–Escudero et al. 1999).

Sin embargo, como en el problema de las k-Medias podadas, se cree que es bastante raro el encontrar una distribución donde esta unicidad falle, cuando se trata con datos "razonables" para el agrupamiento y cuando los parámetros k y α han sido propiamente escogidos.

5.5. El algoritmo TCLUST

El problema empírico presentado tiene obviamente una complejidad computacional muy alta. Un algoritmo exacto parece no ser feasible aún para tamaños de muestra moderados. Entonces la existencia de un algoritmo adecuado para resolver aproximadamente el problema de la muestra puede ser tan importante como el procedimiento en sí mismo.

El algoritmo TCLUST, es un algoritmo basado en el principio EM, planteado para buscar soluciones aproximadas. El EM es el método usual para obtener una solución al problema de la mezcla de verosimilitudes (Dempster et al. 1997). Aquí, se sigue un enfoque "seco" donde cada punto es asignado únicamente a un cluster. Las restricciones sobre los valores propios serán incorporadas a través del algoritmo de Dykstra (1983).

El algoritmo TCLUST puede ser descrito de la siguiente forma:

1. Seleccionar valores aleatorios para los centros m_j^0s, las matrices de covarianzas S_j^0s y los pesos de los grupos p_{j^0}s para $j = 1, \ldots, k$.

2. Desde el $\theta^l = (p_1^l, \ldots, p_k^l, m_1^l, \ldots, m_k^l, S_1^l, \ldots, S_k^l)$ retornado por la iteración previa:

   a) Obtener $d_i = D(x_{i,\theta^l})$ para las observaciones $\{x_1, \ldots, x_n\}$ y mantener el conjunto H teniendo las $[n(1-\alpha)]$ observaciones con las mas grandes d_is.

   b) Dividir H en $H = \{H_1, \ldots, H_k\}$ con $H_j = \{x_i \in H : D_j(x_i, \theta^l) = D(x_i\theta^l)\}$.

   c) Obtener el número de datos n_j en H_j, su media y matriz de covarianzas muestrales, m_j y S_j, $j = 1, .., k$.

   d) Considere la descomposición en valores propios de $S_j = U_j' D_j U_j$ donde U_j es una matriz ortogonal y $D_j = diag(\Lambda_j)$ es una matriz diagonal (con los elementos en la diagonal dados por el vector Λ_j). Si el vector entero de valores propios $\Lambda = (\Lambda_{1,\ldots,\Lambda_k})$ no satisface la restricción de valores propios, obtener un nuevo vector $\tilde{\Lambda} = (\tilde{\Lambda}_1, \ldots, \tilde{\Lambda}_k)$ a través del algoritmo de Dykstra que obedezca la restricción ER y que $\|\tilde{\Lambda} - \Lambda^{-1}\|^2$ sea lo mas chico posible. Λ^{-1} denota el vector compuesto por los inversos de los elementos del vector Λ. Notar que la restricción ER para Λ corresponde a la misma restricción ER aplicada a Λ^{-1}.

   e) Actualizar θ^{l+1} usando:

   - p_j^{l+1} <-- $n_j/[n(1-\alpha)]$

- $m_j^{l+1} \leftarrow m_j$
- $S_j^{l+1} \leftarrow U_j' \tilde{D}_j U_j$ y $\tilde{D}_j = diag(\tilde{\Lambda}_j)^{-1}$

3. Realizar F iteraciones del proceso descrito en el paso 2 (valores moderados para F son usualmente suficientes) y computar la función de evaluación $L(\theta^F; P_n)$.

4. Obtener valores de partida aleatorios (e.g. comenzar desde el paso 1) varias veces, mantener las soluciones que llevan a valores mínimos de $L(\theta^F; P_n)$ e iterar completamente sobre ellos para elegir el mejor.

Las probabilidades "a posteriori" computadas (paso E), $D_j(x_i, \theta^l) = p_j f(x_i; m_j, S_j)$, son convertidas a una clasificación discreta donde se deja sin asignar la proporción α de observaciones las cuales son las más difíciles de clasificar. Es fácil de ver que esto lleva a una asignación óptima.

Después se obtiene un nuevo θ^{l+1} por maximizar (paso M) la esperanza condicional una vez que todas las observaciones no podadas han sido asignadas a los grupos. La proposición 3 garantiza que el algoritmo presentado puede ser usado para realizar esta maximización.

Notar que la obtención de las matrices de dispersión óptimas se descompone en la búsqueda de los correspondientes valores y vectores propios óptimos. Para cada elección de valores propios, la elección de los mejores vectores propios surge simplemente de los vectores propios unitarios de la matriz de covarianzas muestral de las observaciones asignadas a cada grupo. Esta descomposición es de alguna forma similar a la considerada en la propuesta de Gallegos, donde las "formas" y las "escalas" son tratadas de forma separada.

Si se ve a $D(x_i, \theta^l)$ como medida inversa de atípico para la observación x_i con respecto a la elección de θ^l, entonces el paso 2 puede ser visto como cierto tipo de pasos de "concentración". Garcia-Escudero y Gordaliza (2006) analizan otros intentos para extender el principio del paso de "concentración" a la configuración de clustering robusto heterogéneos.

Recordar que la esquema de inicialización aleatoria (paso 1) y el refinamiento final (paso 4) eran muy importantes en el algoritmo Fast-MCD. Para inicializar el procedimiento en el paso 1, se ha visto que simplemente elegir k puntos de la muestra para los centros, k matrices identidad para las matrices de covarianzas y los mismos pesos para los grupos (iguala $1/k$) provee una un punto de partida razonable en la mayoría de los casos.

Con respecto a la restricción de valores propios, se podría necesitar $\Lambda = (\Lambda_1, \ldots, \Lambda_k)$ con $\Lambda_j = (\lambda_{1,j}, \ldots, \lambda_{p,j})$ pertenecientes al cono $\mathcal{C}$, donde

$$\mathcal{C} = \{(\Lambda_1, \ldots, \Lambda_k) \in \mathcal{R}^{p \times q} : \lambda_{u,v} - c \cdot \lambda_{r,s} \leq 0 \text{ para todo } (u,v) \neq (r,s)\}. \quad (5.17)$$

Si $\Lambda \in \mathcal{C}$, se necesita remplazar L^{-1} por $\hat{\Lambda} \in \mathcal{C}$ con $\|\tilde{\Lambda} - \Lambda^{-1}\|^2$ mínimo. El algoritmo de Dykstra sirve para resolver aproximadamente ese problema, donde además de la lógica detrás del Fast-MCD (Rousseeuw y van Driessen 1999) y detrás de algoritmo de k-Medias podadas (García-Escudero et al. 2003), también subyacerán mínimos cuadrados con restricciones cuando $\mathcal{C}$ es la intersección de varios conos cerrados convexos mediante el reordenamiento a proyecciones iterativas en los conos individuales.

Notar que $\mathcal{C}$ puede ser visto como la intersección de los conos

$$\mathcal{C}_h = \{(\Lambda_1, \ldots, \Lambda_k) \in \mathcal{R}^{p \times q} : \lambda_{u,v} - c \cdot \lambda_{r,s} \leq 0\}, h = (u,v,r,s)\},$$

y las proyecciones en los conos $\mathcal{C}_h$ son rápidas de obtener. Entonces un número fijo de proyecciones individuales pueden ser realizadas reteniendo la mejor solución alcanzada después de esas iteraciones y satisfaciendo las restricciones. Alternativamente, soluciones basadas en programación cuadrática pueden ser utilizadas (ver, Goldfarb e Idnani (1983)).

El siguiente resultado sirve para formalizar lo apropiado del algoritmo TCLUST:

Teorema 11 *Si los conjuntos $H_j = \{x_i : z_j(x_i) = 1\}, j = 1, \ldots, k$, son mantenidos fijos, el máximo de (2.2) para $P = P_n$ puede ser obtenido a través de los siguientes pasos:*

1. *Fijada μ_j y Σ_j, la mejor elección de π_j es $\pi_j = n_j/[n(1-\alpha)]$, donde $n_j = \#H_j$.*

2. *Fijada Σ_j y los valores óptimos para π_j dados en (1), la mejor elección para μ_j es la media muestral m_j de las observaciones en H_j.*

3. *Fijado los valores propios para la matriz Σ_j y los valores óptimos dados en (1) y (2), la mejor elección para el conjunto de vectores propios son los vectores propios unitarios de la matriz de covarianza S_j de las observaciones en H_j.*

4. *Con las selecciones óptimas hechas en (1), (2), y (3), la mejor elección para los valores propios corresponde a la proyección del vector conteniendo los inverso de los valores propios en el cono $\mathcal{C}$ en (3.1).*

5.6. Estudio de Simulación

Se considera el mismo escenario del capítulo anterior:

- 100 observaciones de una $N\left[\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 & 0,5 \\ 0,5 & 0,5 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & -0,5 \\ -0,5 & 0,5 \end{pmatrix}\right]$
- 50 observaciones uniformes en el cuadrado $[-10, 10]^2$

Se compara ahora la performance del algoritmo TCLUST contra el de Mezcla de Normales. De la misma forma que se viene trabajando se realizan 150 repeticiones y se estudia el porcentaje de datos correctamente clasificados .

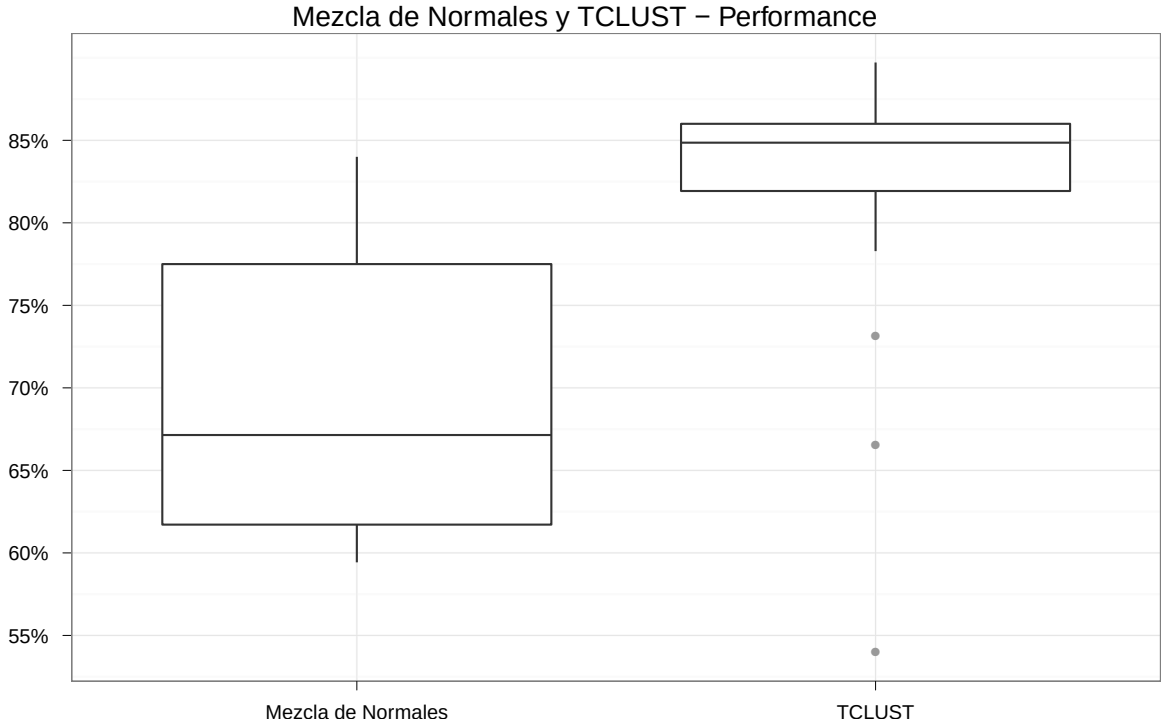

Figura 5.1: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo

Se puede apreciar que la poda de datos contribuye a una mejor eficiencia del algoritmo en presencia de ruido global. Se podría observar también que si el ruido es local aumenta en forma considerable las diferencias entre la performance de un algoritmo y otro si se sigue modelando con 3 grupos.

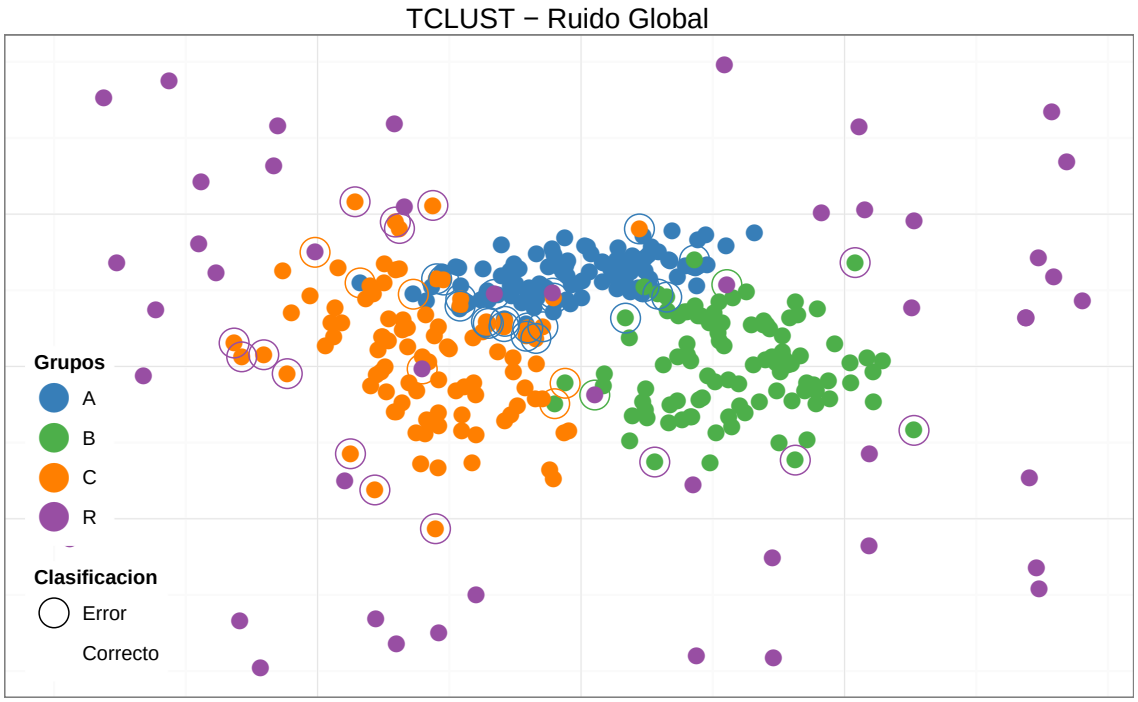

Figura 5.2: Clasificación mediante TCLUST con un 15 % de datos podados

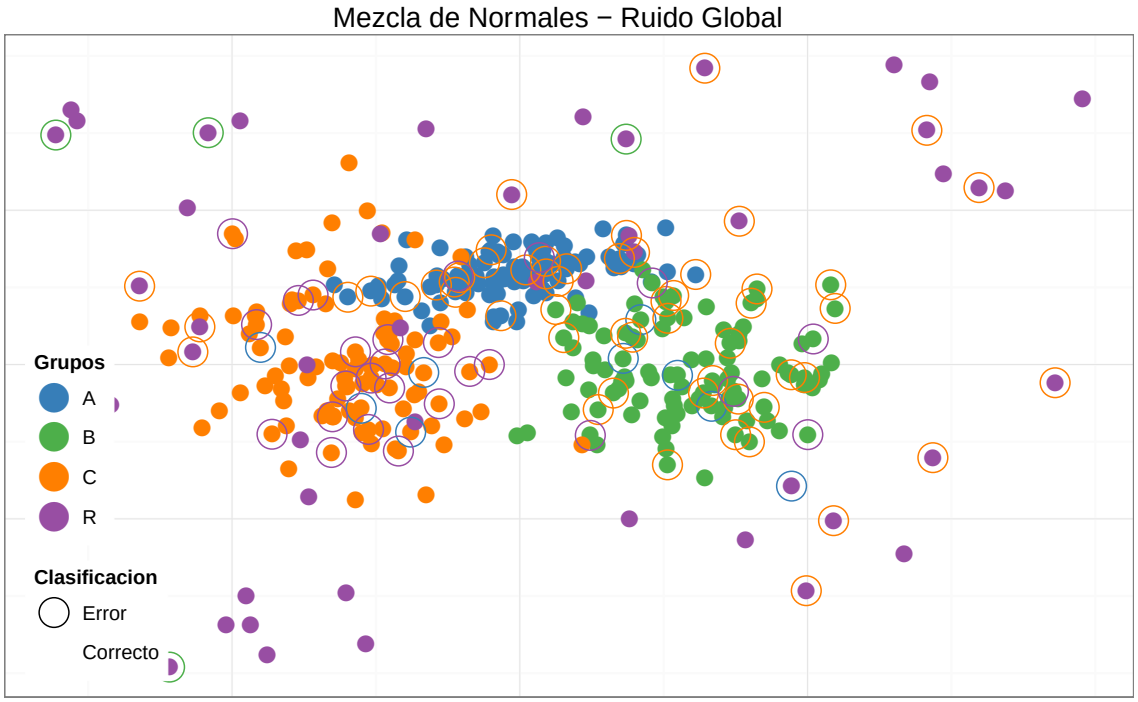

Figura 5.3: Clasificación mediante Mezcla de Normales con un 15 % de datos podados

Capítulo 6

Comparación de Performance de Algoritmos Robustos

En este capítulo se analizarán los tres algoritmos robustos propuestos según su desempeños bajo diversos escenarios.

Se realizaran modificaciones sobre la forma de los grupos (esférico o elíptico), sobre el tamaño de éstos (grupos de igual o distinto tamaño) y sobre la forma del ruido (global o local), variando el porcentaje de contaminación introducido.

6.1. Escenarios Considerados

Los escenarios a analizar son los siguientes:

Escenario 1: Se consideran tres grupos esféricos de igual tamaño y ruido global uniformemente distribuido con un porcentaje de contaminación del 10 %.

Escenario 2: Tres grupos esféricos de igual tamaño y ruido global uniformemente distribuido con un porcentaje de contaminación del 25 %. Se mantiene el modelado del caso anterior pero con una mayor cantidad de ruido.

Escenario 3: Se consideran tres grupos esféricos de igual tamaño y ruido uniforme sesgado hacia una de los cuadrantes. El porcentaje de contaminación es del 20 %.

Escenario 4: Se comienza a variar los supuestos sobre la distribución de los grupos. Se consideran grupos elípticos de distinto tamaño. El ruido es global y uniformemente distribuido, con un porcentaje de contaminación del 10 %.

Escenario 5: En este escenario se elije el paradigma más general respecto a la tipología del ruido, es decir, los grupos con distribución elíptica y porcentaje de contaminación del 20 %.

Un 10 % es ruido global sesgado hacia algunos cuadrantes, mientras que el otro 10 % es ruido local ubicado alejado de los centros de los tres clusters.

En cada uno de los escenarios anteriormente se realizan 150 repeticiones y se estudia el porcentaje de datos correctamente clasificados por:

- Variante Robusta de k-Medias
- Mezcla de Distribuciones t
- El algoritmo TCLUST

6.1.1. Escenario 1

Se considera en este caso 500 observaciones bivariadas, tres grupos conformados cada uno por 150 observaciones y un 10 % de ruido global distribuido uniformemente:

- 150 observaciones de una $N\left[\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 150 observaciones de una $N\left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 150 observaciones de una $N\left[\begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 50 observaciones uniformes en el cuadrado $[-10, 10]^2$

Como se puede apreciar en los respectivos diagramas de caja en la figura 6.8, quien clasifica mejor es el algoritmo mediante Mezcla de distribuciones t.

El poco ruido global es soportado por las colas pesadas de la distribuciones t, lo que hace posible un estimación eficiente de los centros de los clusters.

Sin embargo la variante robusta de k-Medias presenta una eficiencia no tanto menor, pero con una mayor variabilidad.

Es de hacer notar la alta variabilidad de algoritmo TCLUST. Posiblemente la poda incorrecta de algunas observaciones en cada simulación produce este efecto.

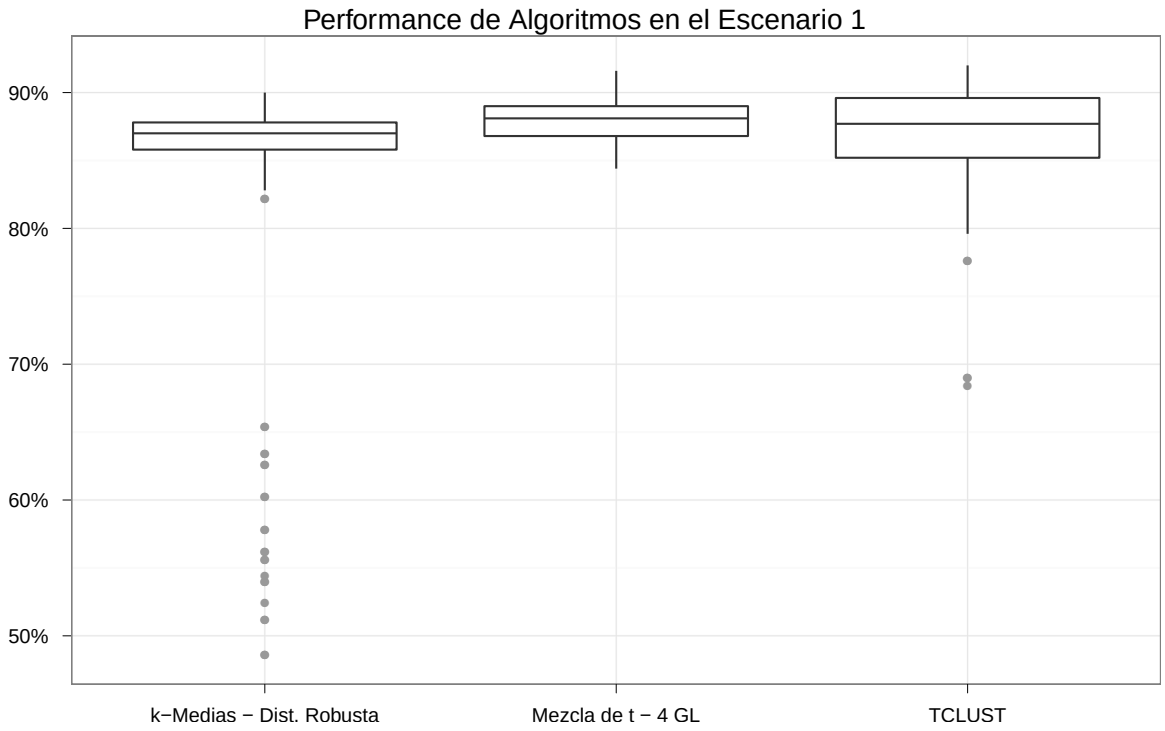

Figura 6.1: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo en el Escenario 1

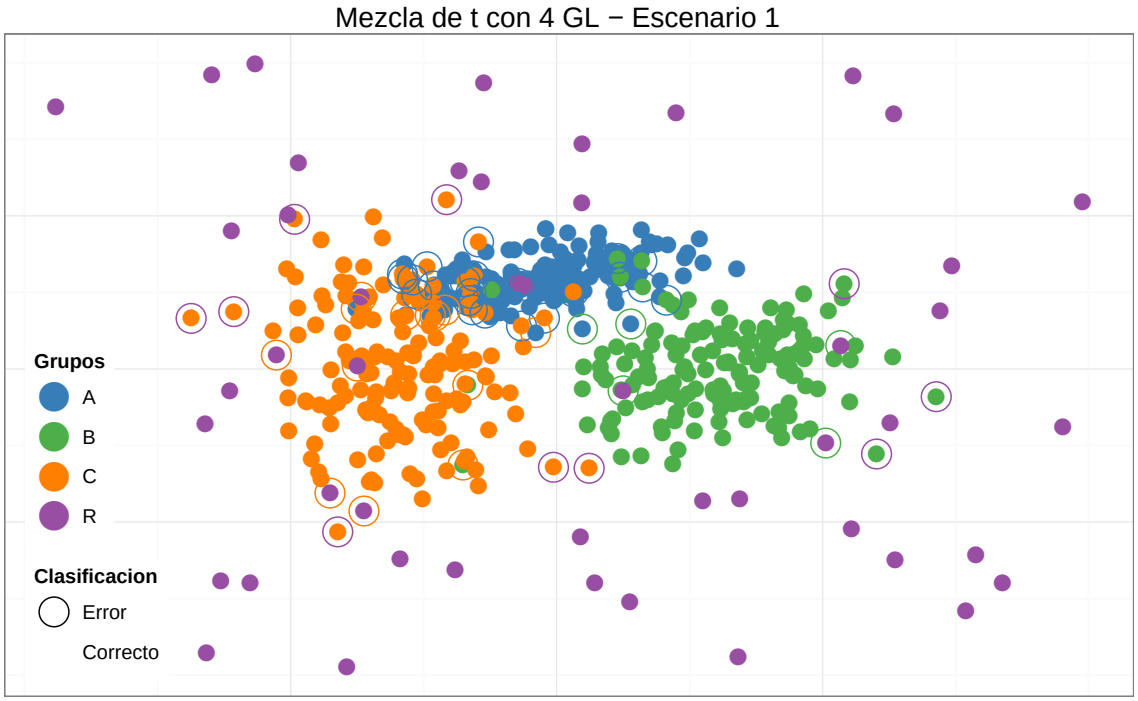

Figura 6.2: Clasificación de una simulación del escenario 1 por Mezcla de t

6.1.2. Escenario 2

Se consideran 400 datos bivariados, se mantienen los tres grupos pero ahora con 100 observaciones en cada grupo y aumenta el porcentaje de ruido a un 25 %:

- 100 observaciones de una $N\left[\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones uniformes en el cuadrado $[-10, 10]^2$

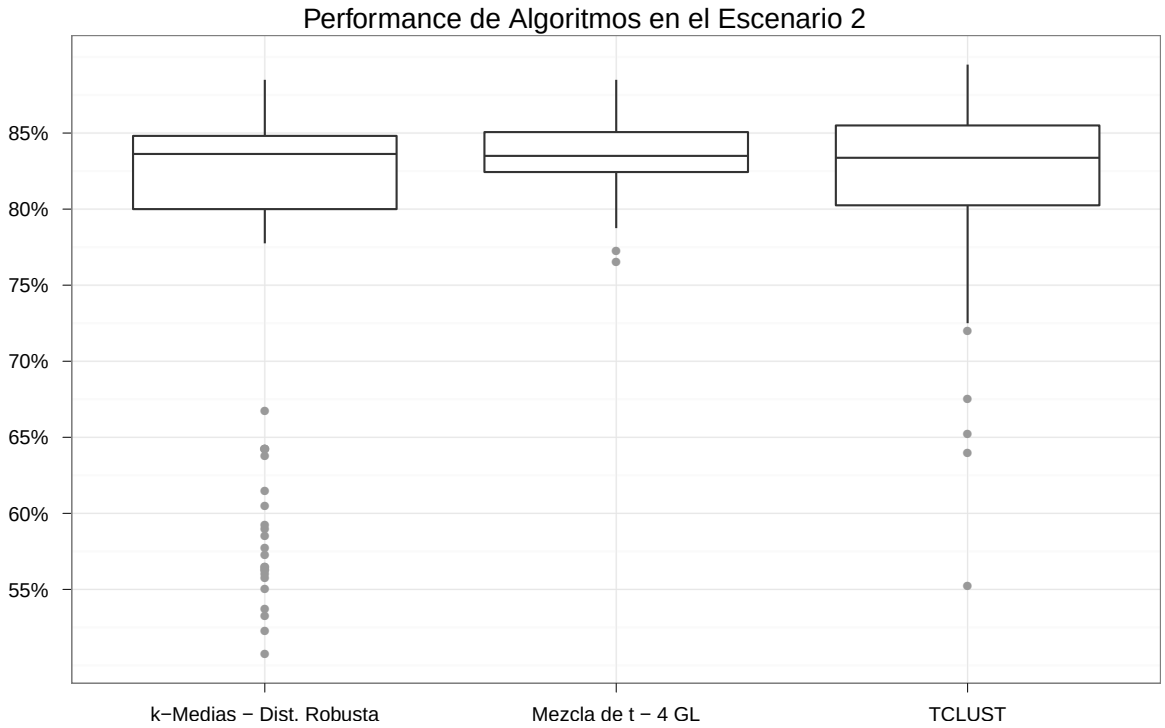

Figura 6.3: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo en el Escenario 2

Bajo esta tipología de datos, como se aprecia en los diagramas de caja expuestos en la figura 6.3, la mezcla de distribuciones t baja notoriamente su performance respecto al escenario anterior, debido al aumento en el número de outliers.

Aunque disminuyendo los grados de libertad de las distribuciones t se mejora la eficiencia, las colas de la distribución no soportan el alto número de outliers, produciendo una estimación errónea de los centros de los cluster.

La variante robusta de k-Medias, al partir de una métrica acotada entre 0 y 1, el alto porcentaje de outliers no afecta en gran medida la estimación de los centros de los grupos.

Al igual que en el escenario anterior se observa como el algoritmo TCLUST presenta una alta variabilidad frente a una contaminación global.

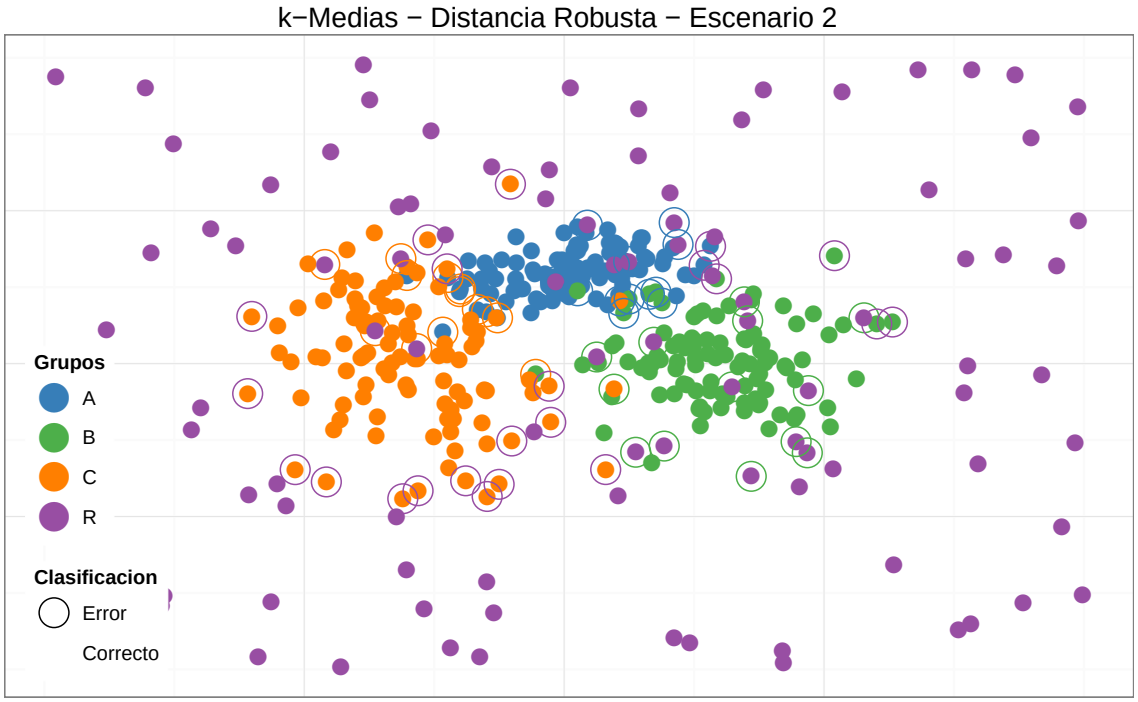

Figura 6.4: Clasificación de una simulación del Escenario 2 por k-Medias variante robusta

6.1.3. Escenario 3

En este escenario se mantienen los supuestos del caso anterior pero se sesga los datos atípicos al cuadrado $[0, 10] \times [0, 10]$:

- 100 observaciones de una $N\left[\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones uniformes en el cuadrado $[0, 10]^2$

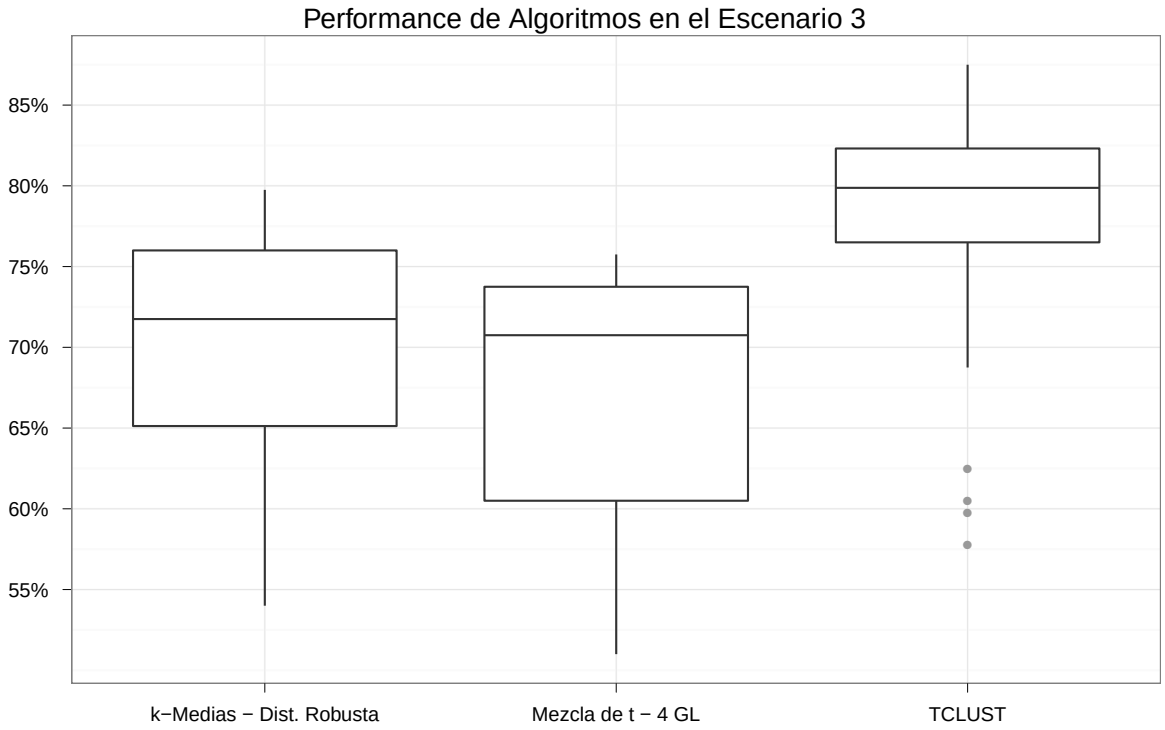

Figura 6.5: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo en el Escenario 3

Se observa en la figura 6.5 como esta tipología de outliers impacta negativamente sobre la mezcla de distribuciones, pero la variante robusta de k-Medias se mantiene casi invariante frente a este cambio.

Al igual que en el ejemplo anterior el TCLUST presenta una alta dispersión, pero en casi todas las simulaciones un mayor eficiencia que la mezcla de distribuciones t.

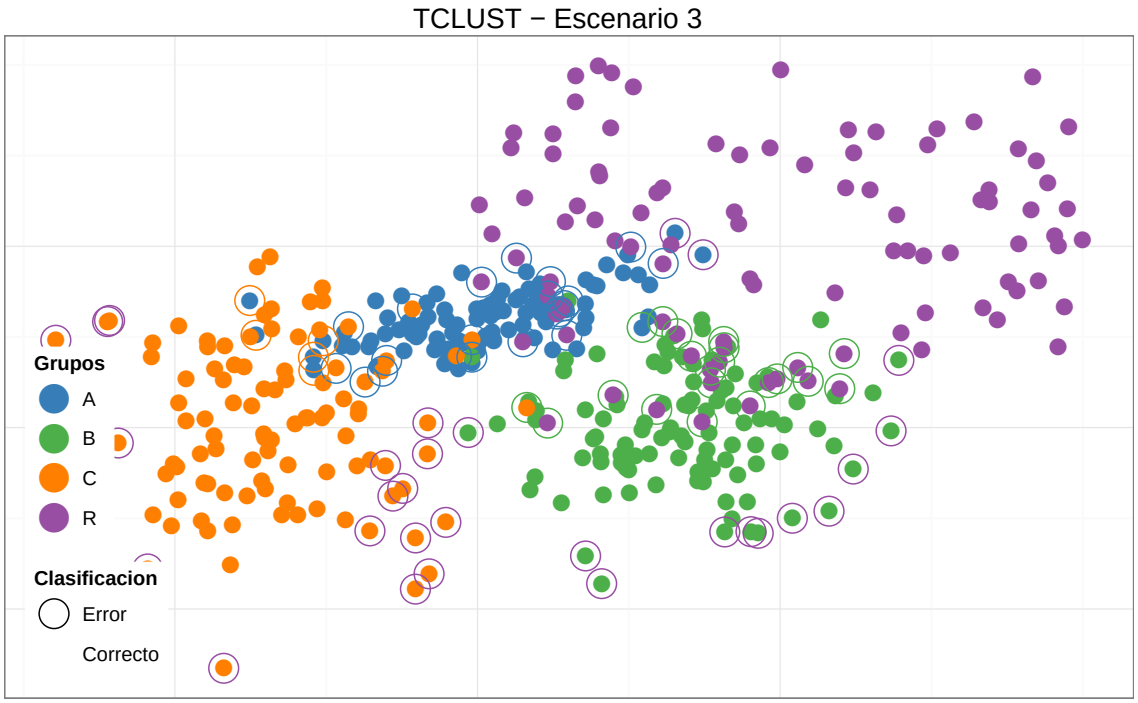

Figura 6.6: Clasificación de una simulación del Escenario 3 por TCLUST

6.1.4. Escenario 4

Se levanta ahora el supuesto de distribuciones esféricas y de igual tamaño de los grupos . Al igual que en los casos anteriores, se simulan normales bivariadas pero con diferentes matrices de varianzas y covarianzas. El ruido, como en el escenario 1, es global de un 10 % del total de la muestra, distribuido uniformemente en el cuadrado $[-10, 10] \times [-10, 10]$

- 150 observaciones de una $N\left[\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 & 0,5 \\ 0,5 & 0,5 \end{pmatrix}\right]$
- 200 observaciones de una $N\left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 100 observaciones de una $N\left[\begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & -0,5 \\ 0,5 & 5 \end{pmatrix}\right]$
- 50 observaciones uniformes en el cuadrado $[-10, 10]^2$

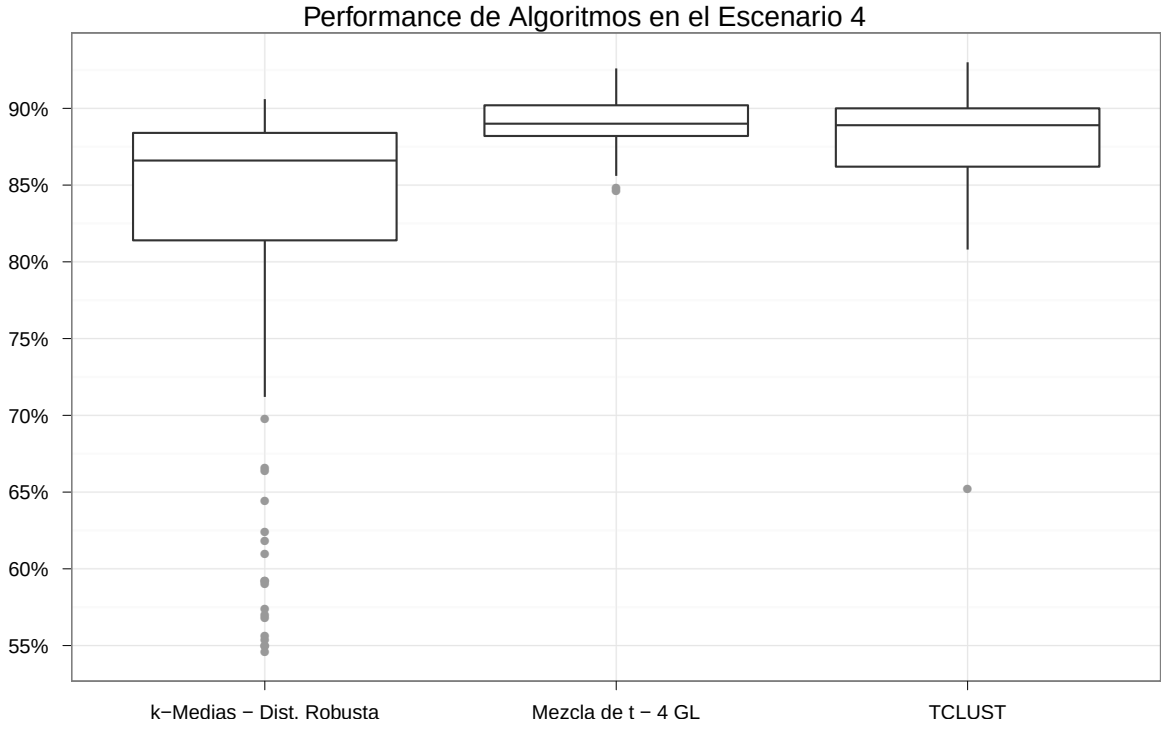

Figura 6.7: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo en el escenario 4

Es visible en los diagramas de caja representados en la figura 6.7 como la variante robusta del algoritmo de k-Medias es el que peor clasifica en estos casos.

Este algoritmo esta diseñado para detectar grupos esféricos y de igual tamaño. Cuando estos supuestos se levantan, el algoritmo pierde eficiencia. Si bien los restantes algoritmos clasifican de forma semejante, el algoritmo de distribuciones t presenta menor variabilidad.

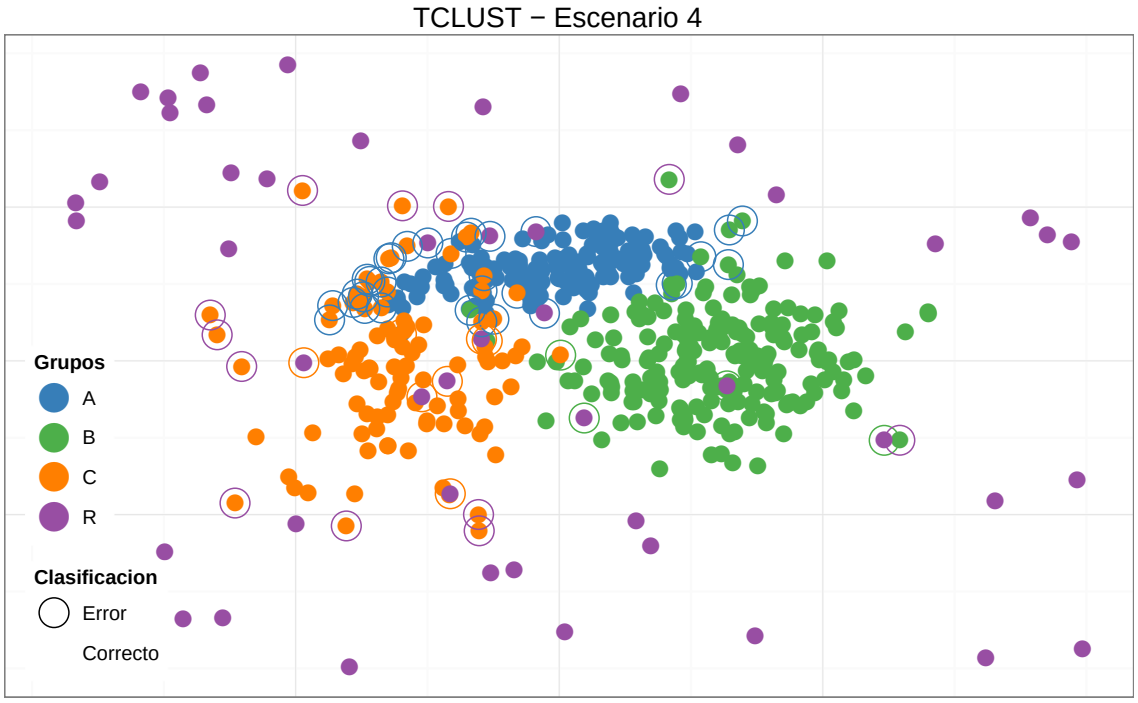

Figura 6.8: Clasificación de una simulación del Escenario 4 por TCLUST

6.1.5. Escenario 5

En este último escenario se trabaja con grupos de igual tamaño, pero los outliers se presentan de diferentes formas. Como se puede apreciar en la figura 6.10 se tiene, por un lado, cierto ruido uniforme global sesgado hacia los cuadrantes izquierdos, y por otro, un ruido local apartado de los 3 grupos en el cuadrante inferior derecho.

Se simulan 510 datos provenientes delas siguientes distribuciones,

- 150 observaciones de una $N\left[\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 & 0,5 \\ 0,5 & 0,5 \end{pmatrix}\right]$
- 150 observaciones de una $N\left[\begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]$
- 150 observaciones de una $N\left[\begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & -0,5 \\ 0,5 & 5 \end{pmatrix}\right]$
- 40 observaciones uniformes en $[-10, 0] \times [-10, 10]$
- 20 observaciones de una $N\left[\begin{pmatrix} 5 \\ -8 \end{pmatrix}, \begin{pmatrix} 2 & -0,5 \\ 0,5 & 1 \end{pmatrix}\right]$.

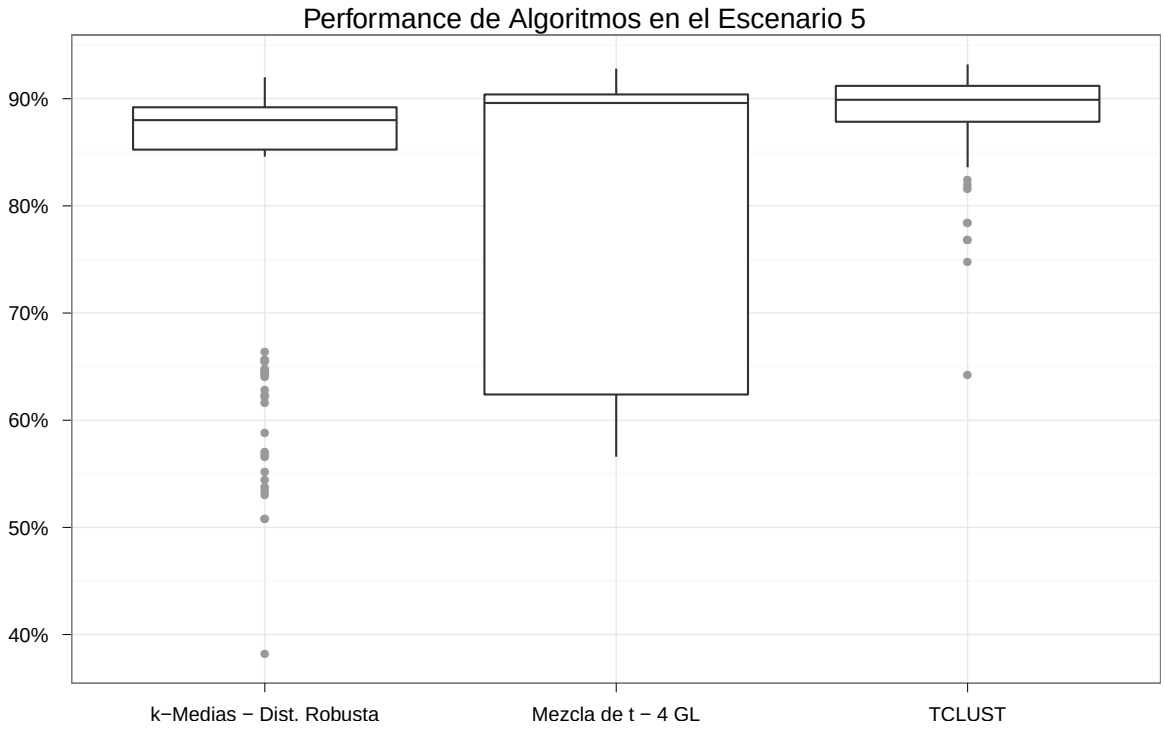

Figura 6.9: Diagramas de cajas del porcentaje de observaciones bien clasificadas por cada algoritmo en el Escenario 5

Repetidas las simulaciones de la misma manera que en los anteriores escenarios es posible realizar el diagrama de caja, ver figura 6.9.

Como se puede apreciar este tipo de tipologías de outliers, sesgadas hacia algún sector o locales, afectan de manera significativa la eficiencia del algoritmo de mezclas de t, presentando una alta performance los algoritmo con poda de datos (TCLUST y la variante robusta del k-Medias), teniendo este último una gran variabilidad que lo hace muy inestable.

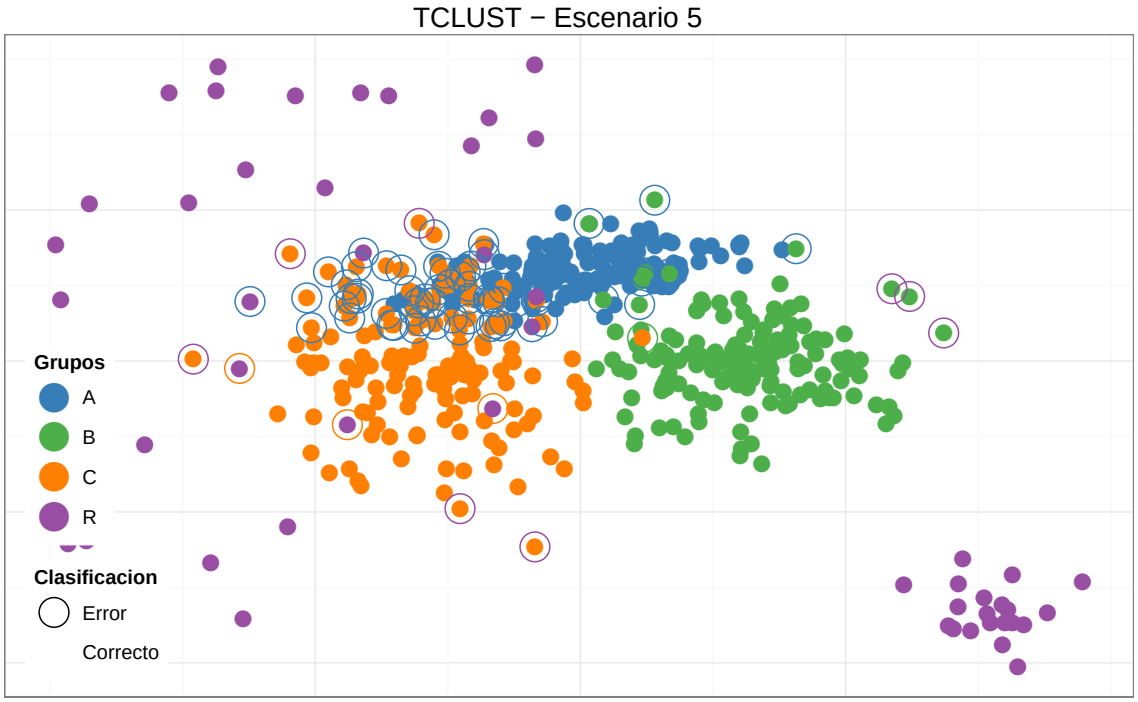

Figura 6.10: Clasificación mediante TCLUST en el escenario 5

6.2. Conclusiones

Como se pudo comprobar vía simulación, la performance de los algoritmos tratados presentan una alta dependencia respecto a la distribución de los datos, así como también a la tipología de los outliers.

Cuando la contaminación se presenta en un bajo porcentaje, en forma global y uniforme, la mezcla de t parece ser la mejor opción, puesto que las colas pesadas soportan estos valores y no es necesario la poda.

Cuando la contaminación es elevada o sesgada respecto a la distribución de los datos, si los grupos son esféricos y de igual tamaño la variante robusta de k-Medias es el que clasifica mejor.

Sin embargo, si los grupos presentan formas elípticas y el ruido toma formas menos previsibles se observa como el TCLUST supera a los algoritmos anteriores, teniendo una elevada precisión con respecto a los datos bien clasificados en cada simulación.

Capítulo 7

Aplicación a Datos Reales

Con el objetivo de identificar observaciones atípicas, se aplicaron las técnicas a un conjunto de datos provisto por una inmobiliaria de Montevideo.

El mismo cuenta con 190 observaciones que representan propiedades que estuvieron o están a la venta en la inmobiliaria en el último año, de las cuales se tienen las siguientes variables:

Variable	*Descripción*
`id`	Identificador numérico de la Propiedad
`zona`	Nombre de la zona en que se encuentra la Propiedad
`precio`	Precio en dólares de la Propiedad
`m2const`	Metros cuadrados construidos en la Propiedad
`m2terreno`	Metros cuadrados de la Propiedad

Cuadro 7.1: Conjunto de datos de propiedades de Montevideo y Canelones

De acuerdo a lo conversado con la gerencia de la inmobiliaria, ésta percibe 4 estratos dentro del conjunto de datos. El objetivo del análisis será delimitar esos estratos e identificar datos atípicos con el fin de evaluar la política de precios empleada.

7.1. Descripción de los Datos

El conjunto de datos contiene observaciones de 7 zonas:

		Precio (Miles)		Metros		Metros Const.	
Zona	Obs.	Media	Desvío	Media	Desvío	Media	Desvío
Barra de Carrasco	10	119.10	44.43	660.20	186.87	157.90	46.21
Carrasco	51	359.69	243.65	811.69	671.11	299.45	266.45
Carrasco Norte	24	153.83	96.76	878.08	708.77	202.38	130.03
Malvín	12	204.58	70.40	559.67	264.89	267.75	120.39
Parque Miramar	46	197.46	281.78	778.35	875.16	198.93	104.37
Punta Gorda	42	256.79	145.40	620.26	289.58	260.36	131.36
Shangrilá	5	94.00	54.70	903.40	290.19	141.60	36.54

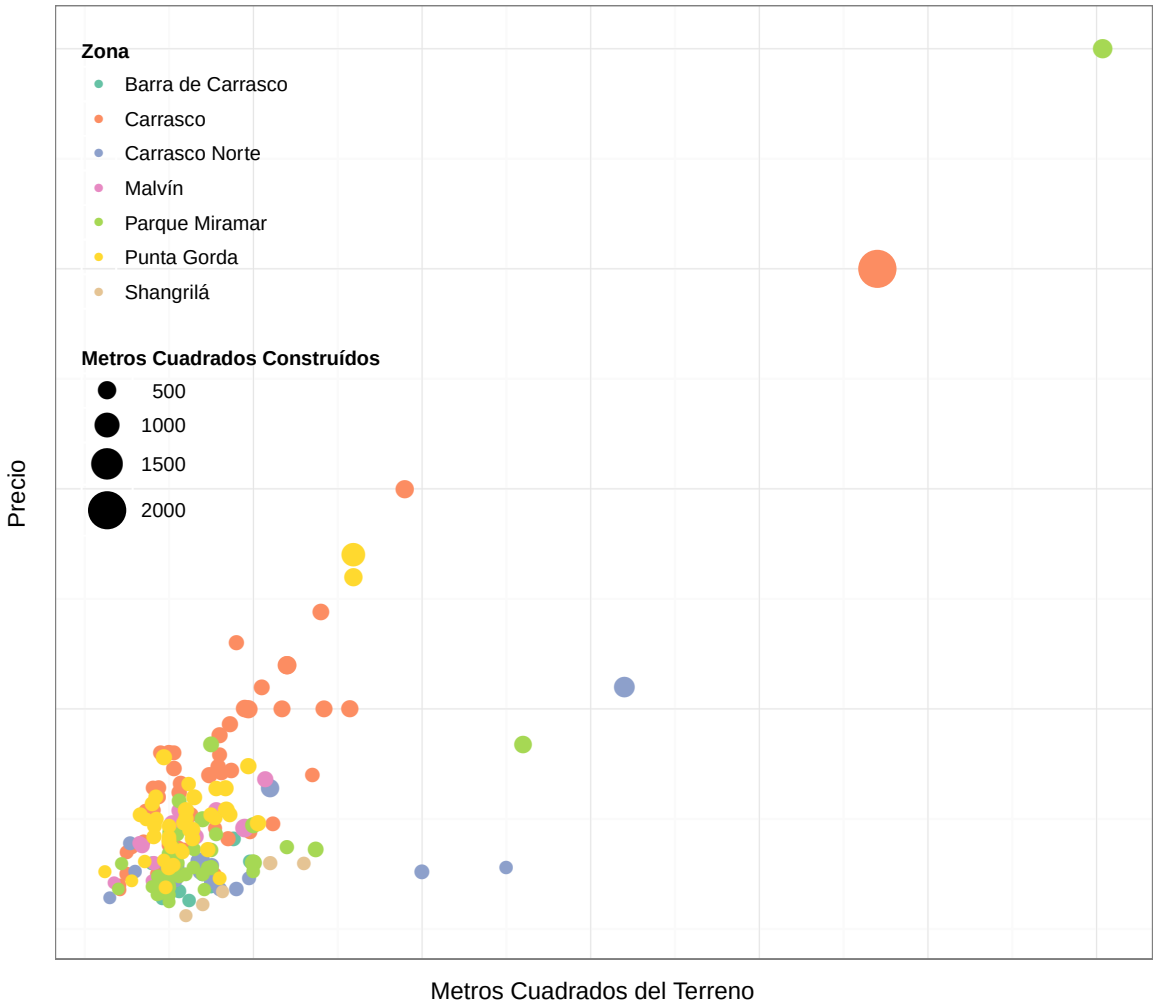

Figura 7.1: Conjunto de Datos

Se puede apreciar una correlación lineal positiva entre los metros de la propiedad y su precio en todas las zonas. Dicha relación lineal varía su pendiente de acuerdo a la zona. También se verifica dicha relación entre precio y metros cuadrados construídos.

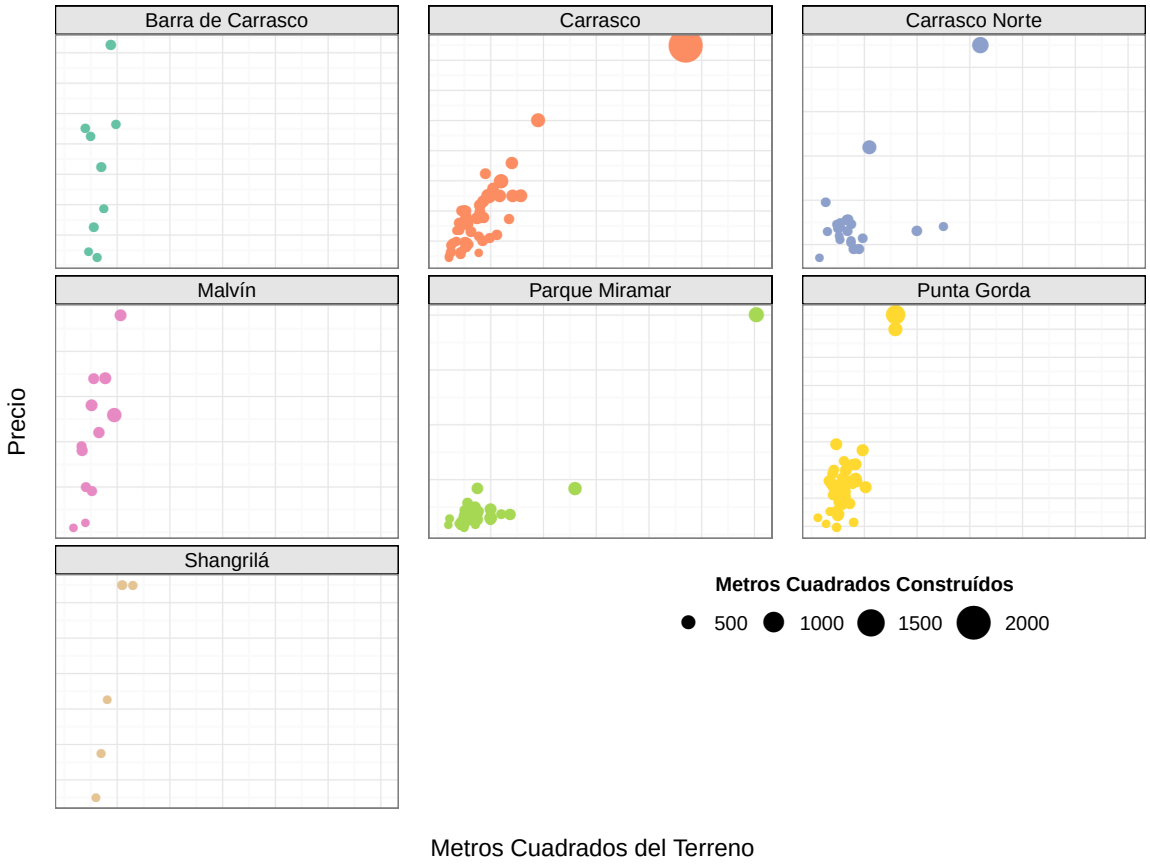

Figura 7.2: Metros Cuadrados del Terreno contra Precio, Tamaño por Metros Cuadrados Construidos. Escala de Precio variable.

Mediante una inspección visual primaria, podrían existir datos atípicos en las zonas de Carrasco, Carrasco Norte, Parque Miramar y Punta Gorda.

En las zonas de Barra de Carrasco y Malvín, las gráficas sugieren que el tamaño del terreno y sus metros cuadrados construídos no contribuyen al precio de la propiedad. Dicha relación (independencia) parece ser más débil en Shangrilá.

En Punta Gorda, y en menor medida en Carrasco Norte, los metros construidos parecen tener un gran impacto en el precio de la propiedad.

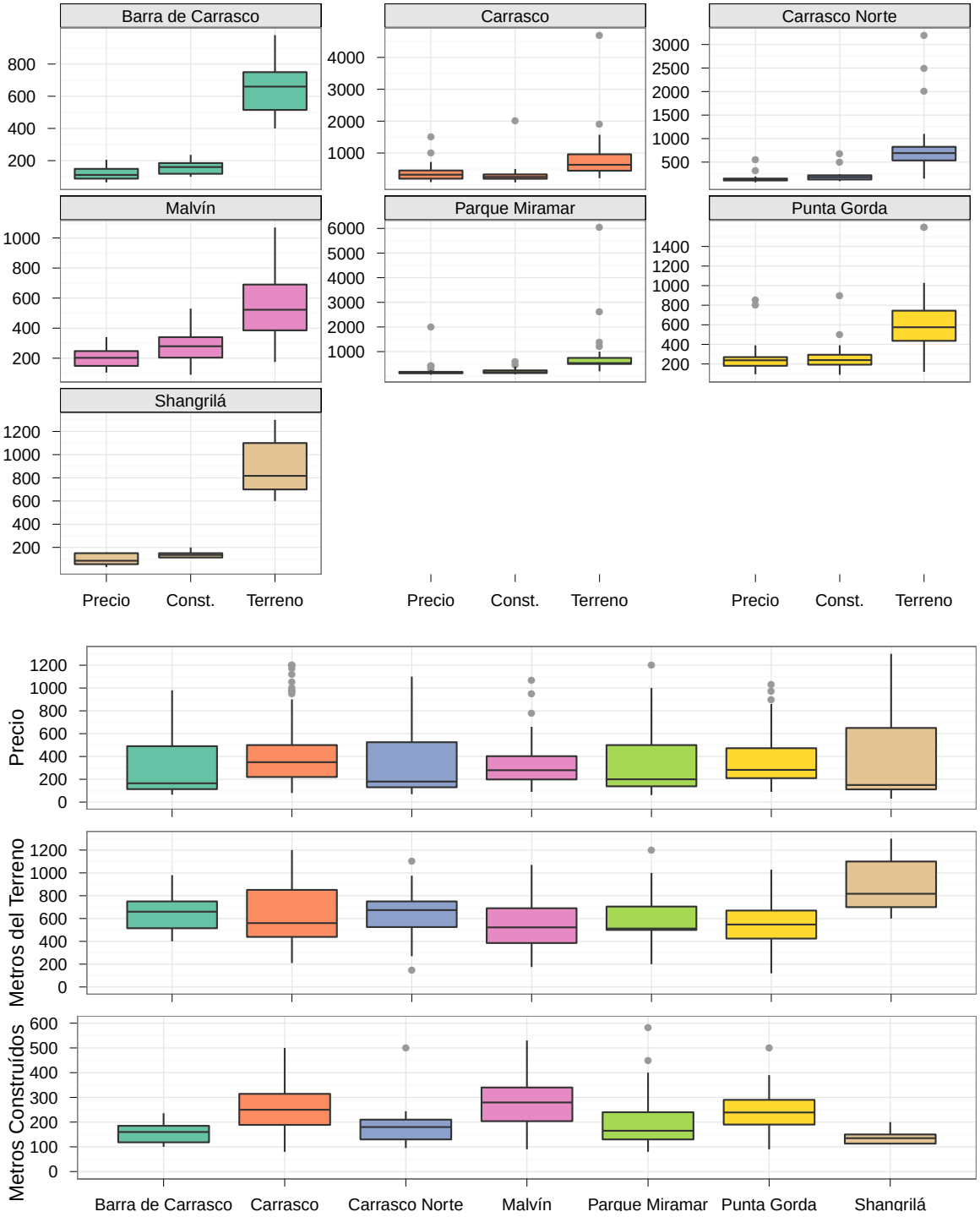

Figura 7.3: Diagrama de Cajas por Zona

7.2. Algoritmo TCLUST

Se aplicó el algoritmo TCLUST al conjunto de datos para encontrar 4 grupos con una poda del 10 %.

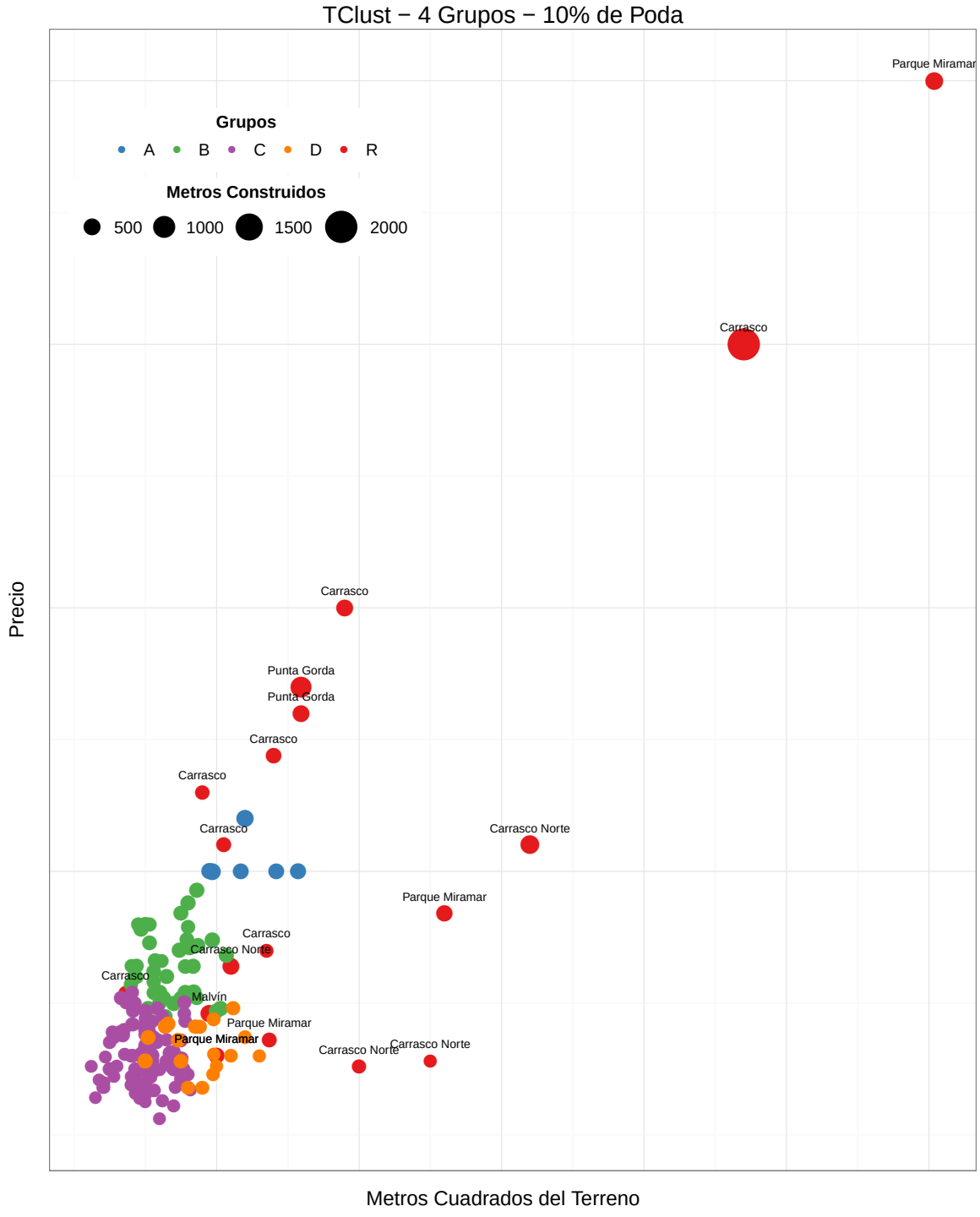

Figura 7.4: Grupos identificados por TCLUST

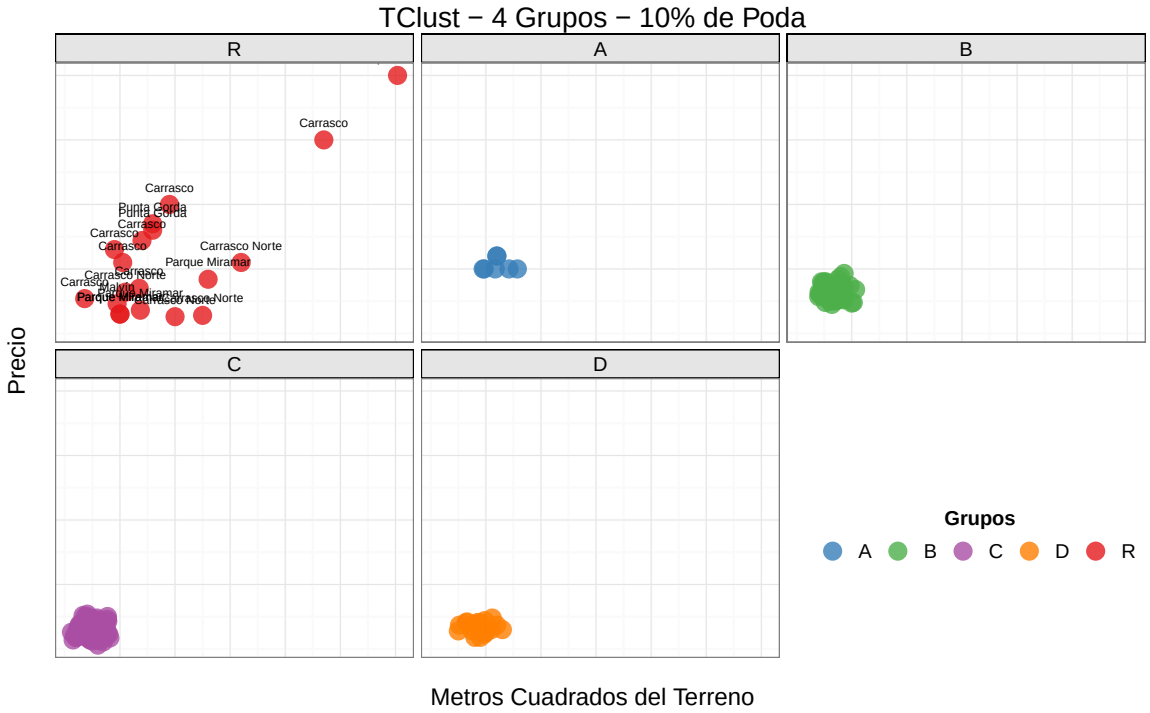

Figura 7.5: Grupos identificados por TCLUST - 2

		Precio (Miles)		Metros		Metros Const.	
Zona	Obs.	Media	Desvío	Media	Desvío	Media	Desvío
R	19	576.84	498.73	1926.53	1409.86	493.00	410.99
A	7	528.57	48.80	1211.86	224.22	452.86	47.16
B	40	321.12	61.92	678.27	183.74	297.32	44.89
C	106	146.43	51.36	509.97	160.14	160.92	48.01
D	18	166.28	44.25	882.50	224.10	234.89	55.29

Cuadro 7.2: Descriptiva de los grupos identificados por TCLUST

Como se puede apreciar en el Cuadro 7.2, el grupo `A` se encuentra caracterizado por pocas propiedades de más alto valor, con terrenos y edificaciones más grandes.

El grupo `B` se caracteriza por propiedades del 60 % del valor - en promedio - que las del grupo `A`, con terrenos de la mitad de tamaño pero la mitad del terreno se encuentra edificado en vez de un cuarto del primer grupo.

El grupo `C` es el más numeroso y está compuesto por las propiedades más pequeñas, de menor valor y con menos metros cuadrados construidos.

El grupo `D` se trata de 18 propiedades cuyo precio promedio es un 15 % mayor a las del grupo `C`, pero el tamaño del terreno es un 70 % superior y los metros construidos un 40 % mayor.

Figura 7.6: Grupos identificados por TCLUST - Zonas por Grupos

En cuanto a la composición de zonas de los grupos (Figura 7.6), el grupo A - propiedades de mayor valor - está compuesto únicamente por propiedades en Carrasco.

El grupo `B` está conformado en su mayoría por propiedades en Punta Gorda y Carrasco, con algunas de Parque Miramar y Malvín.

El grupo `C` - el más numeroso - tiene propiedades de todas las zonas, pero mayoritariamente de Parque Mirmar y Punta Gorda, Carrasco Norte y Carrasco en menor medida.

El grupo `D` contiene propiedades de todos las zonas pese a su pequeño tamaño.

Tanto Shangrilá como Carrasco Norte y Barra de Carrasco solamente tienen propiedades en los grupos `C` y `D` (sin tomar en cuenta los atípicos).

En cuanto a los datos atípicos, el grupo `R`, están integrados principalmente por propiedades en Carrasco, Parque Miramar y Carrasco Norte. Shangrilá y Barra de Carrasco no presentan atípicos detectados por TCLUST.

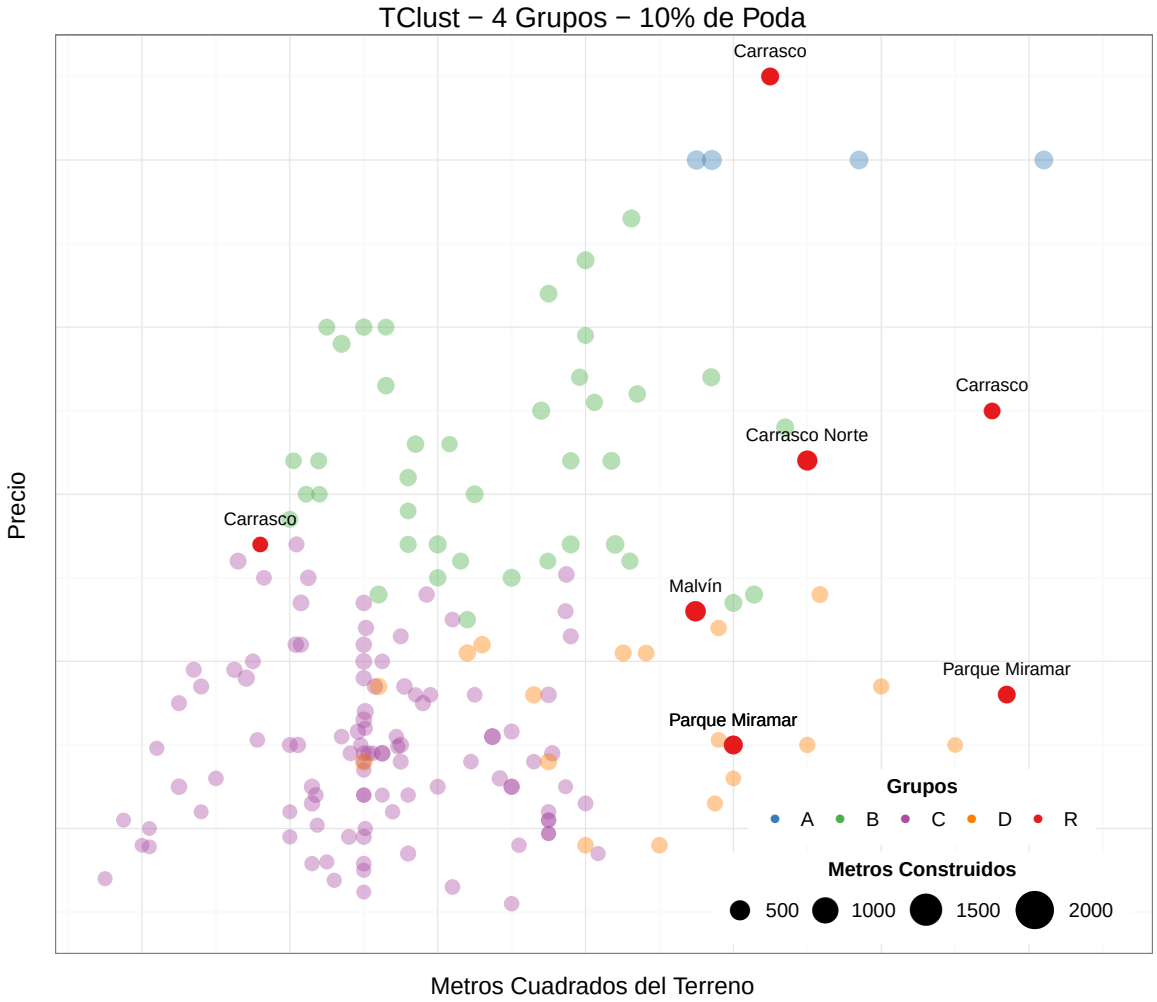

Figura 7.7: Datos atípicos "interiores" identificados por TCLUST

7.3. Algorítmo EMMIX

Se aplicó el algoritmo EMMIX al conjunto de datos para encontrar 4 grupos con una poda del 10 %.

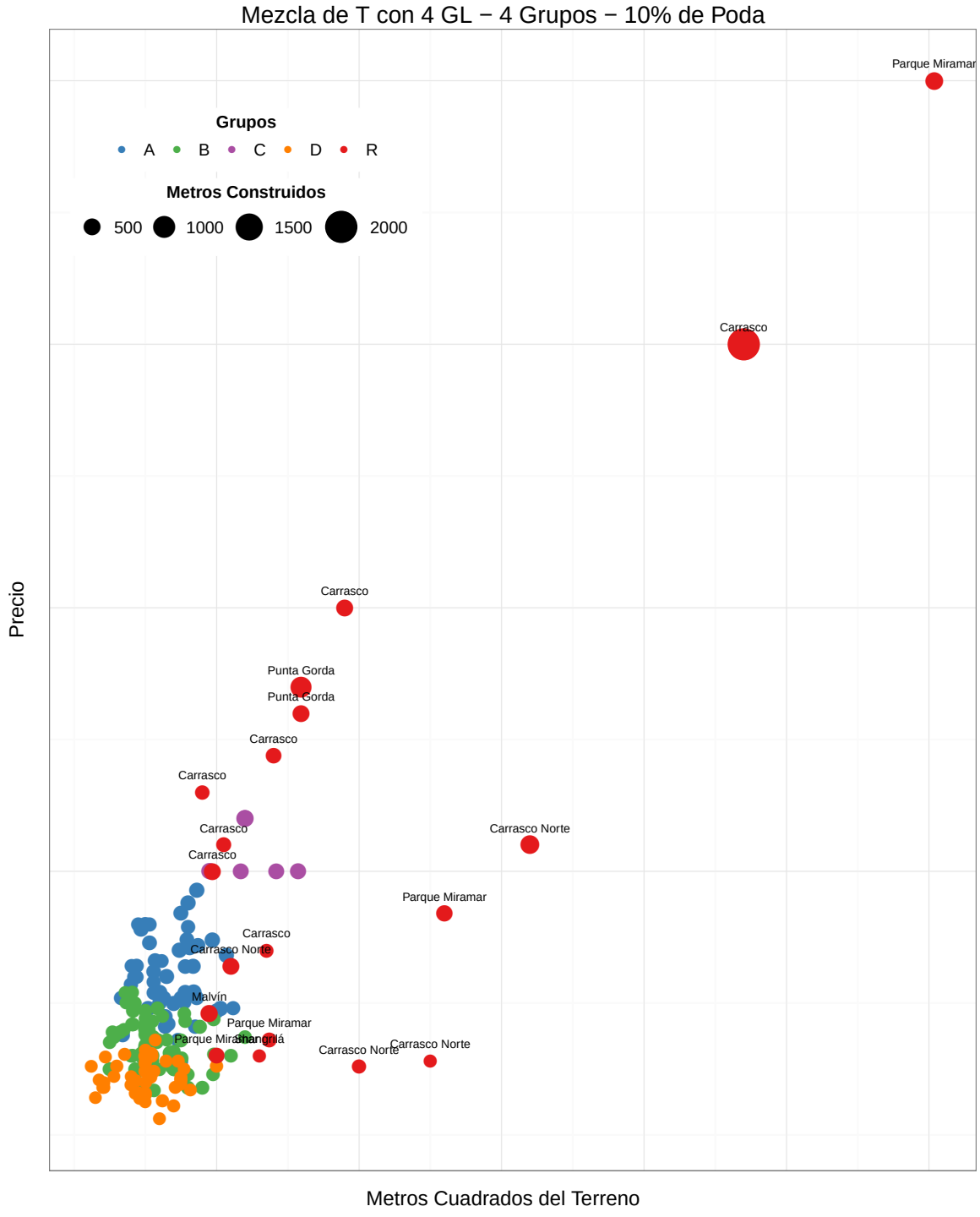

Figura 7.8: Grupos identificados por EMMIX

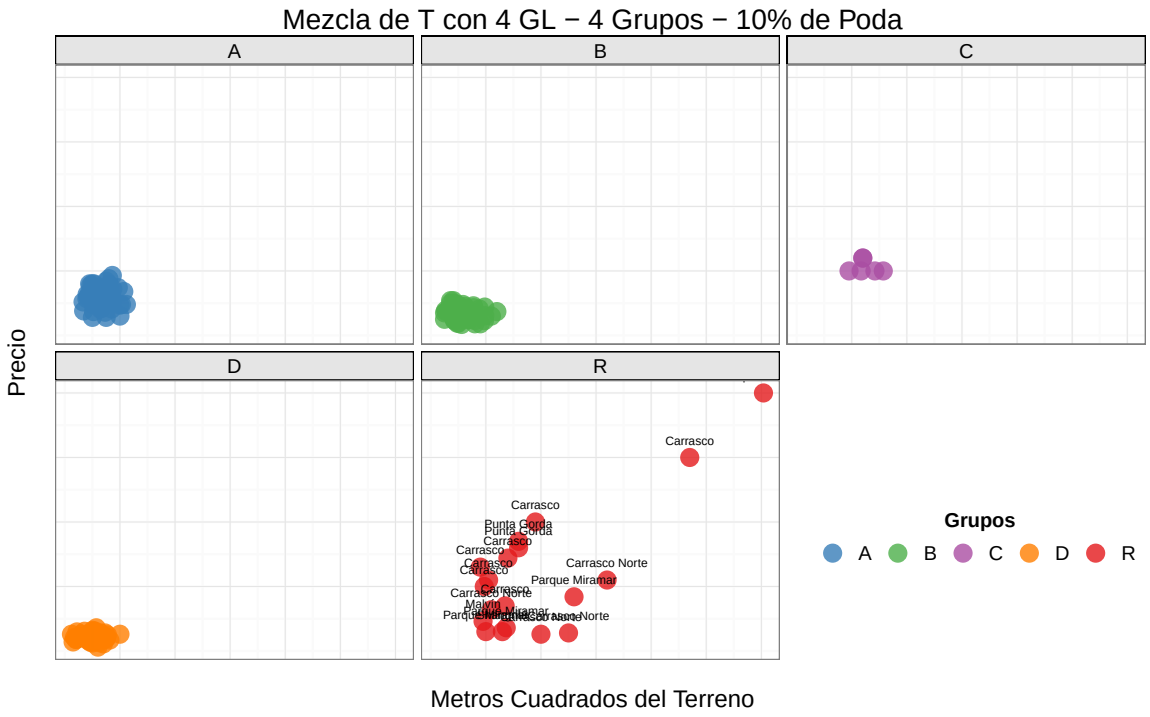

Figura 7.9: Grupos identificados por EMMIX - 2

		Precio (Miles)		Metros		Metros Const.	
Zona	Obs.	Media	Desvío	Media	Desvío	Media	Desvío
A	52	292.35	78.14	679.69	195.69	293.25	42.04
B	66	170.82	45.66	573.48	201.28	193.38	28.72
C	6	533.33	51.64	1252.00	216.31	445.00	46.37
D	47	109.66	31.64	509.34	196.82	116.09	19.02
R	19	588.95	493.64	1974.47	1368.87	504.05	407.84

Cuadro 7.3: Descriptiva de los grupos identificados por EMMIX

Como se puede apreciar en el Cuadro 7.3, el grupo `C` identificado por EMMIX es, salvo por una observación, igual al grupo `A` identificado por TCLUST. Son las propiedades de más alto valor, con terrenos y edificaciones más grandes.

El grupo `A` es análogo al grupo `B` identificado por TCLUST: propiedades del 60 % del valor - en promedio - que las del grupo `A`, con terrenos de la mitad de tamaño pero la mitad del terreno se encuentra edificado.

El grupo `B` es el más numeroso, análogo al grupo `D` de TCLUST, con propiedades un 70 % más caras que el grupo `D` (del EMMIX), terrenos un poco más grandes pero el triple de metros cuadrados construidos. Presenta el valor del metro cuadrado construido (sin considerar el tamaño del terreno) más bajo de los grupos, apoxidamente USD 880.

El grupo D se trata de las propiedades con menor valor, tamaño del terreno y metros construidos. Tiene el mismo valor del metro cuadrado construido (sin considerar el tamaño del terreno) que el grupo A, aproximadamente USD 1000.

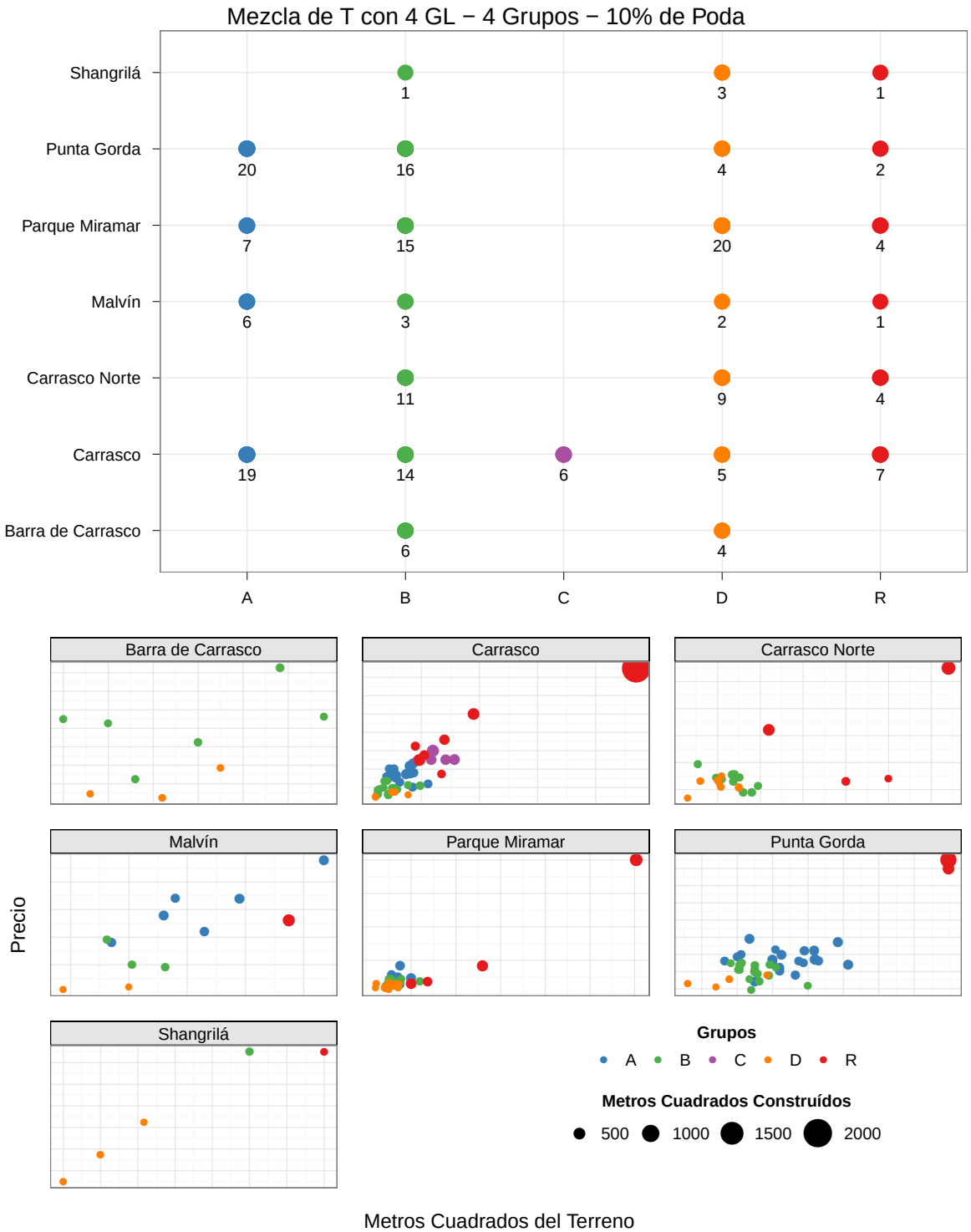

Figura 7.10: Grupos identificados por EMMIX - Zonas por Grupos

En cuanto a la composición de zonas de los grupos (Figura 7.10), si bien las proporciones varían, la composición es la misma que la de los grupos identificados por TCLUST.

Lo mismo es para los datos atípicos, únicamente introduciendo a Shangrilá con una observación y detectando otra observación de Carrasco.

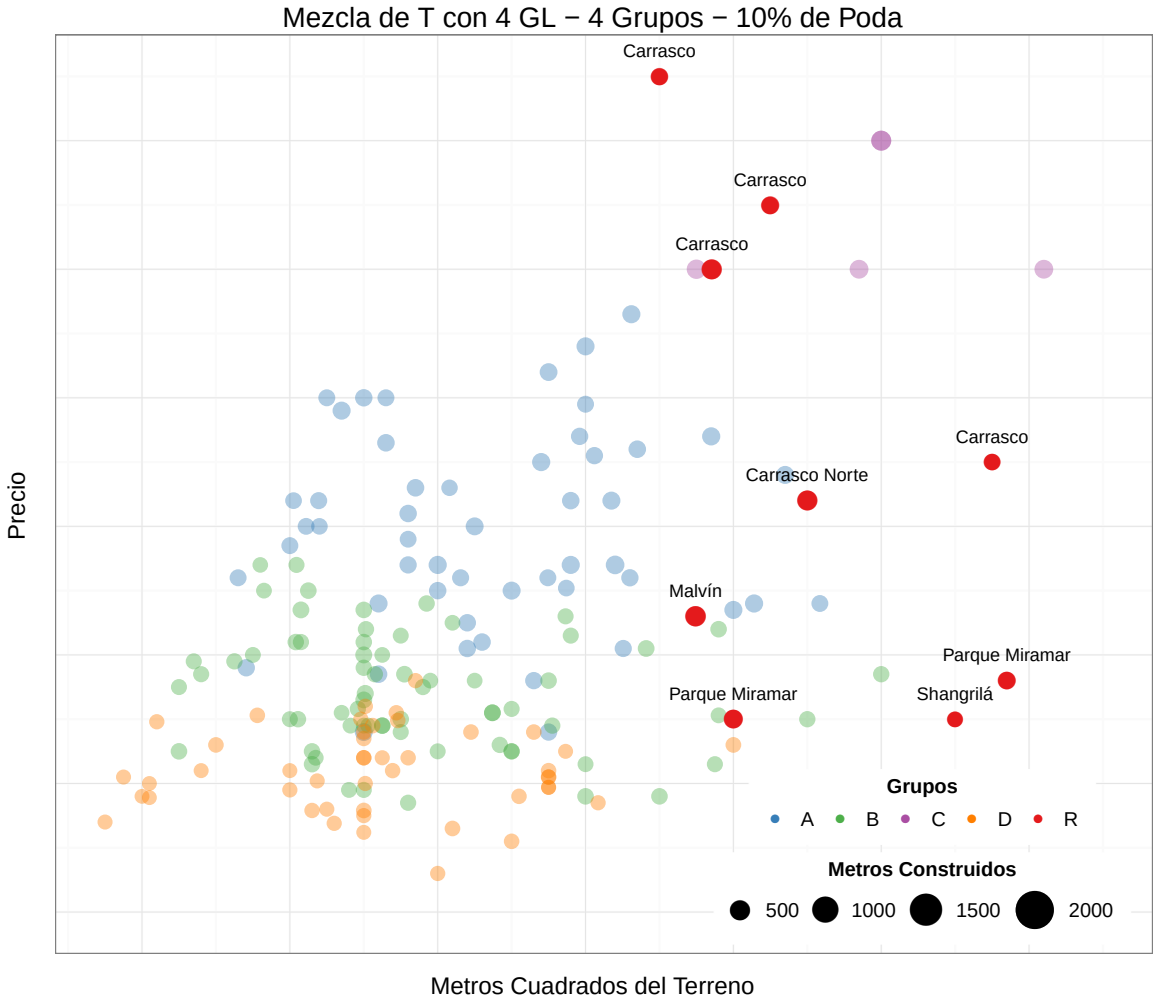

Figura 7.11: Datos atípicos "interiores" identificados por EMMIX

7.4. Algorítmo de K-Means Robusto

Se aplicó el algoritmo K-Means Robusto al conjunto de datos para encontrar 4 grupos con una poda del 10 %.

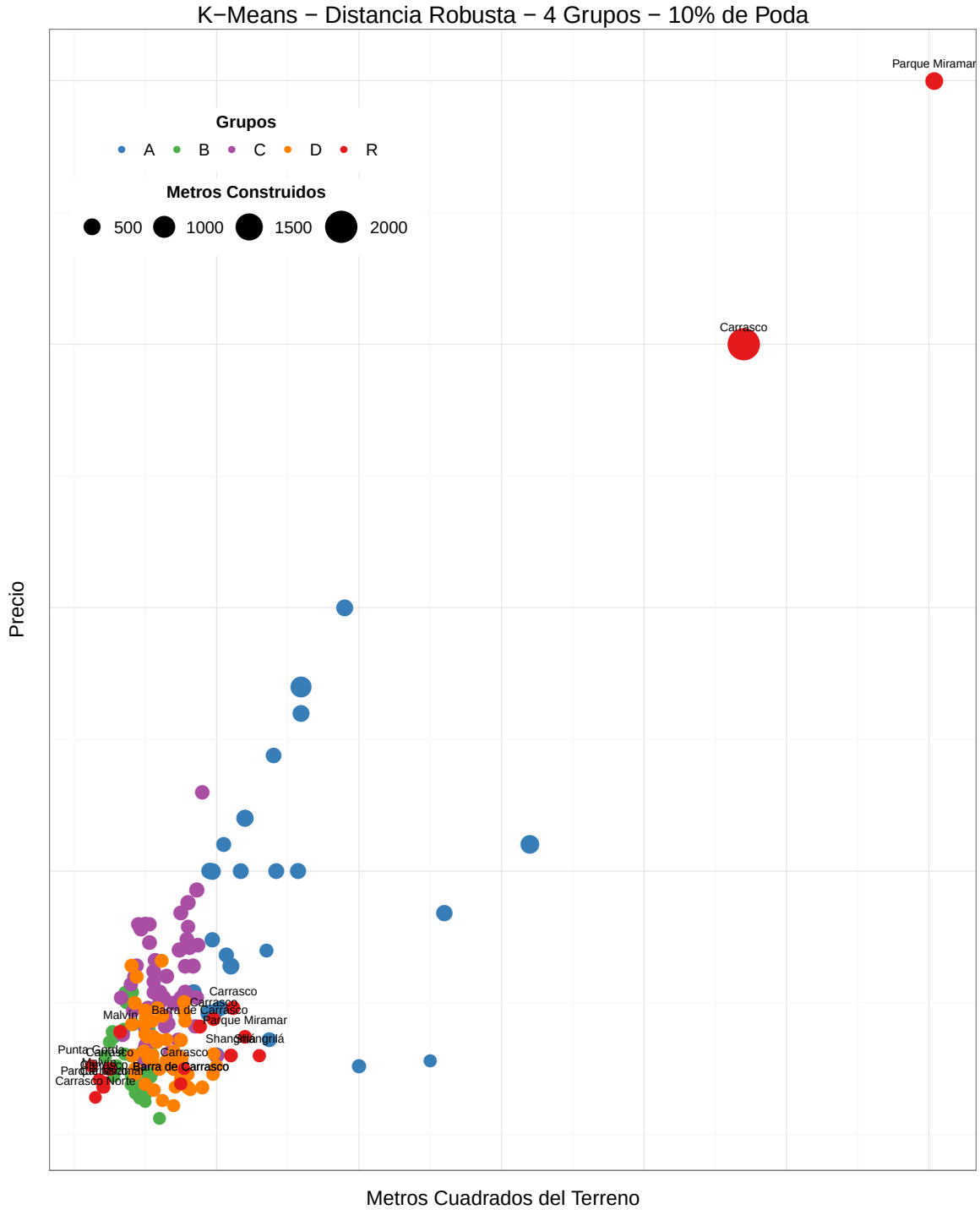

Figura 7.12: Grupos identificados por K-Means Robusto

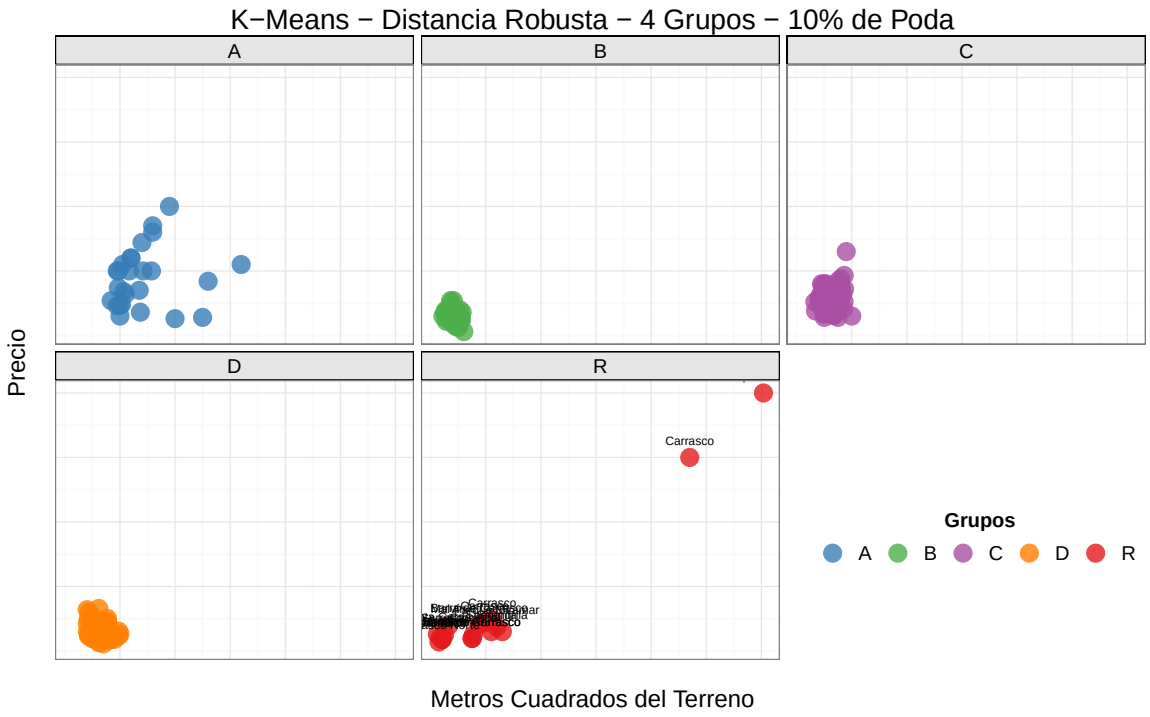

Figura 7.13: Grupos identificados por K-Means Robusto - 2

		Precio (Miles)		Metros		Metros Const.	
Zona	Obs.	Media	Desvío	Media	Desvío	Media	Desvío
A	26	444.04	229.20	1422.96	586.35	423.69	147.95
B	44	139.45	52.33	446.05	93.19	134.89	27.30
C	49	286.43	101.75	637.73	161.91	287.24	35.67
D	52	162.13	63.17	635.73	153.39	184.71	33.15
R	19	309.11	516.87	1117.37	1567.70	267.58	435.38

Cuadro 7.4: Descriptiva de los grupos identificados por K-Means Robusto

Los grupos identificados por K-Means Robusto son los más esféricos en comparación con TCLUST y EMMIX.

El grupo `A` es análogo al grupo `A` identificado por TCLUST y `B` por EMMIX, las propiedades de mayor valor. Sin embargo, el mismo engloba la mayoría de los datos declarados como atípicos por los otros algoritmos.

Los grupos `B`, el de segundo menor precio promedio, presenta la misma relación con el grupo `D` (propiedades de menor valor), precio levemente mayor pero sustancial incremento en el tamaño del terreno y sus metros construidos.

El grupo `C` se trata de propiedades con un precio 70 % mayor que las del grupo `D`, mismo tamaño del terreno pero con un 50 % más de metros construidos.

Figura 7.14: Grupos identificados por K-Means Robusto - Zonas por Grupos

En cuanto a la composición de zonas de los grupos, la estructura es más difusa.

A excepción de Shangrilá y Barra de Carrasco, todos los grupos tienen propiedades de todas las zonas, inclusive los detectados como atípicos.

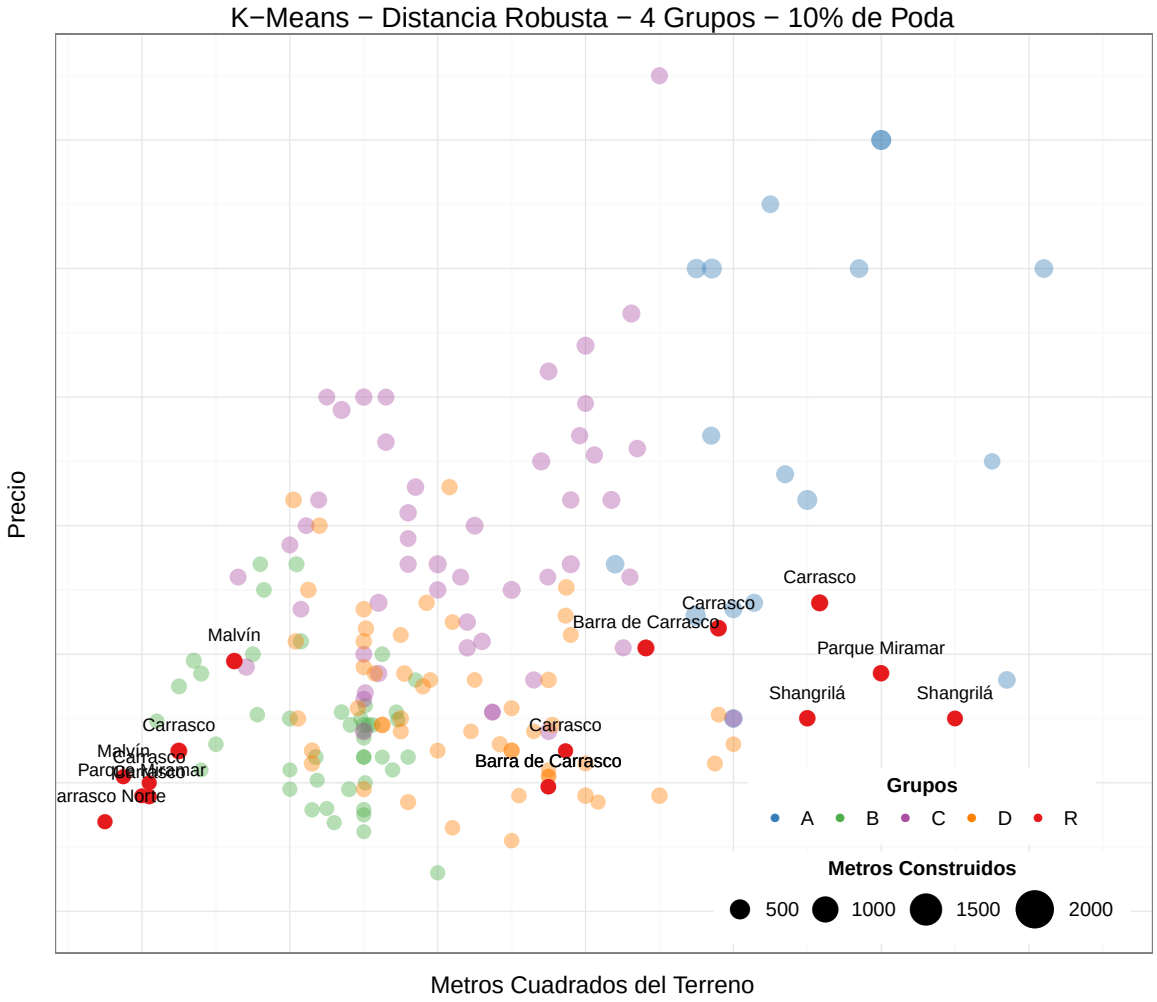

Figura 7.15: Datos atípicos "interiores" identificados por K-Means Robusto

7.5. Conclusiones

7.5.1. Grupos Detectados

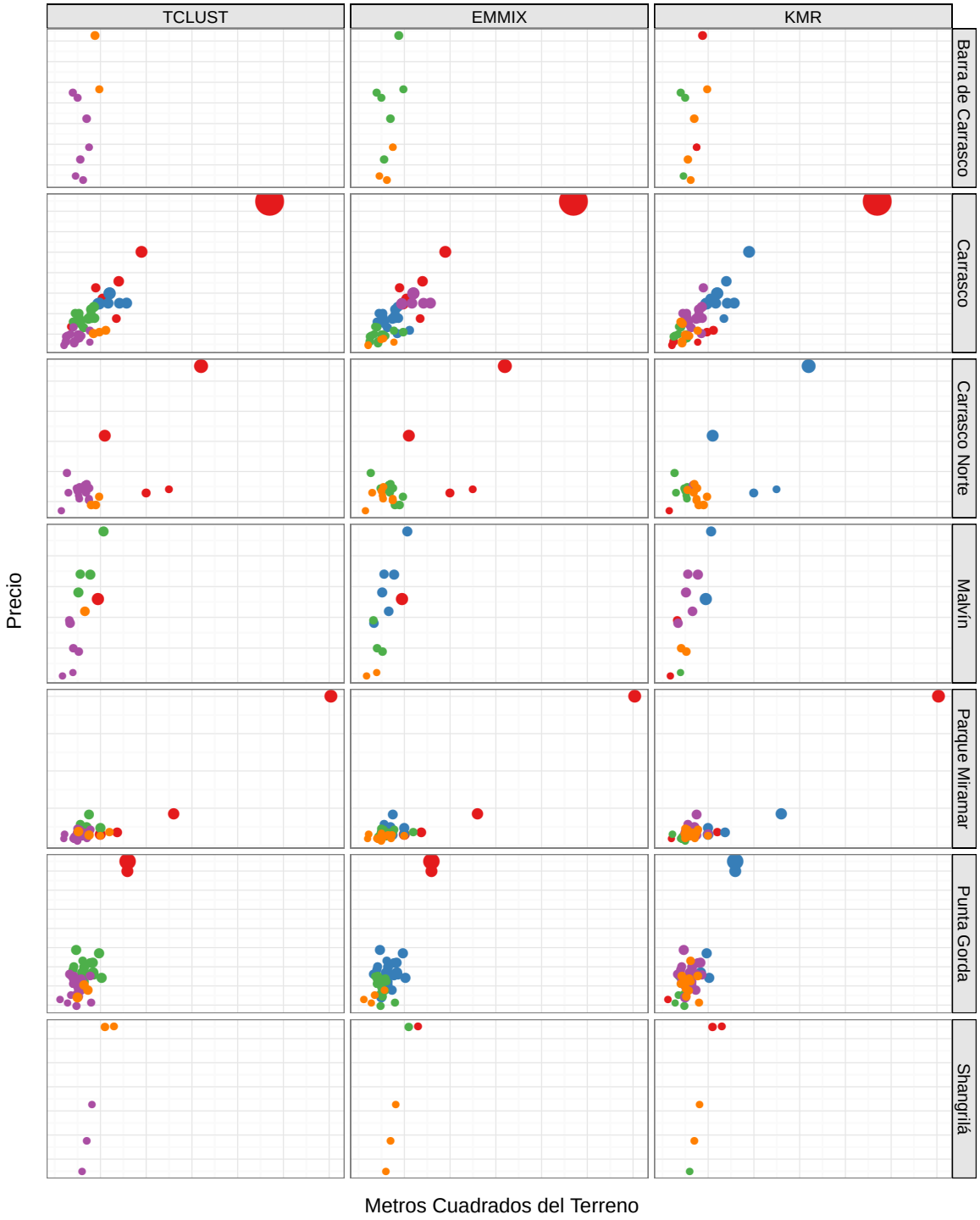

Figura 7.16: Comparación de Grupos por Zonas - Colores no alineados excepto atípicos

Tanto TCLUST como EMMIX detectan grupos muy similares, tanto en composición de zonas como características de las propiedades. Esta estructura es detectada por K-Means Robusto pero de forma más difusa, lo que hace suponer que dicha estructura es razonable para el problema.

Los grupos identificados por TCLUST son los más elípticos y de distinto tamaño, mientras que los de K-Means Robusto son los más esféricos y de tamaño similar.

Esto se debe al alto valor utilizado para la restricción del cociente de los valores propios de las matrices de varianzas (en este caso 90). Dicho parámetro permite identificar grupos de estructura más heterogénea entre sí que el algoritmo EMMIX.

Consultando con la gerencia de la inmobiliaria, la misma confirma la estructura detectada (características y relaciones entre grupos) a partir de su conocimiento del campo.

La misma se notó sorprendida por los grupos `B` y `D` identificados por TCLUST, ya que los mismos captan un conjunto de propiedades que consideran muy interesantes.

7.5.2. Observaciones Atípicas Detectadas

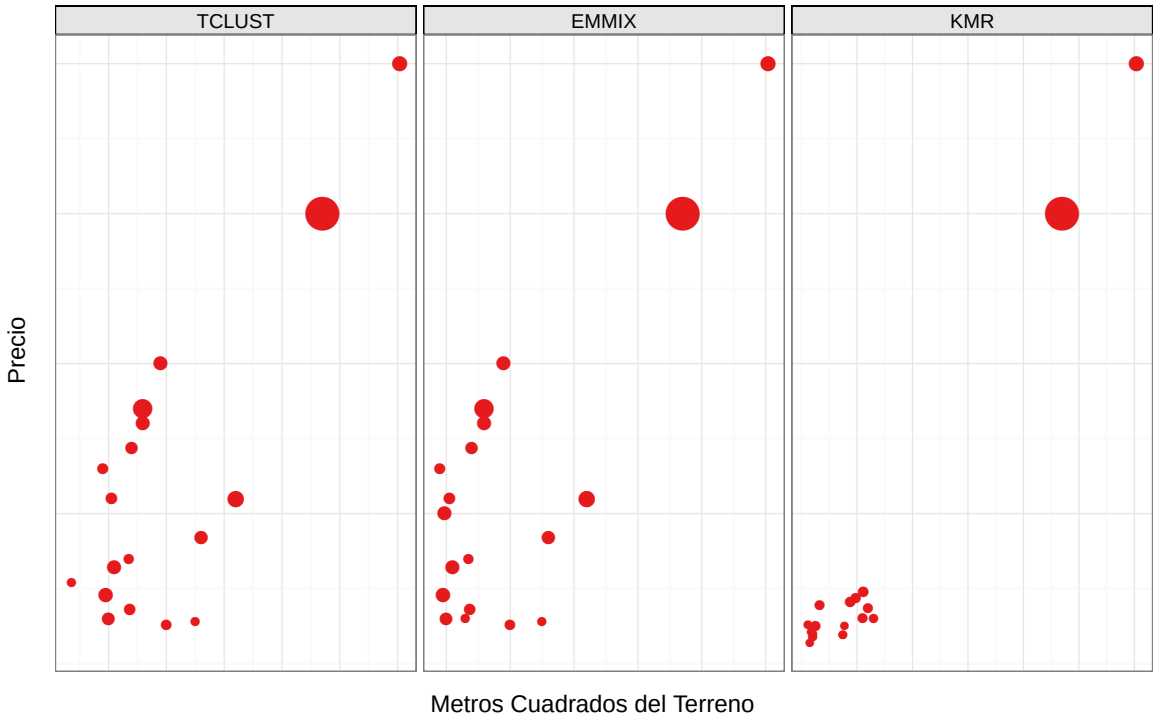

Figura 7.17: Comparación de Atípicos por Técnica

Los outliers detectados por TCLUST y EMMIX son muy similares, mientras que K-Means Robusto detecta en las "colas" de los datos porque dedica un grupo a la

mayoría de los detectados por las otras técnicas.

Dicho grupo, "atípicos exteriores", se trata de propiedades que por su ubicación, calidad de la construcción y tamaño, su precio es fijado más arbitrariamente ya que son "menos comparables" con las restantes.

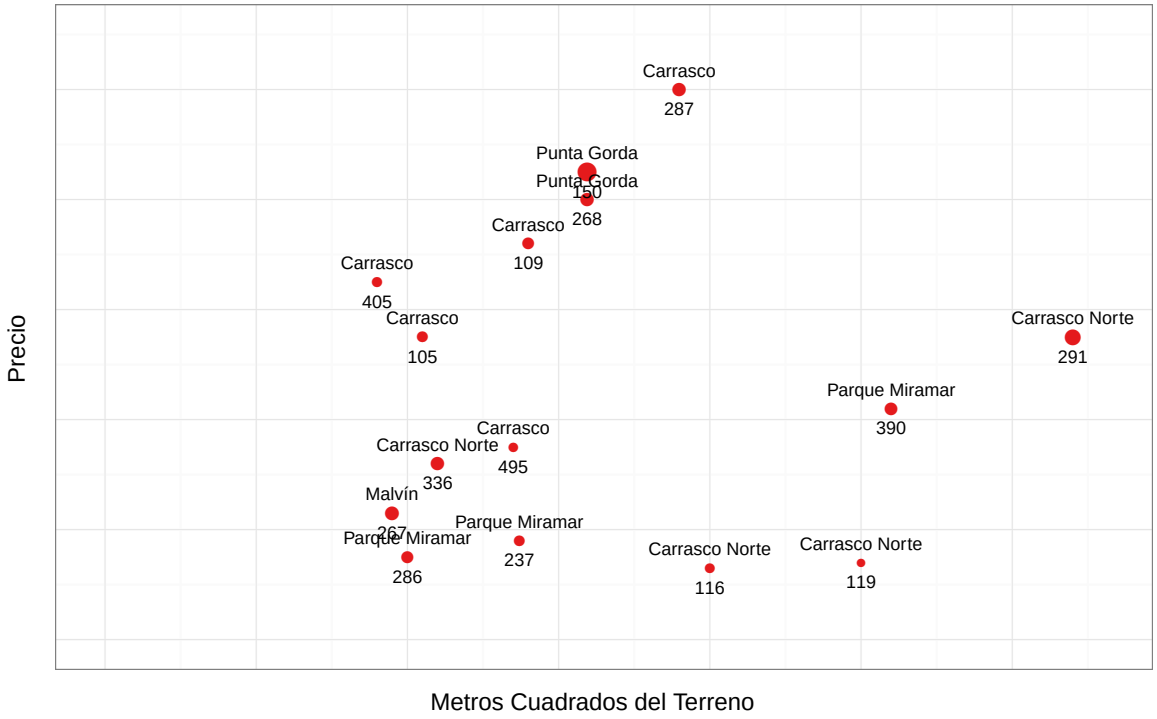

Figura 7.18: Atípicos "interiores" detectados por al menos dos técnicas

Por lo tanto, es de mayor interés estudiar los atípicos "interiores", es decir, aquellos que no surgen de la inspección visual ya que se encuentran con más "profundidad" en la nube de puntos.

En la Figura 7.18 se visualizan los atípicos "interiores" detectados por al menos dos técnicas, donde se pueden apreciar 3 grupos dentro de los mismos.

El primer grupo, integrado por las propiedades 116, 119, 390 y 291, se tratan de propiedades de muy bajo valor para el tamaño de su terreno. El mismo puede derivarse de la inspección visual y se tratan de propiedades con ubicaciones poco deseadas y construcciones que necesitan importantes reparaciones.

Un segundo grupo está conformado por las observaciones 405 y 105 en la zona de Carrasco (las observaciones 109, 268, 150 y 287 se consideran "exteriores"). Dichas observaciones son un "punto medio" entre las "exteriores" y las "comparables".

En tercer grupo es el de mayor interés, ya que son los atípicos con mayor profundidad en la nube.

Con la excepción de la propiedad 237 de Parque Miramar, todas ellas han sido señaladas como excelentes ofertas. En especial, la propiedad 267 en Malvín, lamentablemente ya vendida.

ID	Zona	Precio (Miles)	Metros Const.	Metros del Terreno
105	Carrasco	550	319	1050
109	Carrasco	720	380	1400
110	Carrasco	1500	2000	4700
287	Carrasco	1000	500	1899
405	Carrasco	650	270	900
495	Carrasco	350	221	1350
116	Carrasco Norte	130	244	2000
119	Carrasco Norte	140	140	2500
291	Carrasco Norte	550	680	3200
336	Carrasco Norte	320	500	1100
267	Malvín	230	530	949
237	Parque Miramar	180	310	1370
279	Parque Miramar	2000	583	6038
286	Parque Miramar	150	400	1000
390	Parque Miramar	420	450	2600
150	Punta Gorda	850	900	1594
268	Punta Gorda	800	500	1594

Cuadro 7.5: Detalle de Atípicos detectados por al menos dos técnicas

Capítulo 8

Conclusiones finales y trabajos futuros

En el presente trabajó se intentó introducir el concepto de Robustez y analizar algoritmos para clustering que lo empleen.

A partir de la elección de un modelo, se define para el problema lo que se considera "típico" y "atípico" a través de una forma de captarlo.

Como no hay un algoritmo mejor que otro *per se*, lo mismo sucede con la robustez: los algortimos son robustos en cierto sentido bajo determinado modelo. Esto se intentó demostrar mediante los estudios de simulación.

A partir de técnicas robustas se puede detectar mejor lo "típico", como se mostró en el capítulo de aplicación a datos reales. A su vez, los datos "atípicos" tienen un interés en sí mismos, como la identificación de propiedades que constituyen buenas ofertas.

El tamaño de los conjuntos de datos ha crecido a órdenes donde el suponer que fueron generados por un único proceso - y que a su vez es conocido perfectamente - es, cuando menos, extremadamente poco realista.

La robustez es entonces, prácticamente, una necesidad en la actualidad.

Una posible investigación a futuro - que no fue tratada en el trabajo monográfico por motivos de tiempo y extensión - es la de obtener clusters robustos mediante el uso de cópulas.

Los métodos desarrollados en este trabajo consideran la matriz de dispersión como la "fuerza motriz" del análisis. Entonces, se asume que toda la información acerca de la dependencia entre los componentes del vector aleatorio está contenida en la matriz de covarianzas.

Krzysztof Jajuga (2005) [19] propone un enfoque alternativo a los métodos clásicos. En vez de analizar conjuntamente los parámetros de escala y dependencia, dados

en la matriz de varianzas y covarianzas, el análisis se realiza separadamente para los parámetros de escala (a través del análisis univariado), y para los parámetros de dependencia.

El enfoque está basado en el llamado *análisis de cópulas*. Este camino es retomado por Francesca Di Lascio en su tesis doctoral en el 2008 [22], implementando el algoritmo *CoClust*.

La importancia del análisis de cópulas es la de permitir levantar el supuesto de distribuciones elípticas, supuesto necesario de los métodos anteriores.

El objetivo consistiría en construir métodos robustos de clustering, modelando a través de cópulas y siendo poco sensible a fenómenos de contaminación.

Se podría, a través del trabajo de Mendes, Melo y Nelsen (2007) [27], crear un nuevo algoritmo de clustering basado en cópulas que sea estable frente a perturbaciones en el modelo[1].

[1] Al cual llamaremos `RobCoClust`.

Apéndice A

Distribuciones Esféricas y Elípticas

La estadística multivariada clásica está basada, en general, en el supuesto que los datos provienen de una distribucion Normal, pero éste, en la mayoría de los problemas reales no es razonable.

Una primera idea es intentar extender la familia de las distribuciones normales a otra familia que permita mayor versatilidad en el ajuste del modelo y que simultáneamente conserve ciertas propiedades "buenas" de las variables normales.

En este sentido apunta la familia de distribuciones elípticas.

Se puede considerar la Normal Multivariante dentro de una familia más extensa, llamada familia de distribuciones elípticas. Esta familia de distribuciones puede ser generada a través de otra familia más elemental, la familia de las distribuciones esféricas.

La clase de distribuciones esféricas frente a las distribuciones elípticas juega el mismo papel que lo hace la $N_p(0, I)$ frente a la $N_p(\mu, \Sigma)$.

A.1. Definiciones y propiedades

Se comienza por optar por una de las posibles definiciones de distribuciones esféricas expuesta por Tkollo y von Rosen (2005)

Definición 5 (Distribución esférica) *Un vector $\boldsymbol{x}_d$ tiene una distribución esférica si su distribución es invariante bajo transformaciones ortogonales. Es decir, para toda matriz ortogonal Γ ($\Gamma.\Gamma' = \Gamma'.\Gamma = I$), $\boldsymbol{x}_d$ y $\Gamma'\boldsymbol{x}_d$ tienen la misma distribución.*

En estos casos la función de densidad debe depender del argumento $\mathbf{x}$ a través de $\mathbf{x}'\mathbf{x}$. Algunos ejemplos de éstas son:

- La distribución Normal Isótropa $N(0, \sigma^2.I_p)$.

- Mixtura de normales del tipo $N(0, \sigma_i^2.I_p)$.
- La distribución t multivariada con n grados de libertad, cuya densidad es

$$f(x) = \frac{\Gamma\left(\frac{1}{2}(n+p)\right)}{\Gamma\left(\frac{1}{2}n\right) n\pi^{p/2}} \left(1 + \frac{1}{n}x'.x\right)^{-\frac{n+p}{2}} .$$

Existen otras caracterizaciones equivalentes a la definición para distribuciones esféricas.

Teorema 12 (Caracterización a través de la función característica) *Sea x es un vector p-dimensional con distribución esférica si y sólo si su función característica $\varphi_x(t)$ cumple una de las siguientes condiciones equivalentes:*

1. *$\varphi_x(\Gamma' t) = \varphi_x(t)$ para cualquier matriz ortogonal $\Gamma_{p\times p}$,*
2. *Existe una función $\phi(.)$, función escalar que cumple $\varphi_x(t) = \phi(t't)$.*

Teorema 13 (Caracterización a través de la distribución uniforme) *Sea x es un vector p-dimensional con distribución esférica si y sólo si admite una representación estocástica del tipo*

$$x \stackrel{d}{=} R.u$$

donde u tiene una distribución uniforme en la esfera unidad, R es una v.a real no negativa, con u y R independientes.

En general, las distribuciones esféricas corresponden a distribuciones de variables aleatorias no correlacionadas. Sin embargo, dentro de las distribuciones esféricas, la Normal Isótropa es la única distribución compuesta por variables aleatorias independientes (ver Lindskog (2000)).

Una familia más general donde se encuentran contenidas las distribuciones esféricas son las distribuciones elípticas.

Definición 6 (Distribuciones Elípticas) *Se dice que un vector $\boldsymbol{x}_d$ tiene una distribución elíptica con parámetros $\mu_{p\times 1}$ y $V_{p\times p}$ si cumple que*

$$x \stackrel{d}{=} \mu + Ay,$$

donde y tiene distribución esférica y $A_{p\times k}$ verifica $A.A' = V$ con $rang(V) = k$,

y denotaremos $x \sim E_p(\mu, V)$.

Al igual que en las distribuciones esféricas, se puede caracterizar las distribuciones elípticas a través de su función característica.

Teorema 14 (Función característica de las distribuciones elípticas) *Si $x \sim E_p(\mu, V)$ con $rang(V) = k$ entonces la función característica de x ($\varphi_x(t)$) es de la forma*

$$\varphi_x(t) = e^{it'\mu}\phi(t'Vt),$$

para alguna función ϕ

Algunos ejemplos de distribuciones elípticas son:

- La distribución Normal $N_p(\mu, \Sigma)$.
- Mixtura de Normales del tipo $N(\mu_i, \Sigma_i)$.
- La distribución t multivariada con n grados de libertad y parámetros de locación μ y escala Σ, $x \sim t_p(n, \mu, \Sigma)$

Teorema 15 (La distribución elíptica se conserva bajo transformaciones afines) *Sea $x \sim E_p(\mu, V)$, $B_{m\times p}$ y ν un m-vector, entonces:*

$$\nu + B.x \sim E_m(\nu + B\mu, BVB').$$

En varias de las técnicas de cluster utilizadas en la monografía se parte del supuesto que los datos provienen de distribuciones elípticas, por tanto se necesitan a prori técnicas que permitan validar de cierta forma dicho supuesto.

A.2. Pruebas para distribuciones elípticas

En esta sección se describen algunas pruebas usuales de bondad de ajuste para las distribuciones elípticas.

A.2.1. Método Gráfico

Una manera de contrastar si una muestra de n vectores aleatorios i.i.d. de dimensión d, $x_1, x_2, \ldots, x_n$ provienen de una distribución elíptica, Li-Fang-Zhu (1997) propone un método basado en los gráficos Q-Q el cuál se sustenta en el siguiente lema. Este método y otros también son desarrollados por Liang en su tesis de doctorado.

Lema 8 *Sea $T(x)$ un estadístico tal que, casi seguramente*

$$T(ax) = T(x) \forall a > 0.$$

Entonces $T(x)$ tiene la misma distribución para todo vector esférico $x \sim S_d^+(\psi)$

A partir de este lema en el trabajo de Li-Fang-Zhu se proponen utilizar 2 estadísticos para $x \sim N_d(0, I_d) \quad iid$ que satisfagan la condición anterior.

$$T_1(x_i) = \frac{d^{1/2}\bar{R}_i}{\sqrt{\frac{1}{d-1}\sum_{j=1}^{d}(R_{i,j} - \bar{R}_i)^2}}$$

$$T_2(x_i) = \frac{\sum_{j=1}^{k} R_{i,j}^2}{\sum_{j=1}^{d} R_{i,j}^2}$$

Siendo $\bar{R}_i = \frac{1}{d}\sum_{j=1}^{d} R_{i,j}$,donde:

$$T_1(x_i) \sim t(d-1) \qquad T_2(x_i) \sim Beta\left(k/2, \frac{d-k}{2}\right)$$

Se propone realizar el Q-Q plot de los cuantiles teóricos contra los prácticos de $T_1(x)$ y de $T_2(x)$.

Si bien esta prueba esta diseñada para distribuciones esféricas, se realiza una transformación x^* a los datos x antes de aplicar los respectivos estadísticos donde:

$$x_i^* = \hat{\Sigma}^{-1/2}(x_i - \hat{\hat{\mu}})$$

Observando ambos QQ plot se puede, de forma gráfica, discernir si los datos provienen o no de una distribución elíptica.

A.2.2. Métodos Numéricos

Al igual que en el método gráfico, se realiza una transformación de los datos, para poder realizar un test sobre datos esféricos.

Sean Y_i los datos transformados, se define $R_i = \|Y_i\|$ y $S_i = Y_i/\|Y_i\|$.

Bajo la hipótesis nula que S_i estaría uniformemente distribuida en la esfera unidad s^{d-1} y los vectores (R_i, S_i) formarían pares de realizaciones independientes.

Para fijar ideas se considera el caso bivariado. Sea $S_i = (cos\Theta_i, sen\Theta_i)'$ donde Θ_i distribuye uniforme en la esfera s^1 tomando valores entre $[0, 2\pi]$. Equivalentemente si se define $U_i = \Theta_i/2\pi$,bajo H_0 distribuye $U(0, 1)$.

Por tanto el test se reduce a una prueba de bondad de ajuste sobre los valores de U_i por un lado y por otro un test de asociación (para analizar la independencia). Este método se encuentra desarrollado con más detalle en Mcneil, Frey y Embrechts (2005).

Apéndice B

Algoritmos

B.1. El algoritmo EM y sus variantes

En este apéndice se muestra el funcionamiento así como aquellos teoremas importantes para la convergencia del algoritmo *EM*(Esperanza- Maximización), así como también, algunas variantes desplegadas a lo largo de la monografía como lo es *ECM* (Esperanza Condicional - Maximización).

El algoritmo *EM*, inicialmente propuesto por Dempster (1977) [3] presenta una técnica iterativa general para realizar una estimación de máxima verosimilitud para parámetros de problemas en la que existen ciertos *"datos ocultos"*.

El algoritmo *EM* puede aplicarse en muchas situaciones en las que se desea estimar un conjunto de parámetros que describen una distribución en probabilidad subyacente, dada únicamente una parte observada de los datos completos producidos por la distribución. Se seguirá, en lineas generales, los trabajos bibliográficos de Lachlan (1997) [21] y Lachlan y Peel (2000)[30].

B.1.1. Los dos pasos del algoritmo EM

En general, se supone que en cada realización del experimento aleatorio se observa un parámetro y_i y existe un parámetro oculto z_i.

Denotamos entonces por $\boldsymbol{Y} = \{y_1, \ldots, y_m\}$ al conjunto de datos en m realizaciones del experimento, por $\boldsymbol{Z} = \{z_1, \ldots, z_m\}$ al conjunto de datos no observados y por $\boldsymbol{X} = \boldsymbol{Y} \cup \boldsymbol{Z}$ al conjunto de datos completos.

Los datos $\boldsymbol{Z}$ pueden considerarse una v.a. cuya distribución de probabilidad depende de los parámetros a estimar y de los datos observados $\boldsymbol{Y}$. El algoritmo *EM* proporciona un procedimiento iterativo para computar estimaciones máximo

verosímiles en ausencia de datos adicionales, donde sería directa la valoración de máximo verosimíl.

Sea $\boldsymbol{Y}$ un vector aleatorio en $\mathbb{R}^p$ con función de densidad $g(\boldsymbol{y}, \boldsymbol{\Psi})$ donde $\boldsymbol{\Psi} = (\psi_1, \ldots, \psi_d)^T$ es el vector de parámetros desconocidos en un espacio paramétrico $\boldsymbol{\Omega}$ y sea $g_c(\boldsymbol{x}, \boldsymbol{\Psi})$ la función de densidad del vector aleatorio de datos completos.

Por tanto la función de *log-verosilmilitud* para los datos completos se determina por:

$$logL_c(\boldsymbol{\Psi}) = log\, g_c(\boldsymbol{x}, \boldsymbol{\Psi}).$$

Formalmente, existen dos espacios muestrales $\mathcal{X}$ y $\mathcal{Y}$ y una función biyectiva. En lugar de observar el vector de datos completos $\boldsymbol{x}$ en $\mathcal{X}$, se observa el vector de datos incompletos $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})$ en $\mathcal{Y}$. A esto le sigue

$$g(\boldsymbol{y}, \boldsymbol{\Psi}) = \int_{\mathcal{X}(\boldsymbol{y})} g_c(\boldsymbol{x}, \boldsymbol{\Psi}) d\boldsymbol{x},$$

donde $\mathcal{X}(\boldsymbol{y})$ es un subconjunto de $\mathcal{X}$ determinado por la ecuación $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})$

Si se resuelve la ecuación

$$\frac{\partial \log L(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}} = 0,$$

a partir de un procedimiento iterativo indirecto en términos de la función de verosimilitud de datos completa $\log L_c(\boldsymbol{\Psi})$.

Para ello partimos de una valor inicial $\boldsymbol{\Psi}^{(0)}$ es algún valor inicial para $\boldsymbol{\Psi}$. El primer paso de la iteración (*E-step*) requiere el cálculo de

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)}) = E_{\boldsymbol{\Psi}^{(0)}}(\log L_c(\boldsymbol{\Psi})/\boldsymbol{y}).$$

El segundo paso (*M-step*) insume la maximización de $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)})$ con respecto a $\boldsymbol{\Psi}$. Esto es,elegimos $\boldsymbol{\Psi}^{(1)}$ tal que

$$Q(\boldsymbol{\Psi}^{(1)}; \boldsymbol{\Psi}^{(0)}) \geq Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(0)}),$$

para todo $\boldsymbol{\Psi} \in \boldsymbol{\Omega}$.

Así sucesivamente, de forma que el $(k+1)$-ésimo paso está definido de la siguiente manera:

- **E-Step** Calcular $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$, donde

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = E_{\boldsymbol{\Psi}^{(k)}}(logL_c(\boldsymbol{\Psi})/\boldsymbol{y}).$$

- **M-Step** Elegir $\mathbf{\Psi}^{(k+1)}$ tal que

$$Q(\mathbf{\Psi}^{(k+1)}; \mathbf{\Psi}^{(k)}) = Q(\mathbf{\Psi}; \mathbf{\Psi}^{(k)}),$$

  para todo $\mathbf{\Psi} \in \mathbf{\Omega}$.

Y así sucesiva y alternadamente repetimos los pasos EM de forma de que la diferencia

$$L(\mathbf{\Psi}^{(k+1)}) - L(\mathbf{\Psi}^{(k)}),$$

sea arbitrariamente pequeña en caso de convergencia de la sucesión de valores de verosimilitud $\{L(\Psi^{(k)})\}$.

B.1.2. Un ejemplo: Estimación de k-Medias

Para ilustrar el funcionamiento del algoritmo EM se muestra su uso en la estimación de k-Medias $\theta = (\mu_1, \ldots, \mu_k)$ de una mezcla de normales con desviación estándar σ conocida.

Sea $Z = \{z_j\}$ los datos producidos por la distribución. Los datos no observados son $X = \{(x_{1j}, \ldots, x_{kj})\}$, $x_{ij} = \{0, 1\}$ y $\sum_{i=1}^{k} x_{ij} = 1$ las cuáles marcan de que distribución proviene el dato z_j. Para un único conjunto de datos $y_j = (z_j, x_{1j}, \ldots, x_{kj})$ su verosimilitud dado $h' = (\mu_1^{'}, \ldots, \mu_k^{'})$ es

$$p(y_j|h') = p(z_j, x_{1j}, \ldots, x_{kj}|h') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{k} x_{ij}(z_j - \mu_i^{'})^2}.$$

La verosimilitud para el conjunto de m datos es

$$log\, p(Y|h^{'}) = \log \Pi_{i=j}^{n} p(y_j|h^{'}) = \sum_{j=1}^{m} \log\, p(y_j|h^{'}) =$$
$$\sum_{j=1}^{m} \left(\log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{i=1}^{k} x_{ij}(z_j - \mu_i^{'})^2 \right).$$

Se puede calcular ahora la esperanza sobre la distribución de los datos ocultos :

$$E \log p(Y|h^{'}) = E \sum_{j=1}^{m} \left(\log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{i=1}^{k} x_{ij}(z_j - \mu_i^{'})^2 \right) =$$
$$\sum_{j=1}^{m} \left(\log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{i=1}^{k} E[x_{ij}](z_j - \mu_i^{'})^2 \right).$$

De forma sencilla se puede calcular los valores esperados de los datos ocultos $E[x_{ij}]$ a partir de la hipótesis actual y de los datos observados Z siendo esta la probabilidad de que la muestra z_j haya sido generada pro la distribución normal i.

$$E[x_{ij}] = \frac{p(x = z_j|\mu = \mu_j)}{\sum_{n=1}^{k} p(x = z_j|\mu = \mu_n)} = \frac{e^{-\frac{1}{2\sigma^2}(z_j-\mu_i)^2}}{\sum_{n=1}^{k} e^{-\frac{1}{2\sigma^2}(z_j-\mu_n)^2}}.$$

Se realiza ahora el paso de maximización respecto a $h^{'} = \{\mu_1^{'}, \ldots, \mu_1^{'}\}$, es decir

$$\arg\max_{h^{'}} \sum_{j=1}^{m} \left(\log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{i=1}^{k} E[x_{ij}](z_j - \mu_i^{'})^2 \right) = \arg\min_{h^{'}} \sum_{i=1}^{k} \sum_{j=1}^{m} E[x_{ij}](z_j - \mu_i^{'})^2.$$

La hipótesis de máxima verosimilitud es la que minimiza la suma ponderada de los errores al cuadrado, donde la contribución en cada instancia z_j al error, que define $\mu_i^{'}$, está ponderada por $E[x_{ij}]$.

B.1.3. Propiedad del Algoritmo EM

Dempster, Laird, y Rubin (1977) demuestran ciertas propiedades del algoritmo que lo hacen estable y eficiente en determinados ámbitos.

Teorema 16 (Monotonía del algoritmo EM) *La función de verosimilitud es no decreciente bajo el algoritmo EM. Esto es quiere decir*

$$L(\mathbf{\Psi}^{(k+1)}) \geq L(\mathbf{\Psi}^{(k)}),$$

para $k = 0, 1, 2, \ldots$.

Por lo tanto si la sucesión creciente de verosimilitudes $\{L(\mathbf{\Psi}^{(k)})\}$ es acotada entonces $L(\mathbf{\Psi}^{(k)})$ converge monotónicamente a L^*.

B.1.4. Convergencia de una sucesión EM a un valor estacionario

Bajo ciertos supuestos sobre la función de verosimilitud se pueden obtener condiciones suficientes de convergencia del algoritmo. La demostraciones de esta sección se encuentran explicitadas en Mc Lachlan y Krishnan (1997) o Mc Lachlan y Peel (2000).

Perturbación en el algoritmo EM

En general, L^* es un valor estacionario. Esto es $L^* = L(\Psi^*)$ para algún punto Ψ^* el cual verifica

$$\partial L(\mathbf{\Psi})/\partial \mathbf{\Psi} = 0,$$

o lo que es equivalente

$$\partial log\, L(\mathbf{\Psi})/\partial \mathbf{\Psi} = 0.$$

L^* es en general un máximo local, en caso de no serlo, se perturba Ψ^* y el algoritmo diverge del punto silla.

Si L es multimodal en Ω la convergencia depende del punto de partida $\Psi^{(0)}$ pero si L es unimodal en Ω converge al MLE para todo $\Psi^{(0)}$.

Condiciones de convergencia

Teorema 17 (Límites puntuales del algoritmo) *Si $Q(\Psi;\phi)$ es una función continua ambos parámetros (condición de Wu (83)). Entonces todos los limites puntuales de cualquier instancia $\{\Psi^{(k)}\}$ de el algoritmo EM son puntos estacionarios de $L(\Psi)$ y $L(\Psi^{(k)})$ converge monotónicamente a algún valor $L^* = L(\Psi^*)$ para algún punto estacionario Ψ^*.*

Teorema 18 (Convergencia a un punto estacionario) *Sea $\{\Psi^{(k)}\}$ es una instancia de algoritmo EM con la propiedad adicional de que:*

$$[\partial Q(\Psi;\Psi^k)/\partial\Psi]_{\Psi=\Psi^{k+1}} = 0.$$

Suponiendo que $\partial Q(\Psi;\phi)/\partial\Psi$ es continua en Ψ y ϕ. Entonces Ψ^k converge a un punto estacionario Ψ^* con $L(\Psi^*) = L^*$ el limite de $L(\Psi^{(k)})$ si

$$\mathcal{L}(L^*) = \{\Psi^*\},$$

o

$$\|\Psi^{(k+1)} - \Psi^{(k)}\| \to 0, \quad k \to \infty, \quad \text{y es discreta} \quad \mathcal{L}(L^*).$$

Corolario 2 (Convergencia al máximo) *Si se supone que $L(\Psi)$ es unimodal en Ω con Ψ^* siendo el único punto estacionario y sea $\partial Q(\Psi;\phi)/\partial\Psi$ es continuo en Ψ y ϕ. Entonces cualquier sucesión EM $\{\Psi^{(k)}\}$ converge a un único máximo Ψ^* de $L(\Psi)$,esto es, converge a la única estimación máximo verosímil de Ψ.*

B.2. El algoritmo ECM

Una extensión del algoritmo EM ,es el algoritmo *ECM* (EM condicional) propuesto por Meng y Rubin (1993) . La idea es modificar el paso de maximización volviéndolo más simple condicionándolo a una función de los parámetros bajo estimación.El algoritmo EMC por tanto remplaza el M-paso del algoritmo EM por un número de computabilidad más sencilla, llamado CM-paso. Si bien el algoritmo ECM es mas lento con respecto al número de iteraciones , es mas rápido en términos computacionales,preservando las propiedades de convergencia del EM (como por ejemplo la monotonía)

Definición 7 (Construcción del algoritmo ECM) *Se suplanta el M-paso es remplazado por un número $S > 1$ de pasos. $\Psi^{(k+s/S)}$ denota el valor de Ψ en el s-ésimo paso de CM en la $(k+1)$-ésima iteración, donde $\Psi^{(k+s/S)}$ es elegido para maximizar*

$$Q(\Psi; \Psi^{(k)}),$$

conforme a

$$g_s(\Psi) = g_s(\Psi^{k+(s-1)/S}).$$

Siendo $C = \{g_s(\Psi), s = 1, 2, \ldots, S\}$ un conjunto de S preseleccionadas funciones.

Así $\Psi^{k+(s-1)/S}$ satisface :

$$Q(\Psi^{(k+s/S)}; \Psi^{(k)}) \geq Q(\Psi; \Psi^{(k)}) \quad \forall \Psi \in \Omega_S(\Psi^{(k+(s-1)/S)}),$$

donde

$$\Omega_S(\Psi^{(k+(s-1)/S)}) \equiv \{\Psi \in \Omega : g_S(\Psi) = g_S(\Psi^{(k+(s-1)/S)})\}.$$

El valor de Ψ en el paso final CM es $\Psi^{(k+S/S)=\Psi^{(k+1)}}$ Se tiene que

$$Q(\Psi^{(k+1)}; \Psi^{(k)}) \geq Q(\Psi^{(k+(S-j)/S)}; \Psi^{(k)}) \forall j = 1, 2, \ldots, S.$$

B.3. El algoritmo de Dykstra

Dykstra es un algoritmo de proyección alternativo para resolver el problema de optimización convexa de encontrar el punto de una intersección de un numero finito de conjuntos cerrados y convexos mas cercano a otro punto dado. Fue introducido en 1983 por Richard Dykstra [6]. En el trabajo monográfico es utilizado para encontrar la distancia de un punto a una intersección de conos convexos.

Es un algoritmo muy útil cuando se conoce la distancia a cada uno de los convexos de forma sencilla y se quiere hallar la distancia a la intersección de ellos. En la práctica

es también importante encontrar adecuados criterios de parada del algoritmo debido a que este varía sustancialmente de un problema a otro.

Se considera el algoritmo de Dykstra para resolver el siguiente problema

$$(P) \qquad \min_{x\in\Omega} \|x^0 - x\|.$$

donde x^0 es un punto dado, Ω es un conjunto cerrado y convexo, y la norma $\|\|$ es una norma proveniente de un producto interno. La solución del problema x^* es llamada la proyección de x^0 en Ω y se anota $P_\Omega(x^0)$.

Se considera ahora el caso de que:

$$\Omega = \bigcap_{i=1}^{p} \Omega_i,$$

donde Ω_i son conjuntos cerrados y convexos en $\Re^n$, con $i = 1, 2, \ldots, p$ y $\Omega \neq \phi$. Se asume que dado $z \in \Re^n$ el calculo de $P_\Omega(z)$ es no trivial, pero sin embargo para cada Ω_i es sencilla la obtención de $P_{\Omega_i}(z)$, por ejemplo cuando Ω_i es una bola, una caja, un subespacio afín o un cono.

Dykstra es un algoritmo que proyecta en forma inteligente sobre cada conjunto Ω_i,para completar un ciclo que es repetido iterativamente.

Sabemos que si Ω es un conjunto no vacío, cerrado y convexo en $\Re^n$, dado $x^0 \in \Re^n$ el problema de optimización tratado presenta solución $x^* = P_\Omega(x^0)$ y es única . Esta solución esta caracterizada por el criterio de Kolmogorov.

$$\left\langle x^0 - x^*, x^* - x \right\rangle \geq 0 \quad \forall x \in \Omega, \quad x^* \in \Omega$$

B.3.1. Construcción del algoritmo

Dykstra es un algoritmo que resuelve el problema (P) mediante la generación de dos sucesiones: la de iteraciones de $\{x_i^k\}$ y la de incrementos $\{y_i^k\}$. Ambas secuencias son definidas de manera recursiva mediante las siguientes formulas:

$$\begin{array}{ll} x_0^k = x_p^{k-1} & \\ x_i^k = P_{\Omega_i}(x_{i-1}^k - y_i^{k-1}) & i = 1, 2, \ldots, p. \\ y_i^k = x_i^k - (x_{i-1}^k - y_i^{k-1}) & i = 1, 2, \ldots, p \end{array}$$

para $k = 1, 2, \ldots$ con valores iniciales $x_p^0 = x^0$ y $y_i^0 = 0$ para $i = 1, 2, \ldots, p$.

Observaciones

El incremento y_i^{k-1} asociado con Ω_i, en el ciclo previo es siempre sustraído antes de la proyección sobre Ω_i. Sólo el incremento anterior para cada Ω_i es guardado.

Si Ω_i es un subespacio afín cerrado entonces el operador P_{Ω_i} es lineal y este no es requerido en el k-ésimo ciclo para sustraer y_i^{k-1} antes de proyectar en Ω_i. Para ser precisos $P_{\Omega_i}(y_i^{k-1}) = 0$.

Para $k = 1, 2, \ldots,$ y $i = 1, 2, \ldots, p$ es fácil observar la siguientes relaciones:

$$x_p^{k-1} - x_1^k = y_1^{k-1} - y_1^k$$

$$x_{i-1}^{k-1} - x_i^k = y_i^{k-1} - y_i^k$$

donde $x_p^0 = x^0$ y $y_i^0 = 0$ para todo $i = 1, 2, \ldots, p$.

Teorema 19 (Boyle y Dikstra, 1986) *Sea $\Omega_1, \ldots, \Omega_p$ son conjuntos cerrados y convexos de $\Re^n$ tal que $\Omega = \bigcap_{i=1}^p \Omega_i \neq \phi$. Para todo $i = 1, 2, \ldots, p$ y algún $x^0 \in \Re^n$, la sucesión $\{x_i^k\}$ antes descrita converge a $x^* = P_\Omega(x^0)$ (esto es $\|x_i^k - x^*\| \to 0$ cuando $k \to \infty$).*

Es fácil mostrar que un criterio poco adecuado es tomar la sucesión de proyecciones sobre Ω_i y parar el proceso cuando la norma entre 2 proyecciones consecutivas es menor que una tolerancia fijada anteriormente o cuando el promedio de proyecciones sobre cada Ω_i no varía.

Se presenta otro criterio de parada más robusto presentado en el 2004 por E. Birgin y M. Raydan [37].

Teorema 2

Sea x^0 una elemento de $\Re^n$. Se consideran las dos sucesiones $\{x_i^k\}$ y $\{y_i^k\}$ definidas previamente y sea c^k definida de la siguiente forma:

$$c^k = \sum_{m=1}^{k} \sum_{i=1}^{p} \|y_i^{m-1} - y_i^m\|^2 + 2 \sum_{m=1}^{k-1} \sum_{i=1}^{p} \left\langle y_i^m, x_i^{m+1} - x_i^m \right\rangle.$$

Entonces se cumple que el k^{esimo} ciclo de al algoritmo Dykstra cumple:

$$\|x^0 - x^*\|^2 \geq c^k.$$

Mas aún, la igualdad es cierta cuando k tiende a infinito.

Anotaremos

$$c^k = c_L^k + c_S^k,$$

donde

$$c_L^k = \sum_{m=1}^{k} c_I^m,$$

$$c_I^m = \sum_{i=1}^{p} \|y_i^{m-1} - y_i^m\|^2,$$

y

$$c_S^k = 2\sum_{m=1}^{k-1}\sum_{i=1}^{p} \left\langle y_i^m, x_i^{m+1} - x_i^m \right\rangle.$$

Reglas de Parada

Para cualquier $k \in N$, si $x^k \neq x^*$ entonces $c_I^{k+1} > 0$ y ademas $c^k < c^{k+1}$, $c_L^k < c_L^{k+1}$.

Por tanto 2 reglas de parada pueden ser:

$$c_I^k = \sum_{i=1}^{p} \|y_i^{k-1} - y_i^k\|^2 \leq \epsilon,$$

o similarmente

$$c^k - c^{k-1} = c_I^k + 2\sum_{m=1}^{k-1}\sum_{i=1}^{p} \left\langle y_i^m, x_i^{m+1} - x_i^m \right\rangle \leq \epsilon.$$

Dado $w > 0, w \in R^n$ se define en R^n el producto interno entre x e y (con respecto a w) como

$$< x, y >= \sum_{i=1}^{n} x_i y_i w_i.$$

Sea K un cono cerrado y convexo en R^n, es decir , si $x, y \in K$, $a, b > 0$ implica que $ax + by \in K$.

El cono dual queda definido entonces:

$$K^* = \{y / < y, x > \leq 0 \quad \forall x \in K\}.$$

K^* es también un cono cerrado convexo que cumple $K^{**} = K$.

Si g^* es el punto óptimo del problema,

$$\min_{x \in K} \|g - x\|,$$

se sabe que:

1. $< g - g^*, g^* >= \sum_{i=1}^{n} (g_i - g_i^*) g_i^* w_i = 0.$

2. $< g - g^*, f >= \sum_{i=1}^{n} (g_i - g_i^*) f_i w_i \leq 0 \quad \forall f \in K.$

Barlow y Brunk demuestran en 1972 que $g - g^*$ es solución del problema dual

$$\min_{x \in K^*} \|g - x\|.$$

El problema principal consiste en buscar la solución del problema

$$\min_{x \in \cap_1^r K_i} \|g - x\|,$$

donde $K_i \quad i = 1, 2 \ldots, r$ son conos cerrados y convexos. Se supone conocida la solución en cada K_i, es decir, se conoce la solución del problema,

$$\min_{x \in K_i} \|f - x\|,$$

para toda f para todo i de 1 a r.

La suma $K_1 + K_2 + \ldots + K_r$ es un cono convexo, pero en general no tiene porque ser cerrado (salvo que sea finitamente generado). Lo cuál es cierto pues

$$(K_1 + \ldots + K_r)^* = K_1^* \cap \ldots \cap K_r^*,$$

y además,

$$(K_1 \cap \ldots \cap K_r)^* = K_1^* + \ldots + K_r^*.$$

Se anota $P(f|K_i)$ a la proyección de f sobre K_i y diremos que $n \mod r = i$ sí y solo sí $n = kr + i$ para algún k entero e i entero entre 0 y r.

El procedimiento es el siguiente:

1. Tomemos inicialmente $g_0 = g$,$I_i = 1, \ldots, r$, y $n = 1$.

2. Sea $g_n = P(g_{n-1} - I_{n \, mod \, r})$ y luego se actualiza $I_{n \, mod \, r}$ por $g_n - (g_{n-1} - I_{n \mod r})$.

3. Se remplaza n por $n + 1$.

Cabe hacer notar que si K_i son todos subespacios entonces $P(\cdot|K_i)$ es un operador lineal.

Apéndice C

Algunos comentarios sobre la selección de variables y la estimación del número de cluster

C.1. Selección de variables

Seleccionar variables es detectar básicamente a aquellas no informativas, como las que presentan multicolinealidad.

En muchos casos prácticos la cantidad de variables - que no debe confundirse con la cantidad de información - es demasiado alto. Estas variable de "ruido" no informativas o que proporcionan información redundante (conjunto de variables fuertemente correlacionadas que pueden producir multicolinealidad).

Por tanto, una información semejante a la extraída puede ser resumida en un subconjunto de las variables originales. En clustering interesa encontrar aquellas variables que "explican" de mejor manera los grupos buscados.

Técnicas como componentes principales apuntan a esto pero en la mayoría de los caso las nuevas variables, combinaciones de las anteriores, pierden interpretabilidad. Ademas es un método que produce una deformación en la distancia, y como consecuencia, una deformación en los resultados.

Básicamente para tratar los problemas de variables "con ruido" no informativas y multicolinealidad es susutituir los valores de la variable por su media marginal o por la media condicional respectivamente.

El desarrollo siguiente es extraído del trabajo "Selection of variables for cluster analysis and classification rules" realizado por R. Fraiman, A. Justel y M. Svarc (2006).

C.1.1. Variables No Informativas

Sea $X = (X_1, \ldots, X_p)$ un vector aleatorio con distribución P y fijado en K el número de cluster, se tiene

$$f : \mathbb{R}^p \to \{1, 2, \ldots, K\},$$

función que le asigna a cada observación su cluster, determinando una partición del espacio conformada por $G_k = f^{-1}(k)$ que cumplen:

$$P\left(\cup_{k=1}^{K} G_k\right) = 1.$$

La idea es "cegar" aquellas variables de modo que los datos se mantengan en el mismo cluster original, es clave que la partición se realiza en el espacio original por tanto no puedo reducir mi dimensión. La idea es encontrar un subconjunto de índices $I \subset \{1, 2, \ldots, p\}$ de forma que el nuevo vector construido $Y^I \in \mathbb{R}^p$ este respecto lo mas cerca posible del vector X de información completa. Y^I es una vector en donde las variables "cegadas" son sustituidas por su valor medio.

Si llamamos $I = \{i_1, \ldots, i_d\} \subset \{1, 2, \ldots, p\}$, se define

$$Y^I = \begin{cases} Y_i = X_i & \text{si } i \in I \\ Y_i = E(X_i) & \text{en otro caso} \end{cases}$$

$E(X_i)$ puedes ser sustituida por la mediana o algún otro estimador de locación de la i-ésima coordenada siempre que halla teoremas de consistencia fuertes para el estimador.

Se fija $d > p$. La función objetivo poblacional estará dada por,

$$h(I) = \sum_{k=1}^{K} P(f(X) = k, f(Y^I) = k),$$

e interesa encontrar las d variables donde alcanza su máximo. La versión empírica consiste en los siguientes pasos:

1. Dados los datos $X_1, X_2, \ldots, X_n \in \mathbb{R}^p$ aplicar el procedimiento de partición al conjunto de datos y obtener le función empírica de asignación al cluster:

   $$f_n : \mathbb{R}^p \to \{1, 2, \ldots, K\}.$$

   Se anota $G_k^n = f_n^{-1}(k)$ con $k = 1, 2, \ldots, K$

2. Fijado $d < p$ subconjunto de índices $I \subset \{1, 2, \ldots, p\}$ con $\#I = d$ se define el conjunto de vectores aleatorios $\{X_j^* : 1 \leq j \leq n\}$ que verifican :

$$X_j^*[i] = \begin{cases} X_j[i] & \text{si } i \in I \\ \bar{X}[i] & \text{en otro caso} \end{cases}$$

siendo $X[i]$ la i-ésima coordenada del vector X y $\bar{X}[i]$ es la i-ésima coordenada del vector de medias.

3. Calcular la función empírica objetivo

$$h_n(I) = \frac{1}{n} \sum_{k=1}^{K} \sum_{j=1}^{n} \mathcal{I}_{f_n(X_j=k)} \mathcal{I}_{f_n(X_j^*=k)}.$$

4. Buscar un conjunto $I_{d,n} \equiv I_n$ con $\#I_n = d$ de forma que maximize la función empírica objetivo h_n

Se pueden realizar las siguientes observaciones :

- Bajo ciertas hipótesis sobre el método de partición de cluster el algoritmo es fuertemente consistente.
- Un manera de ahorrar tiempo computacional para alcanzar el conjunto de la d variables óptimas es realizar un algoritmo hacia adelante y luego hacia atrás (forward - backward)

C.1.2. Multicolinealidad

El procedimiento anterior esta diseñado básicamente para encontrar variables de "ruido" no informativas, pero este puede fallar en presencia de multicolinealidad.

Con el fin de hacer frente a este problema, se tiene que cambiar la definición del vector Y^I. Nuestros índices en el complemento de I serán definidos a través del mejor predictor constante: la esperanza condicional.

Se define ahora el vector informativo Z^I al igual que Y^I, pero en lugar de colocar las esperanzas marginales tomamos la esperanza condicional de X_i dadas las demás variables $\{X_l : l \in I\}$, siendo este el mejor predictor de X_i basado en las demás variables.

Sin embargo, este enfoque requiere un tamaño de muestra grande para poder estima la media condicional, y es costoso en términos computacionales. También es delicada la elección del parámetro de suavizado.

En la práctica, la versión empírica consiste en los mismos pasos que en el método basado en el uso de la media, excepto el segundo, que es sustituido por el siguiente paso:

Fijado $d < p$ subconjunto de indices $I \subset \{1, 2, \ldots, p\}$ con $\#I = d$, se fija un entero r (número de vecinos mas cercanos a ser utilizados). Para cada $j = 1, 2, \ldots, n$ se encuentra el conjunto de índices C_j de los r vecinos mas cercanos de $X_j[I]$ entre $\{X_1[I], \ldots, X_n[I]\}$.

Ahora definimos un vector aleatorio $\{X_j^* : 1 \leq j \leq n\}$ que verifica:

$$X_j^*[i] = \begin{cases} X_j[i] & \text{si } i \in I \\ \frac{1}{r} \sum_{m \in C_j} X_m[i] & \text{en otro caso} \end{cases}$$

Observaciones:

- Para procedimientos mas robustos se puede considerar la mediana local $X_j^*[i] = (\{X_m[i] : m \in C_j\})$.
- Al igual que en el caso anterior bajo otros supuestos se puede asegurar la consistencia, y a nivel computacional, se realiza un algoritmo forward - backward.

C.2. Estimación del número de cluster

Hartigan (1975) define el número de cluster en una población de d-variables es el número de componentes conexas del conjunto $\{f > c\}$, donde f denota la subyacente función de densidad en $\mathbb{R}^d$ para alguna constante c dada. En los algoritmos usuales de cluster toman q con un parámetro dado.

Fraiman y Cuevas (2000) proponen estimar el número de cluster basada en el cálculo del número de componentes conexas de un estimador del conjunto $f > c$. Este estimador es construido como unión de bolas con centros en una determinada submuestra, que es seleccionada vía el estimador no paramétrico de f.

Cada una de estas componentes conexas determinarán un cluster. Básicamente la idea de fondo esta ligada a la noción del modo de la distribución, pero la noción de cluster es más fácil de manejar en términos geométricos.

Si f es una densidad desconocida en $\mathbb{R}^d$ y $c > 0$ una constante fija, el objetivo es estimar el número de componentes conexas. Si $T(S)$ es un estimador de este paramétro, donde S es el conjunto de nivel $S(f; c) = \{f > c\}$ a partir de una muestra $X_1, X_2, \ldots, X_n$ procedente de f.

Una primera idea sería estimar $T(S)$ por el estimador plug-in $T\{S(\hat{f}_n; c)\}$ donde$\hat{f}_n$ es la estimación no paramétrica de f.

Sin embargo, es en general dificultoso su cálculo por la complicada estructura geométrica de las curvas de nivel $\{\hat{f}_n > c\}$. Fraiman y Cuevas proponer otro estimador T_n, también inspirados en un enfoque plug-in pero computacionalmente mas simple. La idea es aproximar el conjunto de nivel estimado $\{\hat{f}_n > c\}$ por un conjunto mas sencillo donde el número de componentes conexas pueden ser evaluadas mediante algoritmos mas sencillos.

El estimador consiste en una unión de bolas centradas en las observaciones muestrales X_i tal que $X_i \in \{\hat{f}_n > c\}$. Esta basado en el hecho de que el conjunto $S = \{f > c\}$ puede ser expresado como $S = \{f_c > 0\}$, donde f_c es la densidad de la distribución condicional de X_1 dada S.

Proponen también una versión ampliada de bootstrap, denotada por T_{n,N^n}de el estimador T_n. Esta basado en un bootstrap suavizado en $Z_1, Z_2, \ldots, Z_N$ extraídas de las distribución $\hat{f}_n$ condicionada a $\{\hat{f}_n > c\}$

C.2.1. Estimando el número de cluster

Se considera el problema de estimar $T($$S$, el estimador natural plug-in de él conjunto de nivel $S = S(f; c) = \{f > c\}$ está dado por $\{\hat{f}_n > c\}$. Debido a las dificultades prácticas que presenta la efectiva evaluación de $T\{S(\hat{f}_n; c)\}$ se reemplaza $\{\hat{f}_n > c\}$ por un estimador mas simple definido por,

$$\hat{S}_n = \cup_{i=1}^{k_n} \bar{B}(X_i, \epsilon_n).$$

Siendo el estimador T_n como el número de componentes conexas de $\hat{S}_n$, esto es

$$T_n = T(\hat{S}_n)$$

Un simple algoritmo para evaluar T_n está basado en la siguiente idea: una componente conexa $\hat{S}_n$ es asociada con un "spanning tree" con vértices X_i y bordes mas pequeños que $2\epsilon_n$. Es decir, bajo ciertas condiciones, el estimador es fuertemente consistente, es decir $T_n \to T(S)$ casi seguramente.

Apéndice D

Demostración de la consistencia de K-Medias

El primer paso consiste en encontrar un M (no dependiendo de ω) tan grande que, cuando n es lo suficientemente, al menos en un punto de A_n es contenido en una bola cerrada $B(M)$ centrada en el origen y de radio M. Es conveniente asumir que $\phi(r) \to \infty$ cuando $r \to \infty$; la prueba para ϕ acotada es sólo un poco más complicada.

Se debe encontrar un r de forma tal que la bola K de radio r y centrada en el origen tenga un medida P positiva. Para los propósitos de este primer paso será suficiente que M sea lo suficientemente grande para hacer $\phi(M-r)P(K) > \int \phi(\|x\|)P(dx)$; para el segundo y tercer paso dos requerimientos más serán puestos en M.

Por supuesto hipótesis $\Phi(A_n, P_n) \leq \Phi(A_0, P_n))$ para cualquier conjunto A_0 conteniendo como mucho k puntos. Elegir A_0 para contener un sólo punto en el origen. Entonces

$$\Phi(A_0, P_n) - \int \phi(\|x\|)P_n(dx) \to \int \phi(\|x\|)P(dx)\, c.s. \tag{D.1}$$

Si, para infinitamente muchos valores de n, ningún punto de A_n son contenidos en $B(M)$, entonces

$$\limsup_n \Phi(A_n, P_n) \geq \lim_n \phi(M-r)P_n(K) = \phi(M-r)P(K)\, c.s.$$

Esto haría $\Phi(A_n, P_n) > \Phi(A_0, P_n)$ infinitamente seguido: una contradicción. Sin pérdida de generalidad se puede asumir que A_n siempre contiene al menos un punto de $B(M)$.

Si $k = 1$ el siguiente paso en la prueba puede ser salteado; si $k > 1$ entonces tenemos que mostrar que, para n suficientemente grande, la bola cerrada $B(5M)$, de radio $5M$ y centrada en el origen, contiene todos los puntos de A_n. Para los propósitos de un argumento inductivo, asumir que las conclusiones del teorema son válidas cuando es aplicado.

Apéndice E

Código Implementado y/o Utilizado

E.1. Framework para Simulaciones

```
generarDatos <- function(escenario) {
  etiquetas <- names(escenario)
  datos.grupos <- lapply(names(escenario),
                         function (grupo) {
                           params <- append(escenario[[grupo]]$n,
                                            escenario[[grupo]]$params)
                           do.call(escenario[[grupo]]$generador, params)
                         })
  datos <- do.call(rbind, datos.grupos)

  grs <- length(etiquetas)

  ee <- lapply(names(escenario),
               function(grupo) {rep(grupo, escenario[[grupo]]$n)})
  clv <- as.factor(unlist(ee))

  n <- nrow(datos)
  n.grs <- table(clv)

  return( list(datos = datos,
               grs = grs, n = n, n.grs = n.grs,
               clv = clv)
        )
}

clasificacionVerdadera <- function(escenario) {
  ee <- lapply(names(escenario),
               function(grupo) {rep(grupo, escenario[[grupo]]$n)})
  return(as.factor(unlist(ee)))
}

alinearNiveles <- function(datos, nivelesBien) {
  nb <- levels(nivelesBien)
  qn <- length(nb)
  datos <- as.integer(factor(datos))
  pet <- permn(nb)
```

```
37  aciertos <- sapply(pet, function(x) {
      sum(nivelesBien == factor(datos, levels = 1:qn, labels = x))
39  })
    nivs <- pet[[which.max(aciertos)]]
41  orden <- as.vector(sapply(nb, function (x) {which(x == nivs)}))
    return(factor(datos, levels = orden, labels = nb))
43 }

45 resultadoTecnica <- function(resultado, correcto) {
    acierto <- (resultado == correcto)
47  return(list("resultado" = resultado,
                "acierto" = acierto,
49              "cm" = table(resultado, correcto)
                )
51        )
  }
53
  evaluarTecnica <- function (tecnica, datos){
55  if (tecnica$datos == TRUE) {
      params <- append(list(datos), tecnica$params)
57  } else {
      params <- tecnica$params
59  }
    res <- do.call(tecnica$funcion, params)
61  if (tecnica$resultado != FALSE) {
      res$.resultado <- res[[ tecnica$resultado ]]
63  } else {
      res$.resultado <- res[1]
65  }
    return(res)
67 }

69 simular <- function (q = 5, grupos, tecs, nucleos = 6) {
    options(cores = nucleos)
71  simulaciones <- list()

73  tt <- system.time({
      simulaciones <- foreach (i = 1:q) %dopar% {
75    cat("\n- - - Comenzando Simulacion:", i, "- - - - - \n")
      escenario <- generarDatos(grupos)
77    simulacion <- list()
      for (tecnica in names(tecs)) {
79      salida <- evaluarTecnica(tecs[[tecnica]], escenario$datos)
        alineada <- alinearNiveles(salida$.resultado, escenario$clv)
81      resultado <- resultadoTecnica(alineada, escenario$clv)
        simulacion[[tecnica]] <- append(resultado, list("out" = salida))
83    }

85    list("datos" = escenario$datos, "tecnicas" = simulacion)
    }
87  })[3]

89  cat("\n - - - - - - - - - - - - - - - - - - \n --- Terminado en", tt,
        "segundos.\n -- Gracias por utilizar el framework. \n")
```

```
  return(list("escenario" = grupos, "sims" = simulaciones))
}

eD <- function (simulaciones, criterio) {
  obs.index <- lapply(simulaciones$sims, function (s) {
      seleccion <- sapply(s$tecnicas, function (t) { criterio(t)} )
  })

  return(obs.index)
}

extraerDatos <- function (simulaciones, criterio = todos) {
  obs.index <- lapply(simulaciones$sims, function (s) {
      seleccion <- lapply(s$tecnicas, function (t) { criterio(t)} )
  })
  obs <- data.frame()
  for (i in 1:length(obs.index)) {
   for (t in names(obs.index[[i]])) {
    o <- simulaciones$sims[[i]]$datos[obs.index[[i]][[t]], ]
    tec <- rep(t, nrow(o))
    clas0 <- simulaciones$sims[[i]]$tecnica[[t]]
    clas <- clas0$resultado[obs.index[[i]][[t]]]
    clv <- simulaciones$escenario$clv[obs.index[[i]][[t]]]
    bien <- clv == clas
    nsim <- rep(i, nrow(o))
    obs.d <- data.frame(o, tec, nsim, clas, clv, bien)
    obs <- rbind(obs, obs.d)
   }
  }
  return(obs)
}

obsMalClasificadas <- function (tecnica) {
  return(which(tecnica$acierto == FALSE))
}

obsTodas <- function (tecnica) {
  return(1:length(tecnica$acierto))
}
```

E.1.1. Medidas de Performance

```
evaluarMedida <- function (simulaciones, medida) {
  medidas <- lapply(simulaciones$sims, function (s) {
    do.call(rbind, list( sapply(s$tecnicas, function (t) { medida(t)})
    ))
  })
  return(do.call(rbind, medidas))
}

porcentajeAcierto <- function (tecnica) {
  return(mean(tecnica$acierto))
}
```

```
porcentajeError ← function (tecnica) {
  return((sum(tecnica$cm) - sum(diag(tecnica$cm))) / sum(tecnica$cm))
}

especificidad ← function (tecnica) {
  cm ← tecnica$cm
  verdaderosNegativos ← sum(diag(cm))
  falsosPositivos ← sum(cm[upper.tri(cm)])
  return(verdaderosNegativos / (verdaderosNegativos + falsosPositivos))
}

sensibilidad ← function (tecnica) {
  cm ← tecnica$cm
  verdaderosPositivos ← sum(diag(cm))
  falsosNegativos ← sum(cm[lower.tri(cm)])
  return(verdaderosPositivos / (verdaderosPositivos + falsosNegativos))
}

ruidoMalClasificado ← function (tecnica) {
  cm ← tecnica$cm
  gr ← ncol(cm)
  ruido ← cm[, gr]
  return(sum(ruido[1:(gr - 1)]) / sum(ruido))
}

obsClasificadasRuido ← function (tecnica) {
  cm ← tecnica$cm
  gr ← ncol(cm)
  ruido ← cm[gr, ]
  return(sum(ruido[1:(gr - 1)]) / sum(ruido))
}

gruposAlmenosUno ← function (tecnica) {
  cm ← tecnica$cm
  gruposBien ← (rowSums(cm) + colSums(cm) - 2 * diag(cm)) == 0
  return(sum(gruposBien) > 0)
}

gruposTodosBien ← function (tecnica) {
  cm ← tecnica$cm
  return(sum(diag(cm)) == sum(cm))
}

gruposAlmenos90 ← function (tecnica) {
  cm ← tecnica$cm
  gruposBien ← (rowSums(cm) + colSums(cm) - 2 * diag(cm)) == 0
  return(sum(gruposBien) > 0)
}

gruposResultantesSims ← function (simulaciones) {
  grupos ← names(simulaciones$escenario)
  gruposTecnicas ← lapply(simulaciones$sims, function (s) {
    lapply(s$tecnicas, function (t) {
      sapply(grupos, function (g) {
```

```
67        matrix(s$datos[t$resultado == g], ncol = 2)
      })
69    })
  })
71  return(do.call(rbind, gruposTecnicas))
}
```

E.2. Algoritmos

E.2.1. k-Medias

```
kmr <- function(datos, n.grupos, funcionDistancia = "dRobusta",
2               nSims = 50, trim = 0) {
  ## Se sortean los centros iniciales de cada grupo
4 centros <- datos[sample(1:nrow(datos), n.grupos), ]

6 ## Se inicializan variables y parametros
  distancias <- matrix(nrow = nrow(datos), ncol = n.grupos)
8 variacion <- 1
  max.iter <- 50
10 iter <- 0
  distancia <- match.fun(funcionDistancia)
12
  ## Mientras la variacion de los centros sea significativa
14 ## y no se haya iterado lo suficiente (seguro contra errores)
  while (variacion > 0.001 && iter <= max.iter) {
16
    ## Calcular la distancia de cada observacion a cada centro
18  for (k in 1:n.grupos) {
      distancias[, k] <- apply(datos, 1, distancia, v = centros[k, ])
20  }

22  ## Asignar cada observacion al grupo con el centro mas cercano
    asignacion <- apply(distancias, 1, which.min)
24
    ## Con la nueva asignacion, calcular los nuevos centros de los
26  ## grupos mediante optimizacion
    centrosNuevos <- matrix(nrow = n.grupos, ncol = ncol(datos))
28  for (k in 1:n.grupos) {
      grupo <- datos[which(asignacion == k), ]
30    if (length(grupo) > 0) { # Por si no encuentra obs en el grupo
        cn <- optim(c(0, 0),
32                  funcionCentro(grupo,
                                  "funcionDistancia" = funcionDistancia,
34                                verbose = TRUE))
        centrosNuevos[k, ] <- cn$par
36    } else {
        centrosNuevos[k, ] <- centros[k, ]
38    }
    }
40
    ## Calcular la variacion de los centros como la suma de las
42  ## distancias entre los centros nuevos y viejos
```

```
    variacion <- 0
    for (k in 1:n.grupos) {
      variacion <- variacion + distancia(centros[k, ], centrosNuevos[k, ])
    }
    ## Actualizar los centros y la cantidad de iteraciones
    centros <- centrosNuevos
    iter <- iter + 1
  }

  if (trim > 0) {
    dsMaha <- data.frame("dgind" = c(), "dists" = c())
    ## Calcular la dist. de Mahalonbis de cada observacion a cada centro
    for (k in 1:n.grupos) {
      dgind <- which(asignacion == k)
      dgrupo <- datos[dgind, ]
      dists <- mahalanobis(dgrupo, centros[k, ], cov(dgrupo))
      dsMaha <- rbind(dsMaha, data.frame(dgind,dists))
    }
    orden <- order(dsMaha[,2], decreasing = TRUE)
    obs.trim <- dsMaha[orden[1:floor((length(orden) * trim))], 1]
    asignacion[obs.trim] <- n.grupos + 1
  } else {
    obs.trim <- c()
  }

  ## Valores a retornar
  lista <- list()
  lista$cluster <- asignacion
  lista$centers <- centros
  lista$size <- as.vector(table(lista$cluster))
  lista$iters <- iter
  lista$obs <- datos
  lista$obs.trim <- obs.trim
  class(lista) <- "km"
  return(lista)
}

funcionCentro <- function (grupo, funcionDistancia = "dRobusta",
                           verbose = TRUE) {
  distancia <- match.fun(funcionDistancia)

  if (is.null(dim(grupo))) {
    grupo <- matrix(grupo, ncol = (length(grupo)))
  }

  ptos0 <- apply(grupo, 1, function (x) {paste(x, collapse = ", ")})

  ptos <- paste(funcionDistancia, "(c(", ptos0, "), v)^2", sep = "")
  funcion <- paste("min(", paste(ptos, collapse = " + "), ")")
  return(function (v) {eval(parse(text = funcion)[1])})
}
```

Métricas

```r
dEuclidea <- function(u, v) {
  ## Distancia  EuclÃdea  entre U y V
  sqrt(t(u - v) %*% (u - v))
}

dRobusta <- function (u, v) {
  ## Distancia  Robusta a partir de la  EuclÃdea
  d <- dEuclidea(u, v)
  return(d / (1 + d))
}

dRobustaM3 <- function (u, v) {
  ## Distancia  Robusta a partir de la  EuclÃdea
  d <- dEuclidea(u, v)
  if (d > 3) {
    d <- 3
  }
  return(d)
}

dAlCuadrado <- function (u, v) {
  ## Distancia  Robusta a partir de la  EuclÃdea
  d <- dEuclidea(u, v)
  return(d^2)
}
```

E.2.2. Mezcla de Distribuciones

Se utilizó la implementación de EMMIX provista por Geoff McLachlan en http://www.maths.uq.edu.au/~gjm/emmix/emmix.html

Dicha implementación está en FORTRAN, por lo que se implementó la siguiente función para trabajar desde R:

```r
emmix <- function(x, grs = 4, cov.op = 4, usar.t = FALSE, dof = 4, trim = 0) {
 ## EMMIX  parameters  to fit t- components
  n    <- as.character(nrow(x))      # number  of  samples
  p    <- as.character(ncol(x))        # number  of  features / predictors
  gmin <- as.character(grs)        # min no of  components
  gmax <- as.character(grs)        # max no of  components
  cov  <- as.character(cov.op)        # unresricted     covarience
  randStarts <- '50'  # Cantidad  de  comienzos  aleatorios
  pData      <- '100'   # Porcentaje  de  datos a  utilizar
  kStarts    <- '50'    # Comienzos  de  kMeans
  dof        <- as.character(dof)   # Grados  de  libertad

  ouFileName <- paste('salidaEMMIX', 'txt', sep ='')
  dataFileName <- "datosEMMIX.txt"
  write.table(x, dataFileName, row.names = FALSE,
              col.names = FALSE)

  semillas <- c('4321', '8765', "0912")
  params <- c('3', dataFileName, ouFileName, 'No', n, p, p,
              gmin, gmax, cov, randStarts, pData, kStarts)
```

```
21
   if (usar.t) {
23   params ← c(params, 'Y', '9', '1', rep(dof, grs), '0',
                semillas)
25 } else {
     params ← c(params, 'N', semillas)
27 }

29 system("./EMMIX/EMMIX", input=params)

31 resultado ← list()
   L ← readLines(ouFileName, n = -1)
33 nLines ← length(L)

35 grstr ← " Implied grouping of the entities into"
   glen ← nchar(grstr)
37 asig ← vector()
   ind ← which(substr(L, 1, glen) == grstr) + 1
39 while (L[ind] != '') {
     asig ← c(asig, L[ind])
41   ind ← ind + 1
   }
43 asig ← unlist(strsplit(asig, "[[:space:]*]"))
   asig ← asig[asig != ""]
45 resultado$cluster ← asig
   if(length(levels(as.factor(asig))) < 3) { stop("Menos de 3 grupos!!!!")}
47
   ## Tama~no de los clusters
49 A    ← which(L == " Number assigned to each component")
   xcls ← A + 1
51 clust.size.str ← unlist(strsplit(L[xcls[1]], '[[:space:]*]'))
   clust.size.str ← clust.size.str[clust.size.str != ""]
53 resultado$clust.size ← as.numeric(clust.size.str)

55 ## Tabla con logveros
   A    ← which(substr(L, 1, 34) == " Observation | mixture log density")
57 xcls ← A + 1
   obs.table.str ← (strsplit(L[xcls:(xcls + nrow(x))], '[[:space:]*]'))
59 o ← lapply(obs.table.str, function (x) { x[x != ""]})
   oo ← do.call(rbind, o)
61 oo ← apply(oo, 2, as.numeric)
   oo ← oo[, -1]
63 resultado$obs.logdens ← oo

65 ## Medias y Matrices de Covarianzas
     ## Parametros de las mezclas
67 L2 ← L[(length(L) - 200):length(L)]
   parametros ← list()
69 for (i in 1:grs) {
     media.txt ← " Estimated mean (as a row vector) for each component"
71   matcov.txt ← paste(" Estimated diagonal covariance matrix for component ",
                    i, sep = "")
73   A    ← which(L2 == media.txt)
     B    ← which(L2 == matcov.txt)
```

```
    xcls <- A + 1
    xcls2 <- B + 1

    media.sola <- (strsplit(L2[xcls:(xcls + grs - 1)], '[[:space:]*]'))
    m <- lapply(media.sola, function (x) { x[x != ""]})
    mm <- do.call(rbind, m)
    mm <- apply(mm, 2, as.numeric)

    mat.sola <- unlist(strsplit(L2[xcls2:(xcls2 + 1)], '[[:space:]*]'))
    mat.sola <- as.numeric(mat.sola[mat.sola != ""])

    print(mat.sola)

    mat <- matrix(c(mat.sola[1], mat.sola[2], mat.sola[2], mat.sola[3]), nrow = 2)
    parametros[[i]] <- list()
    parametros[[i]]$media <- mm[i, ]
    parametros[[i]]$matcov <- mat
  }

  resultado$parametros <- parametros
  resultado$texto.salida <- L

  if (trim > 0) {
     orden <- order(resultado$obs.logdens[, 1])
     resultado$obs.trim <- orden[1:floor((length(orden) * trim))]
     resultado$cluster[resultado$obs.trim] <- grs + 1
  }


  return(resultado)
}
```

E.2.3. TCLUST

Se utilizó la implementación de TCLUST provista en
http://www.eio.uva.es/~langel/software/tclust.r

E.3. Capítulo 3

E.3.1. Simulaciones

```
grupos <- list(
              A = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c( 0, 4),
                              sigma = matrix(c(1, 0, 0, 1), ncol = 2))
                ),
              B = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(4, 0), sigma = diag(c(1, 1)))
```

```
                ),
11              R = list( n = 20,
                  generador = "runiforme",
13                params = list(a = -5, b = 0, c = -5, d = 0)
                  )
15              )
  escenario <- generarDatos(grupos)
17 tecnicas <- list(
                  km = list(
19                  funcion = "kmr",
                    params = list(n.grupos = 2,
21                                funcionDistancia = "dEuclidea", trim = 1/10),
                    resultado = "cluster", datos = TRUE
23                  ),
                  kmr_dRob = list(
25                  funcion = "kmr",
                    params = list(n.grupos = 2,
27                                funcionDistancia = "dRobusta", trim = 1/10),
                    resultado = "cluster", datos = TRUE
29                  ),
                  km3 = list(
31                  funcion = "kmr",
                    params = list(n.grupos = 3,
33                                funcionDistancia = "dEuclidea", trim = 0),
                    resultado = "cluster", datos = TRUE
35                  )
                  )
37
  rs1 <- simular(150, grupos, tecnicas)
39
  rs1$escenario$clv <- clasificacionVerdadera(rs1$escenario)
41 dd <- extraerDatos(rs1, obsTodas)
  dd <- within(dd, {bien <- as.factor(bien)})
43 dmc <- extraerDatos(rs2, obsMalClasificadas)
  mm0 <- evaluarMedida(rs2, porcentajeAcierto)
45 mm2 <- evaluarMedida(rs2, especificidad)
  mm3 <- evaluarMedida(rs2, sensibilidad)
47 mm4 <- evaluarMedida(rs2, ruidoMalClasificado)
  mm1 <- evaluarMedida(rs2, porcentajeError)
49 mm5 <- evaluarMedida(rs2, obsClasificadasRuido)
  mm6 <- evaluarMedida(rs2, gruposAlmenosUno)
51 mm7 <- evaluarMedida(rs2, gruposTodosBien)
```

```
1 grupos <- list(
                A = list( n = 100,
3                 generador = "rmvnorm",
                  params = list(mean = c( 0, 4),
5                               sigma = matrix(c(1, 0, 0, 1), ncol = 2))
                  ),
7               B = list( n = 100,
                  generador = "rmvnorm",
9                 params = list(mean = c(4, 0), sigma = diag(c(1, 1)))
                  ),
11              R = list( n = 20,
```

```
                generador = "runiforme",
                params = list(a = -10, b = 10, c = -10, d = 10)
                )
              )
escenario <- generarDatos(grupos)
tecnicas <- list(
                km = list(
                  funcion = "kmr",
                  params = list(n.grupos = 2,
                                funcionDistancia = "dEuclidea", trim = 1/10),
                  resultado = "cluster", datos = TRUE
                  ),
                kmr_dRob = list(
                  funcion = "kmr",
                  params = list(n.grupos = 2,
                                funcionDistancia = "dRobusta", trim = 1/10),
                  resultado = "cluster", datos = TRUE
                  ),
                km3 = list(
                  funcion = "kmr",
                  params = list(n.grupos = 3,
                                funcionDistancia = "dEuclidea", trim = 0),
                  resultado = "cluster", datos = TRUE
                  )
                )

rs3 <- simular(150, grupos, tecnicas)

rs3$escenario$clv <- clasificacionVerdadera(rs3$escenario)
dd <- extraerDatos(rs3, obsTodas)
dd <- within(dd, {bien <- as.factor(bien)})
dmc <- extraerDatos(rs3, obsMalClasificadas)
mm0 <- evaluarMedida(rs3, porcentajeAcierto)
mm2 <- evaluarMedida(rs3, especificidad)
mm3 <- evaluarMedida(rs3, sensibilidad)
mm4 <- evaluarMedida(rs3, ruidoMalClasificado)
mm1 <- evaluarMedida(rs3, porcentajeError)
mm5 <- evaluarMedida(rs3, obsClasificadasRuido)
mm6 <- evaluarMedida(rs3, gruposAlmenosUno)
mm7 <- evaluarMedida(rs3, gruposTodosBien)
```

```
grupos <- list(
              A = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c( 0, 4),
                              sigma = matrix(c(1, 0, 0, 1), ncol = 2))
                ),
              B = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(4, 0), sigma = diag(c(1, 1)))
                ),
              R = list( n = 20,
                generador = "runiforme",
                params = list(a = 0.5, b = 3.5, c = 0.5, d = 3.5)
```

```
                )
              )
escenario ← generarDatos(grupos)
tecnicas ← list(
                km = list(
                  funcion = "kmr",
                  params = list(n.grupos = 2,
                                funcionDistancia = "dEuclidea", trim = 1/10),
                  resultado = "cluster", datos = TRUE
                  ),
                kmr_dRob = list(
                  funcion = "kmr",
                  params = list(n.grupos = 2,
                                funcionDistancia = "dRobusta", trim = 1/10),
                  resultado = "cluster", datos = TRUE
                  ),
                km3 = list(
                  funcion = "kmr",
                  params = list(n.grupos = 3,
                                funcionDistancia = "dEuclidea", trim = 0),
                  resultado = "cluster", datos = TRUE
                  )
                )

rs2 ← simular(150, grupos, tecnicas)

rs2$escenario$clv ← clasificacionVerdadera(rs2$escenario)
dd ← extraerDatos(rs2, obsTodas)
dd ← within(dd, {bien ← as.factor(bien)})
dmc ← extraerDatos(rs2, obsMalClasificadas)
mm0 ← evaluarMedida(rs2, porcentajeAcierto)
mm2 ← evaluarMedida(rs2, especificidad)
mm3 ← evaluarMedida(rs2, sensibilidad)
mm4 ← evaluarMedida(rs2, ruidoMalClasificado)
mm1 ← evaluarMedida(rs2, porcentajeError)
mm5 ← evaluarMedida(rs2, obsClasificadasRuido)
mm6 ← evaluarMedida(rs2, gruposAlmenosUno)
mm7 ← evaluarMedida(rs2, gruposTodosBien)
```

E.4. Capítulo 4

E.4.1. Simulaciones

```
grupos ← list(
              A = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c( 0, 3),
                              sigma = matrix(c(2, 0.5, 0.5, 0.5), ncol = 2))
                ),
              B = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(3, 0), sigma = diag(c(2, 2)))
                ),
```

```
              C = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(-3, 0),
                        sigma = matrix(c(2, -0.5, -0.5, 5), ncol = 2))
                ),
              R = list( n = 50,
                generador = "runiforme",
                params = list(a = -10, b = 10, c = -10, d = 10)
                )
              )
escenario <- generarDatos(grupos)
tecnicas <- list(
                mixN_3g= list(
                  funcion = "emmix",
                  params = list(grs = 3, cov.op = 4,
                                usar.t = FALSE, trim = 50/300),
                  resultado = "cluster", datos = TRUE
                  ),
                mixTgl4_3g = list(
                  funcion = "emmix",
                  params = list(grs = 3, cov.op = 4,
                                usar.t = TRUE, dof = 4, trim = 50/300),
                  resultado = "cluster", datos = TRUE
                  ),
                mixTgl12_3g = list(
                  funcion = "emmix",
                  params = list(grs = 3, cov.op = 4,
                                usar.t = TRUE, dof = 12, trim = 50/300),
                  resultado = "cluster", datos = TRUE
                  )
                )

rs1 <- simular(150, grupos, tecnicas)

rs1$escenario$clv <- clasificacionVerdadera(rs1$escenario)
dd <- extraerDatos(rs1, obsTodas)
dd <- within(dd, {bien <- as.factor(bien)})
dmc <- extraerDatos(rs2, obsMalClasificadas)
mm0 <- evaluarMedida(rs2, porcentajeAcierto)
mm2 <- evaluarMedida(rs2, especificidad)
mm3 <- evaluarMedida(rs2, sensibilidad)
mm4 <- evaluarMedida(rs2, ruidoMalClasificado)
mm1 <- evaluarMedida(rs2, porcentajeError)
mm5 <- evaluarMedida(rs2, obsClasificadasRuido)
mm6 <- evaluarMedida(rs2, gruposAlmenosUno)
mm7 <- evaluarMedida(rs2, gruposTodosBien)
```

E.5. Capítulo 5

E.5.1. Simulaciones

```
grupos <- list(
              A = list( n = 100,
```

```
                generador = "rmvnorm",
                params = list(mean = c( 0, 3),
                              sigma = matrix(c(2, 0.5, 0.5, 0.5), ncol = 2))
                ),
              B = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(3, 0), sigma = diag(c(2, 2)))
                ),
              C = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(-3, 0),
                        sigma = matrix(c(2, -0.5, -0.5, 5), ncol = 2))
                ),
              R = list( n = 50,
                generador = "runiforme",
                params = list(a = -10, b = 10, c = -10, d = 10)
                )
              )
escenario <- generarDatos(grupos)
tecnicas <- list(
                mixN_3g= list(
                  funcion = "emmix",
                  params = list(grs = 3, cov.op = 4,
                                usar.t = FALSE, trim = 50/300),
                  resultado = "cluster", datos = TRUE
                  ),
                tc_3g = list(
                  funcion = "tclust",
                  params = list(K = 3,
                                alpha = 50/300, factor = 50, niter = 10,
                                Ksteps = 10),
                  resultado = "asig", datos = TRUE
                  )
                )

rs1 <- simular(150, grupos, tecnicas)

rs1$escenario$clv <- clasificacionVerdadera(rs1$escenario)
dd <- extraerDatos(rs1, obsTodas)
dd <- within(dd, {bien <- as.factor(bien)})
dmc <- extraerDatos(rs2, obsMalClasificadas)
mm0 <- evaluarMedida(rs2, porcentajeAcierto)
mm2 <- evaluarMedida(rs2, especificidad)
mm3 <- evaluarMedida(rs2, sensibilidad)
mm4 <- evaluarMedida(rs2, ruidoMalClasificado)
mm1 <- evaluarMedida(rs2, porcentajeError)
mm5 <- evaluarMedida(rs2, obsClasificadasRuido)
mm6 <- evaluarMedida(rs2, gruposAlmenosUno)
mm7 <- evaluarMedida(rs2, gruposTodosBien)
```

E.6. Capítulo 6

E.6.1. Escenario 1

```
grupos ← list(
              A = list( n = 150,
                generador = "rmvnorm",
                params = list(mean = c( 0, 3),
                              sigma = matrix(c(2, 0.5, 0.5, 0.5), ncol = 2))
                ),
              B = list( n = 150,
                generador = "rmvnorm",
                params = list(mean = c(3, 0), sigma = diag(c(2, 2)))
                ),
              C = list( n = 150,
                generador = "rmvnorm",
                params = list(mean = c(-3, 0),
                        sigma = matrix(c(2, -0.5, -0.5, 5), ncol = 2))
                ),
              R = list( n = 50,
                generador = "runiforme",
                params = list(a = -10, b = 10, c = -10, d = 10)
                )
              )
escenario ← generarDatos(grupos)
tecnicas ← list(
                kmr_dRob_3g = list(
                  funcion = "kmr",
                  params = list(n.grupos = 3,
                                funcionDistancia = "dRobusta", trim = 50/500),
                  resultado = "cluster", datos = TRUE
                  ),
                mixTgl4_3g = list(
                  funcion = "emmix",
                  params = list(grs = 3, cov.op = 4,
                                usar.t = TRUE, dof = 4, trim = 50/500),
                  resultado = "cluster", datos = TRUE
                  ),
                tc_3g = list(
                  funcion = "tclust",
                  params = list(K = 3,
                                alpha = 50/500, factor = 50, niter = 10,
                                Ksteps = 10),
                  resultado = "asig", datos = TRUE
                  )
                )
rs1 ← simular(150, grupos, tecnicas)

rs1$escenario$clv ← clasificacionVerdadera(rs1$escenario)
dd ← extraerDatos(rs1, obsTodas)
dd ← within(dd, {bien ← as.factor(bien)})
dmc ← extraerDatos(rs2, obsMalClasificadas)
mm0 ← evaluarMedida(rs2, porcentajeAcierto)
```

```
mm2 ← evaluarMedida(rs2, especificidad)
mm3 ← evaluarMedida(rs2, sensibilidad)
mm4 ← evaluarMedida(rs2, ruidoMalClasificado)
mm1 ← evaluarMedida(rs2, porcentajeError)
mm5 ← evaluarMedida(rs2, obsClasificadasRuido)
mm6 ← evaluarMedida(rs2, gruposAlmenosUno)
mm7 ← evaluarMedida(rs2, gruposTodosBien)
```

E.6.2. Escenario 2

```
grupos ← list(
             A = list( n = 100,
               generador = "rmvnorm",
               params = list(mean = c( 0, 3),
                             sigma = matrix(c(2, 0.5, 0.5, 0.5), ncol = 2))
               ),
             B = list( n = 100,
               generador = "rmvnorm",
               params = list(mean = c(3, 0), sigma = diag(c(2, 2)))
               ),
             C = list( n = 100,
               generador = "rmvnorm",
               params = list(mean = c(-3, 0),
                       sigma = matrix(c(2, -0.5, -0.5, 5), ncol = 2))
               ),
             R = list( n = 100,
               generador = "runiforme",
               params = list(a = -10, b = 10, c = -10, d = 10)
               )
             )
escenario ← generarDatos(grupos)
tecnicas ← list(
               kmr_dRob_3g = list(
                 funcion = "kmr",
                 params = list(n.grupos = 3,
                             funcionDistancia = "dRobusta", trim = 100/400),
                 resultado = "cluster", datos = TRUE
                 ),
               mixTgl4_3g = list(
                 funcion = "emmix",
                 params = list(grs = 3, cov.op = 4,
                             usar.t = TRUE, dof = 4, trim = 100/400),
                 resultado = "cluster", datos = TRUE
                 ),
               tc_3g = list(
                 funcion = "tclust",
                 params = list(K = 3,
                             alpha = 100/400, factor = 50, niter = 10,
                             Ksteps = 10),
                 resultado = "asig", datos = TRUE
                 )
               )

rs1 ← simular(150, grupos, tecnicas)
```

```
rs1$escenario$clv ← clasificacionVerdadera(rs1$escenario)
dd ← extraerDatos(rs1, obsTodas)
dd ← within(dd, {bien ← as.factor(bien)})
dmc ← extraerDatos(rs2, obsMalClasificadas)
mm0 ← evaluarMedida(rs2, porcentajeAcierto)
mm2 ← evaluarMedida(rs2, especificidad)
mm3 ← evaluarMedida(rs2, sensibilidad)
mm4 ← evaluarMedida(rs2, ruidoMalClasificado)
mm1 ← evaluarMedida(rs2, porcentajeError)
mm5 ← evaluarMedida(rs2, obsClasificadasRuido)
mm6 ← evaluarMedida(rs2, gruposAlmenosUno)
mm7 ← evaluarMedida(rs2, gruposTodosBien)
```

E.6.3. Escenario 3

```
grupos ← list(
              A = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c( 0, 3),
                              sigma = matrix(c(2, 0.5, 0.5, 0.5), ncol = 2))
                ),
              B = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(3, 0), sigma = diag(c(2, 2)))
                ),
              C = list( n = 100,
                generador = "rmvnorm",
                params = list(mean = c(-3, 0),
                        sigma = matrix(c(2, -0.5, -0.5, 5), ncol = 2))
                ),
              R = list( n = 100,
                generador = "runiforme",
                params = list(a = 0, b = 10, c = 0, d = 10)
                )
              )
escenario ← generarDatos(grupos)
tecnicas ← list(
                kmr_dRob_3g = list(
                  funcion = "kmr",
                  params = list(n.grupos = 3,
                                funcionDistancia = "dRobusta", trim = 100/400),
                  resultado = "cluster", datos = TRUE
                  ),
                mixTgl4_3g = list(
                  funcion = "emmix",
                  params = list(grs = 3, cov.op = 4,
                                usar.t = TRUE, dof = 4, trim = 100/400),
                  resultado = "cluster", datos = TRUE
                  ),
                tc_3g = list(
                  funcion = "tclust",
                  params = list(K = 3,
                                alpha = 100/400, factor = 50, niter = 10,
```

```
                              Ksteps = 10),
                resultado = "asig", datos = TRUE
                )
              )

rs1 <- simular(150, grupos, tecnicas)

rs1$escenario$clv <- clasificacionVerdadera(rs1$escenario)
dd <- extraerDatos(rs1, obsTodas)
dd <- within(dd, {bien <- as.factor(bien)})
dmc <- extraerDatos(rs2, obsMalClasificadas)
mm0 <- evaluarMedida(rs2, porcentajeAcierto)
mm2 <- evaluarMedida(rs2, especificidad)
mm3 <- evaluarMedida(rs2, sensibilidad)
mm4 <- evaluarMedida(rs2, ruidoMalClasificado)
mm1 <- evaluarMedida(rs2, porcentajeError)
mm5 <- evaluarMedida(rs2, obsClasificadasRuido)
mm6 <- evaluarMedida(rs2, gruposAlmenosUno)
mm7 <- evaluarMedida(rs2, gruposTodosBien)
```

E.6.4. Escenario 4

```
grupos <- list(
            A = list( n = 150,
              generador = "rmvnorm",
              params = list(mean = c( 0, 3),
                            sigma = matrix(c(2, 0.5, 0.5, 0.5), ncol = 2))
              ),
            B = list( n = 200,
              generador = "rmvnorm",
              params = list(mean = c(3, 0), sigma = diag(c(2, 2)))
              ),
            C = list( n = 100,
              generador = "rmvnorm",
              params = list(mean = c(-3, 0),
                      sigma = matrix(c(2, -0.5, -0.5, 5), ncol = 2))
              ),
            R = list( n = 50,
              generador = "runiforme",
              params = list(a = -10, b = 10, c = -10, d = 10)
              )
            )
escenario <- generarDatos(grupos)
tecnicas <- list(
              kmr_dRob_3g = list(
                funcion = "kmr",
                params = list(n.grupos = 3,
                              funcionDistancia = "dRobusta", trim = 50/500),
                resultado = "cluster", datos = TRUE
                ),
              mixTgl4_3g = list(
                funcion = "emmix",
                params = list(grs = 3, cov.op = 4,
                              usar.t = TRUE, dof = 4, trim = 50/500),
```

```
                  resultado = "cluster", datos = TRUE
                  ),
                tc_3g = list(
                  funcion = "tclust",
                  params = list(K = 3,
                                alpha = 50/500, factor = 50, niter = 10,
                                Ksteps = 10),
                  resultado = "asig", datos = TRUE
                  )
                )

rs1 <- simular(150, grupos, tecnicas)

rs1$escenario$clv <- clasificacionVerdadera(rs1$escenario)
dd <- extraerDatos(rs1, obsTodas)
dd <- within(dd, {bien <- as.factor(bien)})
dmc <- extraerDatos(rs2, obsMalClasificadas)
mm0 <- evaluarMedida(rs2, porcentajeAcierto)
mm2 <- evaluarMedida(rs2, especificidad)
mm3 <- evaluarMedida(rs2, sensibilidad)
mm4 <- evaluarMedida(rs2, ruidoMalClasificado)
mm1 <- evaluarMedida(rs2, porcentajeError)
mm5 <- evaluarMedida(rs2, obsClasificadasRuido)
mm6 <- evaluarMedida(rs2, gruposAlmenosUno)
mm7 <- evaluarMedida(rs2, gruposTodosBien)
```

E.6.5. Escenario 5

```
grupos <- list(
              A = list( n = 150,
                generador = "rmvnorm",
                params = list(mean = c( 0, 3),
                              sigma = matrix(c(2, 0.5, 0.5, 0.5), ncol = 2))
                ),
              B = list( n = 150,
                generador = "rmvnorm",
                params = list(mean = c(3, 0), sigma = diag(c(2, 2)))
                ),
              C = list( n = 150,
                generador = "rmvnorm",
                params = list(mean = c(-3, 0),
                        sigma = matrix(c(2, -0.5, -0.5, 5), ncol = 2))
                ),
              R = list( n = 50,
                generador = "runiforme2",
                params = list(a = -10, b = 10, c = -10, d = 10)
                )
              )
escenario <- generarDatos(grupos)
tecnicas <- list(
                kmr_dRob_3g = list(
                  funcion = "kmr",
                  params = list(n.grupos = 3,
                                funcionDistancia = "dRobusta", trim = 50/500),
```

```
                  resultado = "cluster", datos = TRUE
                  ),
                mixTgl4_3g = list(
                  funcion = "emmix",
                  params = list(grs = 3, cov.op = 4,
                                usar.t = TRUE, dof = 4, trim = 50/500),
                  resultado = "cluster", datos = TRUE
                  ),
                tc_3g = list(
                  funcion = "tclust",
                  params = list(K = 3,
                                alpha = 50/500, factor = 50, niter = 10,
                                Ksteps = 10),
                  resultado = "asig", datos = TRUE
                  )
                )

rs1 <- simular(150, grupos, tecnicas)

rs1$escenario$clv <- clasificacionVerdadera(rs1$escenario)
dd <- extraerDatos(rs1, obsTodas)
dd <- within(dd, {bien <- as.factor(bien)})
dmc <- extraerDatos(rs2, obsMalClasificadas)
mm0 <- evaluarMedida(rs2, porcentajeAcierto)
mm2 <- evaluarMedida(rs2, especificidad)
mm3 <- evaluarMedida(rs2, sensibilidad)
mm4 <- evaluarMedida(rs2, ruidoMalClasificado)
mm1 <- evaluarMedida(rs2, porcentajeError)
mm5 <- evaluarMedida(rs2, obsClasificadasRuido)
mm6 <- evaluarMedida(rs2, gruposAlmenosUno)
mm7 <- evaluarMedida(rs2, gruposTodosBien)
```

E.7. Capítulo 7

```
casas <- read.csv("casas_inmobiliaria2.csv", header = TRUE, sep = ",")
cg1 <- casas
cg1$precio <- cg1$precio / 1000
cg <- melt(cg1, c("id", "zona"))
levels(cg$variable) <- c("Precio", "Const.", "Terreno")

r2 <- ddply(cg1, .(zona), summarise,
                media_p = mean(precio), sd_p = sd(precio),
                media_m2t = mean(m2terreno), sd_m2t = sd(m2terreno),
                media_m2c = mean(m2cons), sd_m2c = sd(m2cons))
r1 <- ddply(cg1, .(zona), nrow)
resumen <- cbind(r1, r2[, 2:7])
resumen

## TCLUST ##

c1 <- casas[, -c(1, 2)]
c2 <- sapply(c1, scale)

```

```
tc1 <- tclust(as.matrix(c2), 4, 0.10, 90)
cc1 <- data.frame(casas, "cluster" = factor(tc1$asig, levels = 0:4,
                          labels = c("R", "A", "B", "C", "D")))
ols.tclust <- cg1[cc1$cluster == "R", ]
ols.tclust$cluster <- rep("R", nrow(ols.tclust))
o.tclust <- data.frame(cc1, tecnica = rep("TCLUST", nrow(cc1)))

## MIX T ##

c1 <- casas[, -c(1, 2)]
c2 <- sapply(c1, scale)

mt1 <- emmix(as.matrix(c2), 4, cov.op = 4, usar.t = TRUE, dof = 4, trim = 0.10)
mtc1 <- data.frame(casas, "cluster" = factor(mt1$cluster, levels = 1:5,
                          labels = c("A", "B", "C", "D", "R")))
ols.mixt <- casas[mtc1$cluster == "R", ]
ols.mixt$cluster <- rep("R", nrow(ols.mixt))
ols.mixt$tecnica <- rep("MIXT", nrow(ols.mixt))
o.mixt <- data.frame(mtc1, tecnica = rep("EMMIX", nrow(mtc1)))

## KMR ##

c1 <- casas[, -c(1, 2)]
c2 <- sapply(c1, scale)

kmr1 <- kmr(as.matrix(c2[, 2:3]), 4, funcionDistancia = "dRobusta", trim = 0.1)
cc1 <- data.frame(casas, "cluster" = factor(kmr1$cluster, levels = 1:5,
                          labels = c("A", "B", "C", "D", "R")))
ols.kmr <- casas[cc1$cluster == "R", ]
ols.kmr$cluster <- rep("R", nrow(ols.kmr))
ols.kmr$tecnica <- rep("KMR", nrow(ols.kmr))
o.kmr <- data.frame(cc1, tecnica = rep("KMR", nrow(cc1)))
```

Apéndice F

Conjunto de Datos Utilizado

ID	Zona	Precio (Miles)	Metros Const.	Metros del Terreno
23	Parque Miramar	125	170	600
92	Carrasco	205	268	851
91	Carrasco	200	150	525
90	Carrasco	175	175	250
89	Carrasco	115	210	430
95	Punta Gorda	153	138	356
96	Carrasco	270	180	409
141	Punta Gorda	270	390	840
97	Carrasco	310	293	560
98	Carrasco	400	250	530
99	Carrasco	350	350	740
100	Carrasco	370	283	792
101	Carrasco	355	280	812
102	Carrasco	400	250	450
104	Carrasco	440	340	800
105	Carrasco	550	319	1050
106	Carrasco	500	450	950
107	Carrasco	500	420	1572
108	Carrasco	600	500	1200
109	Carrasco	720	380	1400
110	Carrasco	1500	2000	4700
111	Carrasco Norte	70	95	150
112	Malvín	340	340	1070
113	Carrasco Norte	105	125	750
114	Carrasco Norte	130	200	684
115	Carrasco Norte	130	140	300
116	Carrasco Norte	130	244	2000
117	Carrasco Norte	140	180	550
118	Carrasco Norte	145	197	755

119	Carrasco Norte	140	140	2500
120	Carrasco Norte	195	180	270
121	Carrasco Norte	155	238	674
125	Punta Gorda	115	165	800
126	Punta Gorda	145	212	525
129	Punta Gorda	155	150	470
130	Punta Gorda	180	115	570
131	Punta Gorda	185	200	515
132	Punta Gorda	210	215	500
133	Punta Gorda	235	239	415
134	Punta Gorda	225	150	620
136	Punta Gorda	250	290	600
137	Punta Gorda	250	220	425
138	Punta Gorda	252	230	774
139	Punta Gorda	260	279	860
140	Punta Gorda	270	350	600
142	Punta Gorda	285	250	400
144	Punta Gorda	300	330	650
145	Punta Gorda	300	240	422
146	Punta Gorda	320	340	835
147	Punta Gorda	320	295	780
148	Punta Gorda	330	212	616
149	Punta Gorda	370	352	970
150	Punta Gorda	850	900	1594
152	Carrasco Norte	155	238	674
153	Carrasco	89	80	210
154	Carrasco	125	210	430
156	Malvín	150	205	411
157	Malvín	195	200	325
158	Malvín	190	284	341
160	Malvín	210	297	660
161	Malvín	270	345	780
162	Parque Miramar	102	100	437
163	Malvín	240	340	520
164	Parque Miramar	148	115	220
165	Parque Miramar	160	125	502
166	Parque Miramar	158	195	492
168	Parque Miramar	170	260	502
169	Parque Miramar	140	138	730
170	Parque Miramar	235	340	1000
171	Parque Miramar	90	80	200
494	Carrasco	230	180	773
207	Punta Gorda	95	165	480
211	Parque Miramar	62	90	500
208	Carrasco Norte	115	180	975

210	Parque Miramar	120	130	500
215	Barra de Carrasco	65	143	620
216	Shangrilá	85	110	817
217	Punta Gorda	130	110	120
218	Punta Gorda	205	290	640
226	Carrasco Norte	120	100	525
220	Malvín	145	212	525
221	Punta Gorda	240	210	585
223	Punta Gorda	250	155	365
224	Punta Gorda	260	250	330
225	Carrasco Norte	158	180	700
227	Carrasco Norte	105	125	750
228	Carrasco Norte	135	130	500
233	Carrasco	240	245	1117
234	Parque Miramar	250	338	700
235	Parque Miramar	220	200	503
236	Punta Gorda	390	341	470
237	Parque Miramar	180	310	1370
238	Parque Miramar	185	300	520
239	Parque Miramar	79	145	500
240	Carrasco Norte	145	160	482
241	Parque Miramar	79	122	430
243	Parque Miramar	140	120	500
244	Parque Miramar	180	150	650
246	Parque Miramar	215	170	550
247	Parque Miramar	120	130	500
248	Parque Miramar	185	200	1200
250	Carrasco	150	115	496
251	Carrasco	180	150	590
252	Carrasco	185	226	555
253	Carrasco	220	220	980
254	Carrasco	330	300	570
255	Carrasco	320	260	439
256	Carrasco	395	250	800
257	Carrasco	365	300	530
258	Carrasco	500	400	1420
259	Carrasco	500	400	1170
260	Carrasco	600	500	1200
261	Punta Gorda	175	190	580
262	Punta Gorda	110	90	280
263	Punta Gorda	180	280	730
264	Punta Gorda	240	350	1028
265	Punta Gorda	210	210	408
266	Punta Gorda	210	190	415
267	Malvín	230	530	949

268	Punta Gorda	800	500	1594
269	Malvín	105	95	175
270	Malvín	110	90	400
271	Parque Miramar	90	130	710
272	Carrasco	200	165	350
273	Carrasco	260	260	631
274	Carrasco	465	344	862
275	Punta Gorda	225	290	640
276	Barra de Carrasco	69	110	460
279	Parque Miramar	2000	583	6038
281	Carrasco	185	185	280
282	Barra de Carrasco	125	210	700
283	Barra de Carrasco	85	190	560
284	Barra de Carrasco	205	236	882
285	Carrasco Norte	110	130	750
286	Parque Miramar	150	400	1000
287	Carrasco	1000	500	1899
289	Carrasco	100	80	210
290	Malvín	270	275	560
291	Carrasco Norte	550	680	3200
292	Parque Miramar	290	240	560
293	Parque Miramar	120	160	435
294	Parque Miramar	130	150	1000
296	Parque Miramar	75	110	500
299	Punta Gorda	200	250	500
300	Punta Gorda	260	250	749
301	Punta Gorda	235	220	500
304	Parque Miramar	95	110	400
306	Parque Miramar	140	140	645
322	Parque Miramar	150	300	1000
324	Parque Miramar	80	100	450
325	Parque Miramar	180	220	750
326	Punta Gorda	140	332	500
327	Carrasco	500	500	971
336	Carrasco Norte	320	500	1100
345	Carrasco Norte	110	125	539
347	Carrasco	155	127	544
349	Carrasco Norte	90	220	900
350	Parque Miramar	215	180	780
351	Parque Miramar	420	320	750
352	Barra de Carrasco	97	100	750
353	Barra de Carrasco	97	100	750
354	Carrasco Norte	90	220	800
355	Carrasco	400	286	500
357	Shangrilá	150	150	1300

362	Parque Miramar	125	180	700
364	Shangrilá	55	135	700
365	Shangrilá	150	200	1100
368	Shangrilá	30	113	600
369	Carrasco	360	300	870
370	Parque Miramar	95	180	500
371	Parque Miramar	120	130	560
382	Carrasco Norte	149	130	546
386	Carrasco	300	220	440
388	Parque Miramar	150	220	550
390	Parque Miramar	420	450	2600
399	Barra de Carrasco	150	170	400
402	Barra de Carrasco	153	160	980
404	Barra de Carrasco	145	160	500
405	Carrasco	650	270	900
408	Carrasco	320	230	405
419	Carrasco	125	90	773
442	Parque Miramar	165	240	500
447	Parque Miramar	145	140	512
448	Parque Miramar	145	150	506
458	Carrasco	125	220	250
459	Carrasco	190	200	500
465	Parque Miramar	140	280	750
470	Carrasco	270	140	360
482	Parque Miramar	100	110	502
495	Carrasco	350	221	1350

Referencias bibliográficas

[1] Graciela Boente, Ricardo Fraiman, and Víctor Yohai. Qualitative robustness for stochastic processes. *The Annals of Statistics*, 15(3):129–1312, 1987.

[2] N. Day. Estimating the components of a mixture of two normal distributions. *Biometrika*, 56(3):463–474, 1969.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[4] D.L Donoho. Breakdown properties of multivariate location estimators. *Qualifying paper, Harvard University, Boston*, 1982.

[5] Katrien Van Driessen and P.J. Rousseeuw. A fast algorithm for the minimum covariance. *Technometrics*, 41:212–223, 1999.

[6] Richard L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

[7] Bernhard Flury. *A First Course in Multivariate Statistics*. Springer, 1997.

[8] J. Friedman and J.W Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 9:881–890, 1974.

[9] María Teresa Gallegos. A survey of sampling from contaminated distributions. In *Classification, Clustering and Data Analysis: Recent advances and application*, pages 247–255. K. Jajuga, A.Sokolowski, and H.H. Bock eds, 2002.

[10] Daniel Gervini. The influence function of the stahel-donoho estimator of multivariate location and scatter. *Statistics & Probability Letters*, 60(4):425–435, 2002.

[11] A. Gordaliza, C. Matrán, J. A. Cuesta-Albertos, and A. Mayo-Iscar. Exploring the number of groups in robust model-based clustering. *Statistics and Computing*, 20:1–29, 2010.

[12] Gentiane Haesbroeck and Christophe Croux. Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. *Journal of Multivariate Analysis*, 71:161–190, 1998.

[13] R. Frank Hampel. A general qualitative definition of robustness. *The Annals of Mathematical Statistics*, 42(6):1887–1896, 1971.

[14] R. Frank Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

[15] Peter E. Hart, David G. Stork, and Richard O. Duda. *Pattern classification.* Wiley, 2001.

[16] J.A. Hartigan. Asymptotic distributions for clustering criteria. *Annals of Statistics*, 6:117–131, 1978.

[17] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[18] Mia Hubert and Michiel Debruyne. Breakdown value. *WIREs Computational Statistics*, 1:296–302, Noviembre / Diciembre 2009.

[19] K. Jajuga. Model-based clustering: Discussion on some approaches. In *Data Analysis and Decision Support*, pages 73–81. Springer, 2005.

[20] Picek Jan and Jana Jurecková. *Robust Statistical Methods with R.* Chapman & Hall/CRC, 2006.

[21] T. Krishnan and G. J. McLachlan. *The EM Algorithm and Extensions.* John Wiley & Sons, 1997.

[22] Francesca Marta Di Lascio. Analyzing the dependence structure of microarray data: a copula based approach. In *Tesis de Doctorado*, Universidad de Bologna, 2008.

[23] Ricardo R. Douglas Martin, Víctor, and Yohai Maronna. *Robust Statistics: Theory and Methods.* John Wiley & Sons, 2006.

[24] Carlos Matrán, Agustín Mayo-Iscar, and Juan A Cuesta-Albertos. Trimming and likelihood: Robust location and dispersion estiamtion in the elliptical model. *The Annals of Statistics*, 36(5):2284–2318, 2008.

[25] McQueen. Soma methods for classification and analysis of multivariate observation. *Computer and Chemitry*, 4:257–272, 1967.

[26] Volodymyr Melnykov. Finite mixture models and model-based clustering. *Statistics Surveys*, 4:80–116, 2010.

[27] Beatriz V. M. Mendes, Eduardo F. L. De Melo, and Roger B. Nelsen. Models, copulas and applications. *Communications in Statistics Simulation and Computation*, 36:997–1017, 2007.

[28] Shu-Kay Ng, Geoffrey J. McLachlan, and Richard Bean. Robust cluster analysis via mixture models. *Australian Journal of Statistical*, 2(3):157–174, 2006.

[29] K. Pearson. Contribution to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society*, 185:71–110, 1894.

[30] D. Peel and G. J. McLachlan. *Finite Mixture Model*. Wiley, 2000.

[31] D. Peel and G. J. McLachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 10:339–348, 2000.

[32] J. Peter, E. Ronchetti, and M. Huber. *Robust Statistics*. John Wiley & Sons, segunda edición edition, 2009.

[33] David Pollard. Strong consistency of k-means clustering. *Annals of Probability*, 9(1):135–140, 1981.

[34] David Pollard. A central limit theorem for k-means clustering. *Annals of Probability*, 10(4):919–926, 1982.

[35] F.J. Prieto and D. Peña. Multivariate outlier detection and robust covariance estimation. *Technometrics*, 41:286–300, 2001.

[36] Yuri V. Prokhorov. Convergence of random processes and limit theorems in probability theory. *Theory of Probability and its Applications*, 1(2):157–214, 1956.

[37] Marcos Raydan and Ernesto G. Birgin. Robust stopping criteria for dykstra's algorithm. *SIAM Journal on Scientic Computing*, 26(6):1405–1414, 2004.

[38] Gunter Ritter and María Teresa Gallegos. A robust method for cluster analysis. *Annals of Statistics*, 33:347–380, 2005.

[39] Gunter Ritter and María Teresa Gallegos. Trimmed ml estimation of contaminated mixtures. *The Indian Journal of Statistics*, 71-A(2):164–220, 2009.

[40] P.J. Rousseeuw. Least median of squares regression. *Am Stat Assoc*, 79:871–880, 1984.

[41] P.J. Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical Statistics and Applications*, B(1):283–297, 1985.

[42] P.J. Rousseeuw and H.P Lopuhaa. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, 19(1):229–248, 1991.

[43] Volker Strassen. The existence of probability measures with given marginals. *The Annals of Mathematical Statistics*, 36(2):423–439, 1965.

[44] John Wilder Tukey. A survey of sampling from contaminated distributions. In *Contributions to Probability and Statistics*, Los Alamos, New Mexico, 1960. Ingram Olkin,ed.

[45] John Wilder Tukey. Usable resistant/robust techinques of analysis. In *First ERDA Statistical Symposium*, Los Alamos, New Mexico, Noviembre 1975. Batelle Pacific Norhwest Laboratory.

[46] J. H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, 5:329–350, 1970.

[47] H. Ruben Zamar. Estimación robusta. *Estadística Española*, 36(137):327–387, 1994.