UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

PROYECTO FIN DE CARRERA

Validación de clusters basada en la negentropía de las particiones

JESÚS ARAGÓN NOVO

Abril 2015

Validación de clusters basada en la negentropía de las particiones

AUTOR: Jesús Aragón Novo
TUTOR: Luis F. Lago Fernández

Grupo de Neurocomputación Biológica (GNB)
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Abril de 2015

Resumen

Las técnicas de clustering se basan en la agrupación de una serie de puntos de acuerdo a un criterio de similitud, buscando que los puntos pertenecientes a un mismo cluster sean más similares entre si de lo que lo son con el resto de puntos.
El principal objetivo de este proyecto de fin de carrera es el estudio y evaluación de métodos de validación de clusters basados en la negentropía, así como su comparación con otros métodos más tradicionales.
Para ello se ha realizado un estudio del estado del arte, en el que se han evaluado diferentes métodos de clustering así como diferentes métodos de validación. La técnica de clustering que hemos utilizado en este proyecto se basa en ajustar a los datos una mezcla de gaussianas utilizando el algoritmo EM. Cada una de las gaussianas que contiene el modelo devuelto por éste se corresponde con un cluster.
A cada conjunto de datos se le realizan ajustes con diferente número de gaussianas, con lo que conseguimos tener modelos con diferente número de clusters. Los modelos devueltos por el algoritmo EM son evaluados mediante diferentes métodos de validación de clustering, los cuales nos dan una medida de la calidad de los diferentes modelos basándose en el criterio utilizado por cada método de validación. Entre estos métodos se encuentra el método objeto de análisis de este proyecto, Negentropy-based Validation (ΔJ), y dos ya establecidos en el contexto de las mezclas de distribuciones, AIC y BIC, con los que se realizarán las comparaciones.
Para la evaluación del método ΔJ se ha generado una batería de problemas sintéticos, escogiendo las variables que intervienen en cada problema de tal forma que al finalizar el análisis se han obtenido unos resultados que nos han permitido comparar el desempeño de los tres métodos en un rango muy amplio de situaciones.
Gracias al análisis realizado se ha llegado a las siguientes conclusiones: AIC tiene un funcionamiento muy negativo y ΔJ es un método que mejora el desempeño de BIC en la mayoría de los casos, planteándose como un fuerte candidato para su uso en aplicaciones con datos reales.
Parte de los resultados obtenidos en este estudio han sido publicados en una revista internacional (1).

Palabras Clave

Algoritmos de clustering, métodos de validación de clustering , Expectation-Maximization, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Negentropy-based Validation (ΔJ), Twonorm, Threenorm

Abstract

The clustering techniques are based on the grouping of a number of points according to a similarity criterion, looking forward to find in a cluster points more similar to each other than to the rest of the points.
The main objective of this final project at university is the study and evaluation of the clustering validation methods based on the negentrophy, and its comparison with other more traditional methods.
To that end, a study of "the state of the art" has been carried out, in which different clustering and validation methods have been evaluated.
The clustering technique which has been used in this project is based on adjusting a mixture of Gaussians to a dataset using the EM algorithm. Each of the Gaussians contained on the model returned by the algorithm corresponds to a cluster.
Every dataset is been adjust with different number of Gaussians, in order to obtain models with different number of clusters. The models that have been returned by the EM algorithm are evaluated with different clustering validation methods, which give us an approach to the quality of the different methods based on the criterion used by each validation method. Among these methods, we can find the one under study on this project, the Negentrophy-based Validation method (ΔJ), and two other methods already settled on the context of the distribution mixtures, the AIC and BIC methods, with which the comparisons will be make.
For the evaluation of the ΔJ method, a set of synthetic problems have been developed, choosing the variables involve in each problem so that, to the end of the analysis, the results obtained allow us to compare the performance of the three methods at a wide range of situations.
As the result of this analysis, the main conclusions obtained are: AIC has a very negative behavior and ΔJ is a method that improves the performance of BIC on most of the cases, emerging as a strong candidate for its use on real data applications.
Part of the results obtained on this study has been published on an international magazine (1).

Key words

Clustering algorithms , clustering validation methods, Expectation-Maximization, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Negentropy-based Validation (ΔJ), Twonorm, Threenorm

Agradecimientos

Quería agradecer en primer lugar a mi tutor Luis Fernando Lago por su dedicación, ayuda, apoyo y comprensión a lo largo de este proyecto, además de por todo el material que me ha facilitado para su desarrollo (artículos, acceso a los servidores, funciones necesarias para generar ciertos problemas, etc).
También quería agradecer a Manuel Sánchez Montañés, a Ana González Marcos y a Gonzalo Martínez Muñoz por la generación de los datos de acuerdo al procedimiento indicado en el artículo (2) para el análisis de los problemas Twonorm y Threenorm, y por su supervisión y guiado durante el proceso de estas pruebas.

También quería agradecer a todos los profesores que he tenido a lo largo de esta larga etapa como estudiante, pues cada uno a su manera ha aportado algo para que haya conseguido llegar hasta aquí.

A todos los amigos de la universidad con los que he sufrido y disfrutado este gran proceso formativo, de autoconocimiento y desarrollo personal, y a todos los que he conocido en esta época: Alfonso, Edu, Bader, Isa, Álvaro, Laura, Lucas y Juan. Gracias por haber sabido ayudarme a sobreponerme en los malos momentos y haber querido disfrutar conmigo los buenos.

A Amanda, una grandísima amiga que ha sabido aconsejarme de manera realmente objetiva cuando lo he necesitado. Es un orgullo tenerte en mi vida y sé que pase lo que pase siempre estarás ahí.

A Cris por conseguir hacerme feliz cada día cuando llego a casa y te veo, por apoyarme y ayudarme a terminar las cosas cuando parecen interminables, por conseguir crear un espacio nuestro que nada ni nadie puede perturbar, por mirarme como me miras, por haber construido junto a mí una familia "a nuestra manera", por hacer que cada momento juntos sea especial. ¡Te quiero!
Quería agradecer también a tu familia, por preocuparse por mí y hacerme sentir que también son mi familia.

Y por último y más importante, gracias a mi familia: a mis abuelos y a mi tía por haber cuidado de mi cuando ha sido necesario, por hacerme sentir tan querido y haber pasado junto a mi tanto tiempo desde pequeñito; a mi hermana, por ser un ejemplo de superación frente a las adversidades y todos los buenos momentos que hemos pasado juntos; y a mis padres por haberme transmitido unos valores de los que me siento orgulloso, por vuestra generosidad, amabilidad y buen humor, por toda la confianza, sonrisas y tiempo que nos habéis dedicado. Gracias por todo el amor que nos habéis ofrecido, por guiarnos y por haber antepuesto nuestro bienestar ante todo.
Como dicen la familia no se escoge, te toca. ¡Y no puedo estar más feliz con la mía!

Muchísimas gracias a todos.

INDICE DE CONTENIDOS

Resumen 3
Palabras Clave 3
Abstract 4
Key words 4
Agradecimientos 5
1 Introducción 11
1.1 Motivación 11
1.2 Objetivos 11
1.3 Organización de la memoria 12
2 Estado del arte 13
2.1 Introducción 13
2.2 Clasificación no supervisada (clustering) 14
2.2.1 Métodos paramétricos 14
2.2.1.1 Máxima verosimilitud 14
2.2.1.2 Expectation Maximization (EM) 15
2.2.2 Métodos no paramétricos 17
2.2.2.1 K-means 17
2.2.2.2 Clustering jerárquico 18
2.3 Validación de clusters 20
2.3.1 Introducción al problema 20
2.3.2 Akaike Information Criterion (AIC) 20
2.3.3 Bayesian Information Criterion (BIC) 21
2.3.4 Negentropy-based Validation (ΔJ) 21
2.4 Reducción de la dimensión 24
2.4.1 Principal Component Analysis 24
3 Diseño y Desarrollo 26
4 Pruebas y resultados en problemas sintéticos 28
4.1 Introducción 28
4.2 Análisis de problemas 28
4.2.1 Análisis en función de la separación de los clusters 28
4.2.1.1 Descripción del problema 28
4.2.1.2 Resultados y conclusiones 29
4.2.2 Análisis del algoritmo frente a la variación de la distancia de separación entre clusters y el grado de normalidad de los mismos. 34
4.2.2.1 Introducción 34
4.2.2.2 Descripción del problema 34
4.2.2.3 Resultados 37
4.2.2.4 Conclusiones 40
4.2.3 Análisis del algoritmo frente a la variación del número de puntos de los clusters y el grado de normalidad de los mismos. 41
4.2.3.1 Introducción 41
4.2.3.2 Descripción del problema 42
4.2.3.3 Resultados 44
4.2.3.1 Conclusiones 51
4.2.4 Análisis en función de la dimensión. 53
4.2.4.1 Introducción 53
4.2.4.2 Twonorm 53
4.2.4.3 Threenorm 57
5 Conclusiones y trabajo futuro 60

5.1 Conclusiones 60
5.2 Trabajo futuro 61
6 Bibliografía 63
Glosario 65
Anexos I
A Figuras obtenidas en el punto 4.2.2 para ΔJ_{US} I
B Figuras obtenidas en el punto 4.2.3 para ΔJ_{US} I
C Manual del programador I

INDICE DE FIGURAS

FIGURA 1. EJEMPLO DE DENDOGRAMA 20

FIGURA 2. PARTICIÓN DEL CONJUNTO DE DATOS EN K REGIONES NO SOLAPADAS 22

FIGURA 3. PCA 25

FIGURA 4. PROCESO SEGUIDO PARA LA RESOLUCIÓN DE LOS PROBLEMAS ABORDADOS 26

FIGURA 5. EJEMPLOS DE LOS DIFERENTES CLUSTERS UTILIZADOS EN ESTE PROBLEMA. DE IZQUIERDA A DERECHA Y DE ARRIBA ABAJO: UN CLUSTER GENERADO CON UNA DISTRIBUCIÓN NORMAL, CON UNA NORMAL TRUNCADA, CON UNA GAMMA-UNIFORME Y CON UNA UNIFORME EN UN CÍRCULO. 29

FIGURA 6. MEDIA OBTENIDA DEL NÚMERO DE CLUSTERS SELECCIONADO POR CADA ÍNDICE DE VALIDACIÓN PARA UNA DETERMINADA DISTANCIA. 30

FIGURA 7. RESULTADOS DE LOS DIFERENTES MÉTODOS EN FUNCIÓN A LA FORMA DE LOS CLUSTERS QUE CONFORMAN EL PROBLEMA. 31

FIGURA 8. PROGRESIÓN EN LA FORMA DE LA FDP GENERADA EN FUNCIÓN DEL EXPONENTE K 35

FIGURA 9. DISTRIBUCIÓN DE LOS PUNTOS DE UN CLUSTER EN DIMENSIÓN 2 Y EN FUNCIÓN DEL EXPONENTE K O GRADO DE NORMALIDAD. 36

FIGURA 10. EJEMPLO DE DOS CLUSTERS EN DIMENSIÓN 2, CON UNA SEPARACIÓN A $= 5$. C$1 = 0{,}0$ Y C$2 = (5{,}0)$ 36

FIGURA 11. EJEMPLO DE MÉTODO DE SELECCIÓN DE DISTANCIAS PARA DIMENSIÓN 2. DE ARRIBA ABAJO SERÍA: DISTANCIA DONDE BIC FALLA (DISTANCIA 1), DISTANCIA FRONTERA ENTRE ACIERTO Y FALLO (DISTANCIA 2) Y DISTANCIA A LA CUAL BIC ACIERTA (DISTANCIA 3) 43

FIGURA 12. COMPARATIVA BIC Y ΔJ_{UG} PARA DISTANCIA 3 Y DIMENSIONES PEQUEÑAS, 2D Y 3D. 52

FIGURA 13. COMPARATIVA BIC Y ΔJ_{UG} PARA DISTANCIA 3 Y DIMENSIONES GRANDES, 4D Y 10D. 52

FIGURA 14. NÚMERO TOTAL DE ACIERTOS POR CADA DUPLA DIMENSIÓN-NÚMERO DE PUNTOS DEL SEGUNDO CLUSTER. CADA UNA DE LAS CUATRO GRÁFICAS REPRESENTA LOS RESULTADOS PARA CADA UNO DE LOS MÉTODOS DE VALIDACIÓN 54

FIGURA 15. NÚMERO TOTAL DE ACIERTOS POR CADA DIMENSIÓN Y MÉTODO DE VALIDACIÓN. 58

FIGURA 16. EJEMPLOS DE IMÁGENES DE FERET CON SUS HISTOGRAMAS ASOCIADOS, DONDE CADA BARRA REPRESENTA EL PORCENTAJE DE PUNTOS DEL CLUSTER QUE ESTÁ REPRESENTANDO FRENTE AL TOTAL DE PUNTOS DE LA IMAGEN. 62

INDICE DE TABLAS

TABLA 1. EL PRIMER VALOR DE CADA CELDA REPRESENTA LA MEDIA DE CLUSTERS OBTENIDA A PARTIR LOS DIFERENTES RESULTADOS OBTENIDOS POR UNO DE LOS MÉTODOS PARA UNA DISTANCIA DADA, EL SEGUNDO ES LA DESVIACIÓN ESTÁNDAR DE LOS MISMOS RESULTADOS. 30

TABLA 2. MEDIA Y DESVIACIÓN ESTÁNDAR DEL NÚMERO DE CLUSTERS ESCOGIDOS POR CADA MÉTODO PARA UNA DISTANCIA DETERMINADA CUANDO LOS CLUSTERS TIENEN UNA DISTRIBUCIÓN NORMAL. 32

TABLA 3. MEDIA Y DESVIACIÓN ESTÁNDAR DEL NÚMERO DE CLUSTERS ESCOGIDOS POR CADA MÉTODO PARA UNA DISTANCIA DETERMINADA CUANDO LOS CLUSTERS TIENEN UNA DISTRIBUCIÓN NORMAL TRUNCADA. 32

TABLA 4. MEDIA Y DESVIACIÓN ESTÁNDAR DEL NÚMERO DE CLUSTERS ESCOGIDOS POR CADA MÉTODO PARA UNA DISTANCIA DETERMINADA CUANDO LOS CLUSTERS TIENEN UNA DISTRIBUCIÓN GAMMA-UNIFORME. 32

TABLA 5. MEDIA Y DESVIACIÓN ESTÁNDAR DEL NÚMERO DE CLUSTERS ESCOGIDOS POR CADA MÉTODO PARA UNA DISTANCIA DETERMINADA CUANDO LOS CLUSTERS TIENEN UNA DISTRIBUCIÓN UNIFORME EN UN CÍRCULO. 33

TABLA 6. REPRESENTA LAS TRES DISTANCIAS ESCOGIDAS EN FUNCIÓN DE LA DIMENSIÓN DEL PROBLEMA. DISTANCIA 1 ES LA DISTANCIA DONDE BIC FALLA, DISTANCIA 2 ES LA FRONTERA ENTRE EL FALLO Y EL ACIERTO DE BIC, Y DISTANCIA 3 ES LA DISTANCIA A LA CUAL BIC ACIERTA. 42

TABLA 7. AIC. MEDIA Y DESVIACIÓN ESTÁNDAR POR CADA DUPLA DIMENSIÓN-NÚMERO DE PUNTOS PARA EL MÉTODO AIC. 55

TABLA 8. BIC. MEDIA Y DESVIACIÓN ESTÁNDAR POR CADA DUPLA DIMENSIÓN-NÚMERO DE PUNTOS PARA EL MÉTODO BIC. 55

TABLA 9. ΔJ_{US}. MEDIA Y DESVIACIÓN ESTÁNDAR POR CADA DUPLA DIMENSIÓN-NÚMERO DE PUNTOS PARA EL MÉTODO ΔJ_{US}. 56

TABLA 10. ΔJ_{UG}. MEDIA Y DESVIACIÓN ESTÁNDAR POR CADA DUPLA DIMENSIÓN-NÚMERO DE PUNTOS PARA EL MÉTODO ΔJ_{UG} 56

TABLA 11. MEDIA Y DESVIACIÓN ESTÁNDAR SOBRE LOS AJUSTES ELEGIDOS DE 100 PROBLEMAS POR CADA DIMENSIÓN Y MÉTODO DE VALIDACIÓN. 58

1 Introducción

1.1 Motivación

El objetivo del clustering, también conocido como clasificación no supervisada, es agrupar un conjunto de datos de manera automática, dando como resultado un número finito de grupos o clusters. Estos clusters se conforman siguiendo el criterio tal que los elementos pertenecientes a un mismo cluster son más parecidos entre sí de lo que lo pueden ser con los pertenecientes al resto de clusters.

En la actualidad, las técnicas de clustering son técnicas muy utilizadas en la minería de datos y multitud de campos como: análisis y clasificación de imágenes (3) (4), biología (5), medicina (6), marketing (7), negocios (8) o análisis de las redes sociales (9).

Un problema con el que se enfrentan los algoritmos de clustering es medir la validez del clustering realizado. Diferentes métodos pueden obtener diferentes resultados cuando son aplicados al mismo conjunto de datos, e incluso el mismo método puede proporcionar diferentes resultados dependiendo de las condiciones iniciales que se establezcan, como por ejemplo el orden de los datos o los valores de los parámetros.
Para intentar solucionar este problema existen varios métodos de validación de clusters, que asignan una medida de calidad o índice de validación a cada partición basándose en diferentes criterios. Estos métodos nos permiten escoger de entre todas la particiones realizadas, las que mejor se ajusten al criterio que utilice cada método de validación. Por ello, contar con un algoritmo de validación apropiado a estas técnicas es una herramienta de mucha utilidad en las áreas de trabajo anteriormente mencionadas.

El objetivo principal de este proyecto es el estudio de un nuevo conjunto de algoritmos de validación de clusters. Estos índices de validación se basan en la negentropía de las particiones, que es una medida de la distancia a la normalidad. Para su análisis, se realizarán diferentes pruebas con estos algoritmos y se compararán sus resultados con los obtenidos por dos índices más clásicos.

1.2 Objetivos

En este proyecto se plantean dos objetivos principales:

- El estudio y evaluación de índices de validación de clusters basados en la negentropía, así como su comparación con índices de validación más clásicos. Para tal fin se creará una batería de problemas que permita estudiar y cuantificar de manera precisa el comportamiento de los índices de validación en función de parámetros como el número de puntos por cluster, la dimensión de los datos, la forma de los clusters o el grado de solape entre los mismos.

- La evaluación de las distintas técnicas estudiadas utilizando algunas aplicaciones reales.

Para el cumplimiento de los objetivos se llevarán a cabo las siguientes tareas:

1. **Estudio de la literatura y el estado del arte** en relación con el análisis de clusters en general y la validación de clusters en particular.

2. **Implementación de algoritmos de generación y validación de clusters**. El planteamiento inicial consiste en generar los clusters ajustando los datos a una mezcla de gaussianas, utilizando para ello el algoritmo Expectation Maximization (EM), con diferentes parámetros y condiciones iniciales, y validar a posteriori las soluciones generadas usando diferentes índices de validación.

3. **Generación de una batería de problemas sintéticos** que permitan la evaluación de los índices en función de los parámetros indicados en el apartado anterior. El uso de problemas artificiales permite conocer los clusters reales, facilitando la evaluación de los índices.

4. **Evaluación de los índices de validación** usando los problemas generados en el punto 3.

5. **Análisis de resultados y extracción de conclusiones**. Una vez finalizadas todas las pruebas y obtenidos todos los resultados, se extraerán las conclusiones en función a los mismos.

1.3 Organización de la memoria

La memoria consta de las siguientes secciones.

1. **Introducción:** que explica la motivación, los objetivos del proyecto y la estructura de la memoria.

2. **Estado del arte:** se estudian varios algoritmos de clustering, algunos métodos de validación de clusters entre los que se incluye el algoritmo que se pretende estudiar en este proyecto y un algoritmo para reducir la dimensión de los datos.

3. **Diseño y desarrollo:** Se introduce el orden seguido en la realización de las pruebas así como el procedimiento utilizado para extraer los resultados.

4. **Pruebas y resultados en problemas sintéticos:** se describen las pruebas y los resultados obtenidos, haciendo un análisis comparativo entre los diferentes métodos de validación en problemas sintéticos.

5. **Conclusiones y trabajo futuro:** discusión de los resultados obtenidos en todas las pruebas y extracción de conclusiones sobre las mismas. Por último hay una propuesta de trabajos futuros que se podrían realizar como continuación de este proyecto.

2 Estado del arte

2.1 Introducción

Como se introdujo en punto el anterior, el clustering es un método de agrupación de un conjunto de puntos de acuerdo a un criterio, siendo este por lo general distancia o similitud entre los puntos. Suele suceder que los puntos de un mismo cluster o grupo tienen propiedades comunes o son muy similares, mientras que los que pertenecen a diferentes clusters tienen propiedades diferentes.

Los métodos de clustering se pueden dividir en dos grupos atendiendo a las suposiciones iniciales de las que se parte. Estos son:

- **Métodos paramétricos:** se supone que los datos están generados por una mezcla de distribuciones conocida. El objetivo de estos métodos es estimar los parámetros que hacen que la mezcla de distribuciones se ajuste a los datos de la mejor manera posible.
  El método estándar que se basa en maximizar la verosimilitud es el método Expectation-Maximization (EM) (10), el cual se describe en uno de los siguientes sub-apartados de esta misma sección. Este método ha sido el que se ha escogido para la realización de las pruebas a lo largo de todo el proyecto.

- **Métodos no paramétricos:** no se realizan suposiciones acerca de la distribución de los datos, sino que se buscan particiones de los datos en clusters naturales conforme a algún criterio de similitud.
  Dentro de estos métodos se encuentran, por ejemplo, k-means (11) o el clustering jerárquico (12), los cuales también se describirán en siguientes apartados de esta sección aunque más brevemente que el método EM pues no se han utilizado en este proyecto.

Otra forma de clasificar los métodos de clustering es atendiendo al grado de pertenencia de cada punto a un determinado cluster:

- **Exclusivo (*crisp clustering*):** en estos métodos, cada punto pertenece unívocamente a un cluster determinado.

- **Difuso (*fuzzy clustering*):** en estos métodos, los puntos no tienen una pertenencia unívoca a un cluster, sino que se les asigna una probabilidad de pertenencia a cada uno de los clusters mediante una función. Estos métodos eliminan la suposición de que los elementos de un cluster son totalmente diferentes a los elementos de otros clusters.

Comúnmente a la hora de realizar un *fuzzy clustering* suelen utilizarse métodos paramétricos, esto es, suponer que los datos observados provienen de una mezcla de distribuciones. Las estructuras matemáticas de estas distribuciones suelen suponerse de un cierto tipo, pero los parámetros específicos han de ser hallados, como por ejemplo la media y la varianza en el caso de una distribución normal. Una vez se han escogido el número de

componentes a ajustar y sus parámetros, el grado de pertenencia de un determinado punto x al cluster c suele estar relacionado con la probabilidad $p(c|x)$.

Existen diferentes métodos para seleccionar los parámetros del modelo de mezclas. El más popular es Expectation-Maximization (EM), que consiste en maximizar la verosimilitud de los parámetros dadas las observaciones, siguiendo un proceso iterativo. Estos métodos suelen resultar en modelos diferentes por cada ejecución debido a las condiciones iniciales. Por ello, se necesita un criterio para poder escoger el modelo que se considere óptimo. Los métodos de validación de clustering aportan una medida de calidad o índice de validación de los modelos atendiendo a unos determinados criterios, y son los utilizados para la elección del modelo óptimo.

Hay multitud de índices de validación en la literatura, pero los más populares en el ámbito de mezcla de modelos estadísticos suelen ser correcciones a la verosimilitud, ejemplos de estos son Akaike's Information Criterion (AIC) (13) y Bayesian Inference Criterion (BIC) (14).

AIC y BIC son los métodos de validación de clustering que se han escogido para realizar el análisis comparativo con los índices de validación basados en la negentropía de las particiones, Negentropy-based Validation (1) (15) (16) (17), los cuales son los que se busca analizar en este proyecto.

2.2 Clasificación no supervisada (clustering)

2.2.1 Métodos paramétricos

Se parte del supuesto de que los datos han sido generados por una mezcla de distribuciones conocidas, y se tiene como objetivo estimar los parámetros que hacen que la mezcla se ajuste de la mejor manera posible a los datos, esto es, que la verosimilitud de los parámetros dados los datos observados sea máxima.

Sea el conjunto de datos $X = \{x_1, x_2, \dots, x_n\}$, modelamos su función de densidad de probabilidad como una mezcla de k distribuciones:

$$f(x|\theta) = \sum_{j=1}^{k} p(x|\theta_j)\pi_j \qquad \textbf{[1]}$$

Donde θ_j es el conjunto parámetros para la distribución j, π_j la probabilidad a priori para la distribución j y θ el conjunto de todos los parámetros $\theta = \{\theta_j, \pi_j\}$

2.2.1.1 Máxima verosimilitud

Supóngase que se tiene una muestra de n observaciones independientes extraídas de una función de distribución desconocida con función de densidad de probabilidad $f_0(\cdot)$. Se sabe, sin embargo, que f_0 pertenece a una familia de distribuciones $\{ f(\cdot|\theta),\ \theta \in \Theta \}$ llamada modelo paramétrico, de manera que f_0 se corresponde con el valor $\theta = \theta_0$, que es el verdadero valor del parámetro.

Se desea encontrar el valor $\hat{\theta}$, llamado estimador, que esté lo más próximo posible al verdadero valor θ_0.

Lo que plantea este método es encontrar primero la función de densidad conjunta de todas las observaciones, que bajo condiciones de independencia, es la siguiente:

$$f(x_1, x_2, \ldots, x_n|\,\theta) = f(x_1|\,\theta) \cdot f(x_2|\,\theta) \cdots f(x_n|\,\theta)$$

Donde tanto x_i como θ pueden ser vectores.

Observando esta función bajo un ángulo ligeramente distinto, se puede suponer que los valores observados $x_1,\ x_2, \ldots,\ x_n$ son fijos mientras que θ puede variar libremente. Ésta es la verosimilitud de θ dados los parámetros observados:

$$Q(\theta \mid x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f(x_i|\theta) \quad \textbf{[2]}$$

En la práctica, se usa el logaritmo de la función anterior:

$$\mathrm{L}(\theta) = \log \mathrm{Q}(\theta) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \log f(x_i|\theta) \quad \textbf{[3]}$$

Lo que se pretende con este método es encontrar el conjunto de parámetros θ que maximiza $L(\theta)$

$$\hat{\theta} = arg\,máx \sum_{i=1}^{n} log\, f(x_i|\theta) \quad \textbf{[4]}$$

Utilizando la expresión [1], donde f es una mezcla de k distribuciones, la función de verosimilitud logarítmica quedaría de la siguiente forma:

$$L(\theta) = \sum_{i=1}^{n} log \sum_{j=1}^{k} p(x_i|\theta_j)\pi_j \quad \textbf{[5]}$$

Este problema no tiene una solución analítica cerrada y se resuelve con métodos iterativos como el método EM explicado a continuación.

2.2.1.2 Expectation Maximization (EM)

El algoritmo Expectation Maximization (10), es un método iterativo para ajustar una distribución a un conjunto de datos mediante máxima verosimilitud.

La idea de EM es suponer que conocemos la distribución z_i que ha generado cada punto x_i ($z_i = j$ si el punto x_i ha sido generado por la distribución j). Entonces, la función verosimilitud tendría el siguiente aspecto:

$$LC(\theta) = \sum_{i=1}^{n} \log p\left(x_i \middle| \theta_{z_i}\right) \pi_{z_i} \quad \textbf{[6]}$$

El problema que nos encontramos es que no conocemos que distribución ha generado cada punto, por lo que necesitamos aproximar el problema de otro modo.

Como z_i no lo conocemos, lo que hacemos es promediar usando su probabilidad a posteriori:

$$\widetilde{LC}(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{k} p(z_i = j | x_i, \theta)\, log\, p(x_i|\theta_j)\, \pi_j \quad \textbf{[7]}$$

Por el teorema de Bayes, la probabilidad a posteriori de $z_i = j$ se puede escribir como:

$$p(z_i = j | x_i, \theta) = \frac{p(z_i = j, x_i|\theta)}{p(x_i|\theta)} = \frac{\pi_j p(x_i|z_i = j, \theta)}{\sum_{j=1}^{k} \pi_j p(x_i|z_i = j, \theta)} \quad \textbf{[8]}$$

Las expresiones [7] y [8] sugieren proceder en dos pasos diferenciados:

Expectación
A partir de un conjunto de parámetros estimados previamente θ^{old}y ${\pi_j}^{old}$, calculamos la probabilidad a posteriori de $z_i = j$ usando [8]

$$p(z_i = j | x_i, \theta^{old}) = \frac{{\pi_j}^{old} p(x_i|z_i = j, \theta^{old})}{\sum_{j=1}^{k} {\pi_j}^{old}{}_j p(x_i|z_i = j, \theta^{old})} \quad \textbf{[9]}$$

Y a partir de ella obtenemos el valor esperado de la verosimilitud como:

$$\widetilde{LC}(\theta, \theta^{old}) = \sum_{i=1}^{n} \sum_{j=1}^{k} p(z_i = j | x_i, \theta^{old}) \log \mathrm{p}(\mathrm{x_i}|\theta_j)\, \pi_j \quad \textbf{[10]}$$

Maximización
Maximizamos el valor esperado de la verosimilitud con respecto a θ, dejando fijo θ^{old} para obtener el siguiente conjunto de parámetros:

$$\theta^{new} = \arg máx\, LC(\theta, \theta^{old}) \quad \textbf{[11]}$$

Puede demostrarse que este procedimiento converge a un máximo local de la función verosimilitud de la ecuación [3] (10).

2.2.2 Métodos no paramétricos

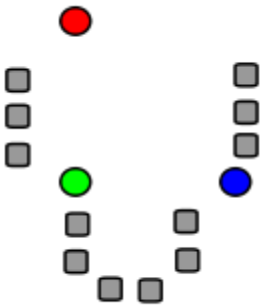
Estos métodos no hacen suposiciones acerca de la posible distribución de los datos, si no que realizan particiones de los datos en clusters atendiendo a alguna medida de similitud. El algoritmo más famoso es k-means (11) que describimos a continuación. Al final de la sección hacemos un repaso a métodos jerárquicos.

2.2.2.1 K-means

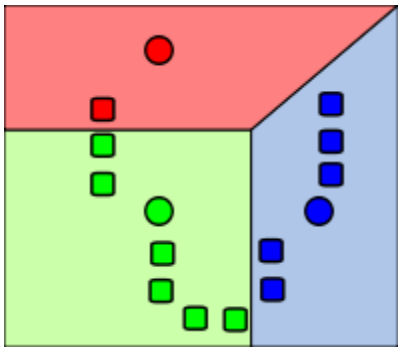
El método k-means (11) se basa en un criterio de agrupamiento, siguiendo el siguiente procedimiento:
Primeramente se han de escoger empíricamente un número de centroides k, que se corresponderá con el número de clusters a los que se pretenda ajustar. Los centroides han de ser situados del mejor modo posible, puesto que el resultado se verá afectado por la posición inicial.
Tras la elección de los centroides, se toma uno a uno cada punto del conjunto de datos a tratar asociándose al centroide más cercano, es decir menor distancia punto-centroide, hasta que el último haya sido asociado. Esta distancia acostumbra a ser la distancia euclídea aunque pueden emplearse multitud de distancias diferentes. Cuando el último punto se ha asociado se ha completado la primera iteración del algoritmo y se ha realizado el primer agrupamiento.
Tras esto, se vuelven a calcular los k nuevos centroides, cada uno a partir del promedio de los puntos asociados a cada uno de los k clusters obtenidos en el paso anterior. Este proceso se repite iterativamente hasta que el método converge, es decir, hasta que ningún punto cambia de un cluster a otro.

Explicando este procedimiento de un modo gráfico sería como sigue:

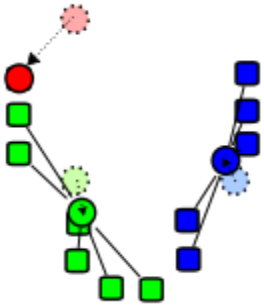
1. Se tiene un conjunto de datos (cuadrados grises), y se escogen k centroides al azar dentro del conjunto de datos (puntos de color). En este ejemplo $k = 3$



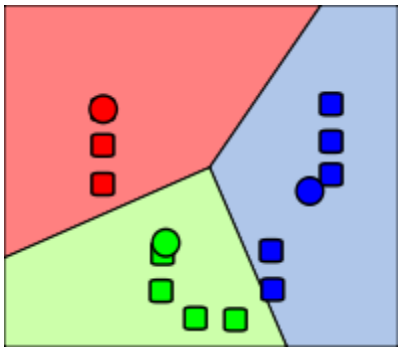
2. Se generan *k* clusters asociando cada punto al centroide más cercano.



3. Se recalculan los centroides como la media de los puntos del cluster.



4. Se repiten los pasos 2 y 3 hasta que ningún punto se mueva de un cluster a otro (convergencia).



2.2.2.2 Clustering jerárquico

Estos métodos tienen como objetivo: bien agrupar clusters para formar un cluster nuevo que contiene los puntos de los clusters agrupados, o bien separar alguno ya existente creando dos nuevos a partir de los puntos que contenía el cluster inicial. Para tomar la decisión de que clusters unir o separar, se basa en una medida de similitud. De este modo, si se efectúa este proceso de manera iterativa se consigue maximizar esa medida de similitud.

Estos métodos se dividen en métodos aglomerativos y divisivos, presentando una gran diversidad de variantes cada una de estas dos categorías:

- Los métodos aglomerativos comienzan el análisis con tantos clusters como puntos a analizar se tenga. A partir de estas unidades, se van formando nuevos clusters hasta que todos los puntos tratados están englobados en un mismo cluster.

- Los métodos divisivos se basan en el proceso inverso al anterior, es decir, se comienza con un cluster que engloba todos los puntos, y a partir de este cluster inicial, se comienza a hacer divisiones formándose cada vez clusters más pequeños, obteniéndose al final tantos clusters como puntos tratados.

El uso de métodos aglomerativos es más usual debido a que es más sencilla la toma de decisión para unir los clusters que para separarlos. Explicando el método aglomerativo de un modo más formal sería:
Sea n el número de puntos en nuestra muestra, tenemos n clusters en el nivel $K = 0$, en el que cada cluster contiene un punto. En el siguiente nivel $K = 1$, se agrupan aquellos dos clusters con menor distancia o mayor similitud; así, en el nivel L tendríamos $n - L$ clusters formados, llegando hasta el nivel $L = n - 1$, en el que solo hay un cluster que contiene todos los puntos de la muestra.

Este método de formar clusters tiene la particularidad de que si en un nivel determinado se agrupan dos clusters, estos quedan ya agrupados para el resto de niveles.

Los métodos jerárquicos permiten la construcción de un árbol de clasificación, o dendrograma (ver Figura 1), en el cual se puede seguir de forma gráfica el procedimiento de unión que se ha seguido mostrando: que clusters se van uniendo, en qué nivel y el nivel de fusión, que no es más que el valor de la medida de asociación entre los clusters cuando estos se agrupan.

Y ya por último, los criterios de parada de unión o división pueden ser cualquiera de los siguientes:

- Que se forme un solo cluster en caso de estar utilizando un método aglomerativo o que se formen n clusters en caso de ser divisivo.

- Que se alcance un número de clusters prefijado.

- Que se detecte que hay razones estadísticas para no continuar agrupando o dividiendo clusters, ya que los más similares no son lo suficientemente homogéneos como para determinar una misma agrupación.

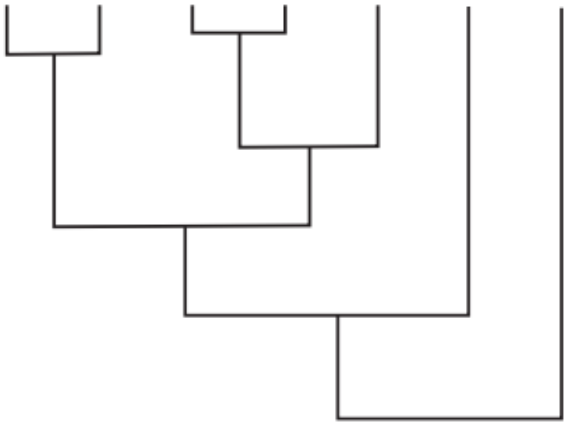


Figura 1. Ejemplo de dendrograma

2.3 Validación de clusters

2.3.1 Introducción al problema

Los métodos de clustering, como se ha explicado previamente, pueden resultar en modelos diferentes por cada ejecución dependiendo del número de componentes a los que se ajuste en cada problema n_c. Además, aún con el mismo valor de dicho parámetro el mismo método puede dar resultados diferentes por cada ejecución dependiendo de las condiciones iniciales que se establezcan. Esto introduce el problema de escoger entre todos los ajustes realizados, y para lidiar con este problema existen diferentes métodos de validación de clusters.

Estos métodos de validación nos dan una medida de la calidad relativa del modelo mediante un índice de validación en base a unos criterios preestablecidos.

Uno de estos métodos de validación es en el que se basa este proyecto y como parte de su análisis, se han realizado comparaciones entre éste y otros dos índices de referencia en el contexto de mezcla de gaussianas.

A continuación se describen los métodos que se han utilizado en este proyecto.

2.3.2 Akaike Information Criterion (AIC)

El método Akaike Information Criterion (AIC) (13) se basa en la elección de un modelo atendiendo a la entropía de información: se ofrece una estimación relativa de la información perdida cuando se utiliza un modelo determinado para representar el proceso que genera los datos.
En el caso general, el índice AIC es:

$$AIC = 2c - 2L \quad \textbf{[12]}$$

Donde c es el número de parámetros en el modelo estadístico y L el valor de la verosimilitud logarítmica [3] para el modelo estimado.

Dado un conjunto de modelos candidatos para los datos, el modelo preferido es el que obtiene el valor mínimo de AIC, recompensando la verosimilitud y penalizando el número de parámetros estimados. Esta penalización se introduce debido a que al aumentar el número de parámetros, la verosimilitud mejora, y esto podría hacer que se tendiera siempre a sobrestimar el número de componentes. Introduciendo esta penalización se persigue evitar el sobreajuste.

AIC no estima la calidad de un modelo de manera absoluta, sino que proporciona una manera de escoger entre un conjunto de modelos, el que mejor se ajusta. Esto hace que si todos los modelos se ajustan mal, AIC no avisará de ello.

2.3.3 Bayesian Information Criterion (BIC)

Bayesian Information Criterion (BIC) (14) es uno de los métodos más utilizados en problemas de mezcla de gaussianas y se basa en criterios bayesianos.

El índice BIC tiene la siguiente forma:

$$BIC = -2 \times L + c\, ln(n) \quad \textbf{[13]}$$

Donde n es el número de datos u observaciones, o equivalentemente, el tamaño de la muestra; c es el número de parámetros libres a ser estimados y L el valor de la verosimilitud logarítmica [3] para el modelo estimado.
Al igual que BIC, AIC no estima la calidad del modelo de manera absoluta.

2.3.4 Negentropy-based Validation (ΔJ)

El método Negentropy-based Validation (ΔJ) (1) (15) (16) (17) se basa en la medida de la normalidad de un cluster. La normalidad está caracterizada por la negentropía, una medida de la distancia a la normalidad que evalúa la diferencia entre la entropía del cluster y la de una distribución normal con la misma matriz de covarianza.
Para conseguir el índice, se calculan incrementos de negentropía con respecto a la distribución inicial de los datos, donde se supone que todos los puntos pertenecen al mismo cluster.

Consideremos una variable aleatoria X en un espacio d-dimensional, distribuida acorde a una función de densidad de probabilidad $f(x)$. Sea $s = \{x_1, \dots, x_n\}$ una muestra aleatoria de X, y consideremos la partición del espacio en un conjunto de k regiones no solapadas $\Omega = \{\omega_1, \dots, \omega_k\}$ que cubre el conjunto completo del espacio. Esta partición impone una estructura de clustering exclusiva, o crisp clustering, sobre los datos con k clusters consistentes en el conjunto de puntos que caen en cada una de las k particiones del espacio.

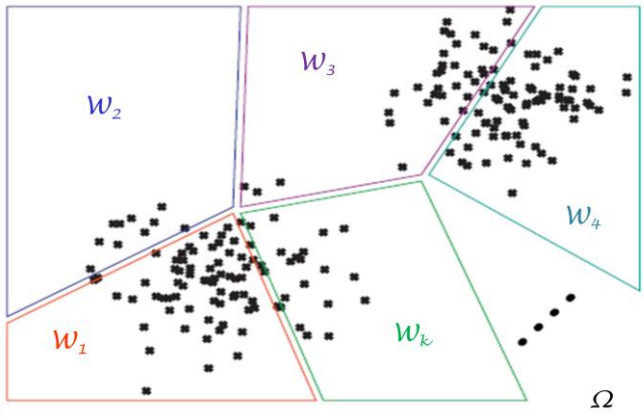


Figura 2. Partición del conjunto de datos en k regiones no solapadas [1]

El índice de incremento de negentropía de la partición de clustering Ω aplicado a X se define como (17):

$$\Delta J(\Omega, X) = \frac{1}{2}\sum_{i=1}^{k} \pi_i \log |\Sigma_i| - \sum_{i=1}^{k} \pi_i \log \pi_i \quad \textbf{[14]}$$

Donde π_i y Σ_i son respectivamente la probabilidad a priori y la matriz de covarianza de X restringida a la región Ω_i. El incremento de negentropía es una medida de la media de la normalidad que se gana haciendo una determinada partición del conjunto de datos. A menor valor de $\Delta J(\Omega)$, más gaussianos son los clusters en media, por lo tanto, la regla para la validación de clusters es seleccionar la partición que minimiza el índice de incremento de negentropía. Puesto que en una situación práctica, no se conoce la distribución completa de X, tenemos que estimar el incremento de negentropía a partir de una muestra finita s. Una estimación directa puede hacerse usando el índice:

$$\Delta J_B(\Omega, s) = \frac{1}{2}\sum_{i=1}^{k} \tilde{\pi}_i \log |\tilde{\Sigma}_i| - \sum_{i=1}^{k} \tilde{\pi}_i \log \tilde{\pi}_i \quad \textbf{[15]}$$

Donde $\tilde{\pi}_i$ y $\tilde{\Sigma}_i$ son las estimaciones muestrales de π_i y Σ_i respectivamente. El subíndice B se ha introducido para enfatizar que esta estimación del incremento de negentropía tiene un sesgo debido a una estimación incorrecta de los términos involucrados en el logaritmo del determinante (18). Esta desviación puede corregirse usando la expresión:

$$\Delta J_U(\Omega, s) = \Delta J_B(\Omega, s) + \frac{1}{2}\sum_{i=1}^{k} \tilde{p}_i C(n_i, d) \quad \textbf{[16]}$$

Donde $C(n_i, d)$ es un término correctivo para el determinante logarítmico el cual depende solo del número de puntos de la muestra en la región i, n_i, y de la dimensión (19):

$$C(n_i, d) = -d\ log\frac{2}{n_i - 1} - \sum_{j=1}^{d} \Psi\left(\frac{n_i - j}{2}\right) \quad \textbf{[17]}$$

Aquí Ψ es la función digamma (20). Puede demostrarse que la expresión en [16] es un estimador no sesgado de ΔJ, es decir:

[1] Imagen obtenida del artículo (17) y adaptada al texto

$$E[\Delta J_U(\Omega, s)]_s = \Delta J(\Omega) \quad \textbf{[18]}$$

Y que la varianza de $\Delta J_U(\Omega, s)$ puede ser estimada de la siguiente manera:

$$\sigma_s^2(\Delta J_U) \approx \frac{1}{4}\sum_{i=1}^{k} \widetilde{\pi_i}^2 \sum_{j=1}^{d} \Psi'(\frac{n_i - j}{2}) \quad \textbf{[19]}$$

Donde Ψ' es la primera derivada de la función digamma. Diferentes usos de estos resultados ofrecen diferentes enfoques de validación, los cuales se presentan a continuación.

Enfoques de validación

La regla general para la validación de clusters basada en el incremento de negentropía es que, dado un set de particiones de clusters $\Pi = \{\Omega_1, \dots, \Omega_k\}$ en un problema dado y definido por la variable aleatoria X, se debería escoger la partición Ω_i para la cual $\Delta J(\Omega_i)$ es mínimo. Esto es:

$$\Delta J(\Omega_i) \leq \Delta J\big(\Omega_j\big)\ \forall\ j = 1, \dots, k$$

Esto significa que los clusters resultantes de Ω_i son, en media, más gaussianos que los otros resultantes de cualquier otra partición de Π. En términos prácticos, nunca sabemos los valores de $\Delta J(\Omega_\mathrm{i})$, sino solo estimaciones obtenidas de una muestra finita s. De las diferentes aproximaciones descritas anteriormente al inicio de esta sección se pueden obtener los siguientes enfoques:

Índice sesgado. La primera posibilidad es usar la estimación $\Delta J_B(\Omega, s)$, de la ecuación [15]. El valor mínimo de ΔJ_B sobre Π será el que nos de la partición válida.

Índice no sesgado V1. Una segunda posibilidad es considerar $\Delta J_U(\Omega, s)$ en la ecuación [16]. Al igual que antes, la minimización de ΔJ_U sobre Π será lo que nos dé la partición válida.

Índice no sesgado V2. La posibilidad anterior no tiene en cuenta la varianza en la estimación debido a que es una muestra finita. Por ello podría pasar que para dos particiones diferentes, pongamos Ω_1 y Ω_2, los valores reales del incremento de negentropía satisfagan que $\Delta J(\Omega_1) < \Delta J(\Omega_2)$, y las estimaciones satisfagan $\Delta J_U(\Omega_1, s) > \Delta J_U(\Omega_2, s)$. Para minimizar ese efecto seguimos la siguiente aproximación en (18) y se consideran las dos particiones equivalentes si:

$$\Delta J_U(\Omega_2, s) + \sigma_s(\Delta J_U(\Omega_2)) < \Delta J_U(\Omega_1, s) - \sigma_s(\Delta J_U(\Omega_1)) \quad \textbf{[20]}$$

En tal caso, seleccionamos la partición más simple, es decir, la que tenga un menor número de regiones. A este enfoque nos referiremos como ΔJ_{US}.

Índice no sesgado V3. Si hacemos la suposición de que el $\Delta J(\Omega)$ real está distribuido normalmente alrededor de $\Delta J_U(\Omega, s)$ con varianza $\sigma_s^2(\Delta J_U)$, podemos estimar la probabilidad de que $\Delta J(\Omega_1) < \Delta J(\Omega_2)$ mediante:

$$P\big(\Delta J(\Omega_1) < \Delta J(\Omega_2)\big) = \int_{-\infty}^{\infty} dx\ f_2(x) F_1(x) \quad \textbf{[21]}$$

Donde $f_i(x)$ y $F_i(x)$ son, respectivamente, la función de densidad y la acumulada de una variable aleatoria Gaussiana $X \sim N(\Delta J_U(\Omega_i, s), \sigma_s(\Delta J_U))$. Por ello, podemos considerar las dos particiones equivalentes si P es menor que un umbral dado α. En ese caso se seleccionará, al igual que en el enfoque anterior, la partición más simple. Consideraremos para las pruebas en las que se use este enfoque $\alpha = 0.8$, es decir, consideraremos que Ω_1 es mejor que Ω_2 solo si $P\big(\Delta J(\Omega_1) < \Delta J(\Omega_2)\big) > 0.8$. Nos referiremos a este método como ΔJ_{UG}.

2.4 Reducción de la dimensión

Cuando los puntos que se van a analizar están caracterizados por muchas componentes, o lo que es lo mismo, son de una dimensión elevada, se puede utilizar un método de reducción de la dimensión como paso previo al método de clustering, consiguiendo de este modo, reducir el ratio número de puntos frente a dimensión de los datos, que suele afectar de manera negativa al proceso de clustering.
El método Principal Component Analysis (PCA) (21) es el que se ha escogido en este proyecto cuando ha sido necesario tratar con datos reales. Este método también se explica brevemente a continuación.

A medida que aumenta la dimensión, el volumen de espacio aumenta exponencialmente, haciendo que los datos disponibles queden más dispersos a lo largo de ese espacio. Cuando estos datos se están tratando estadísticamente, una dimensión elevada provoca que cada uno de estos datos aporte menos información sobre el suceso que lo ha generado. Además en el clustering, que como ya se ha explicado previamente busca encontrar similitudes entre los diferentes puntos para agruparlos, cuando la dimensión es alta, es más complicado encontrar esas similitudes. Esto es lo que se conoce como la maldición de la dimensión (22).

2.4.1 Principal Component Analysis

El principal objetivo de esta técnica es extraer la información importante de un conjunto de datos y representarlos mediante una cantidad menor de variables sintéticas nuevas, a fin de hallar la relación entre éstas y las variables originales.

PCA construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos. En este nuevo sistema de coordenadas la varianza de mayor valor del conjunto de datos es capturada en el primer eje como la primera componente principal; la segunda varianza de mayor valor en el segundo eje como la segunda componente principal y así sucesivamente.

Para construir esta transformación lineal debe construirse primero la matriz de covarianza. Debido a la simetría de dicha matriz, existe una base completa de autovectores de la misma.
La transformación que lleva de las antiguas coordenadas a coordenadas de la nueva base es precisamente la transformación lineal necesaria para reducir la dimensión de los datos.

Se parte de un conjunto de datos X con n muestras, cada una de las cuales tiene d variables que la describen, así X tiene dimensión $(n\,x\,d)$. El objetivo es que cada una de esas muestras sea descrita con l variables donde $l < d$ y es lo que llamamos número de componentes principales.

Partiendo de ese objetivo, el procedimiento que se ha seguido para aplicar PCA es el siguiente:

1. Estandarización de los datos, esto es, se centran restando a cada punto la media del conjunto, y tras esto se dividen entre la desviación estándar.

2. Se calculan los autovalores y autovectores de la matriz de covarianzas de los nuevos datos calculados en el punto 1. Los autovalores representan la cantidad de varianza capturada por el autovector asociado, y el autovector será la componente principal.

3. Se ordenan de mayor a menor los autovalores y los autovectores asociados.

4. Se escogen de mayor a menor l autovectores, los cuales serán las l componentes principales de nuestra nueva base, la primera componente será el autovector cuyo autovalor asociado sea el mayor de todos los calculados; la segunda será el autovector con segundo mayor autovalor, y así sucesivamente hasta conseguir las l componentes principales.

5. Una vez se han escogido las l componentes principales, se proyectan todos los puntos sobre la nueva base que éstas forman. Y el conjunto resultante de la proyección ese es el nuevo conjunto de puntos en dimensión l.

En la Figura 3 puede verse una imagen donde se ha aplicado el algoritmo PCA a un conjunto datos con una distribución normal multivariante. Los vectores representan los autovectores de la matriz de covarianzas escalados mediante la raíz cuadrada del correspondiente autovalor, y desplazados para que su origen coincida con la media estadística. Así puede observarse que la mayor varianza de los datos se da sobre el vector con mayor norma.

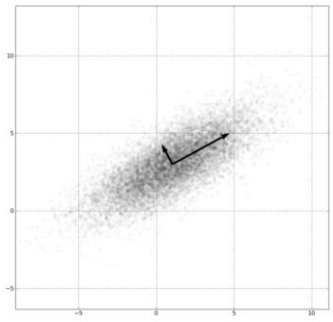


Figura 3. PCA[2]

[2] Figura obtenida de (26)

3 Diseño y Desarrollo

En el capítulo previo se han abordado los puntos teóricos necesarios para la consecución de los objetivos planteados.

El objetivo de este proyecto es hacer un análisis exhaustivo de métodos de validación basados en la negentropía de las particiones. Para ello se les ha sometido a diferentes pruebas que nos permiten evaluar su comportamiento en diferentes situaciones. En cada caso, se contrasta el resultado obtenido por estos métodos con el obtenido en el mismo problema por los métodos AIC y BIC.

Para el desarrollo práctico de todo el proyecto se ha utilizado la herramienta de procesamiento matemático Matlab®, instalada en servidores capaces de trabajar en diferentes núcleos simultáneamente puesto que el coste computacional que conlleva realizar todos los cálculos necesarios es muy elevado. Dividiendo la carga entre los diferentes núcleos se consigue disminuir el tiempo de procesamiento de un modo considerable y proporcional al número de núcleos del servidor.

El procedimiento base que se ha seguido para el análisis de los diferentes problemas ha sido el mismo aunque se ha tenido que ir adaptando el código a los datos que se introducían, al coste computacional que generaba y a la forma de presentar los resultados.

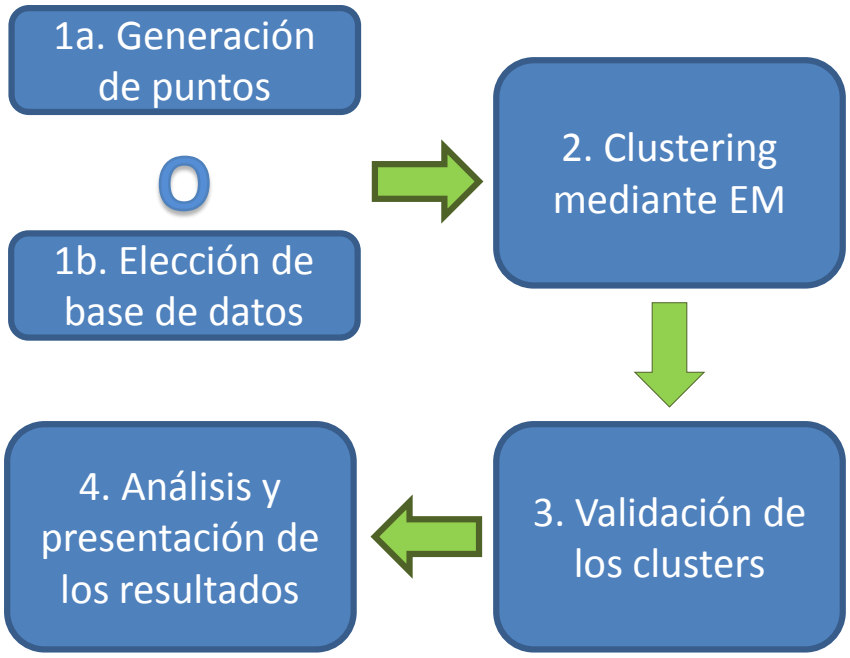


Figura 4. Proceso seguido para la resolución de los problemas abordados.

A continuación se describe cada uno de los puntos enumerados en el diagrama de la Figura 4.

1a.- Generación de puntos: se genera el conjunto de puntos con los que se va a trabajar posteriormente. Estos puntos son almacenados para no tener que realizar este proceso cada vez que se requiera repetir uno de los puntos posteriores.

1b.- Elección de una base de datos: En caso de existir una base de datos que pueda ser utilizada, no es necesario generar el conjunto de datos. Un ejemplo puede ser bases de datos con datos reales.

2.- Clustering mediante EM: Se ajustan los datos de cada problema a una mezcla de n_c componentes gaussianas, utilizando para ello el método EM. Para cada ajuste a un n_c determinado, sobre los mismos datos, se repite m veces el algoritmo EM, resultando en m ajustes diferentes.
Por ejemplo, se quiere que el método EM ajuste a $n_c = \{1, 2, 3, 4\}$ clusters. Para cada uno de esos ajustes, se van a ejecutar $m = 10$ repeticiones. Así, al final de este proceso se dispondrá de $10\left(\frac{ejecuciones}{ajuste}\right) * 4(ajustes) = 40\, ejecuciones$. Nos referiremos como ejecución $E_{n_c,i}$ a la ejecución del algoritmo EM con ajuste n_c y número de repetición i. En el ejemplo, la ejecución $E_{2,5}$ nos estaremos refiriendo a la quinta repetición del ajuste a dos clusters.

Este proceso es el de mayor carga computacional, por ello guardar los resultados una vez calculados resulta imprescindible para poder trabajar después con ellos sin necesidad de volver a ejecutar este proceso cada vez que se requiera realizar el paso siguiente.

3.- Validación de los clusters: Cada uno de los ajustes del método EM devuelve una partición fuzzy, esto es, a cada punto le da una probabilidad de pertenecer a cada cluster. Debido a que el método ΔJ trabaja sobre particiones tipo crisp, es necesario asignar cada punto a un cluster, obteniendo así una partición con la que ΔJ pueda trabajar. El criterio para asignar cada punto es escoger el cluster al que mayor probabilidad tenga de pertenecer.
Tras ello ejecutamos el método AIC, BIC y ΔJ sobre la partición y se almacena su resultado.
Para finalizar, se escoge por cada uno de los métodos de validación la ejecución $E_{n_c,i}$ que mejor resultado ha obtenido por cada uno de los métodos. Estos resultados son almacenados también para un posterior análisis.

4.- Análisis y presentación de los resultados: Los resultados obtenidos en el punto anterior se procesan y se representan del modo escogido para cada uno de los diferentes problemas. Estas representaciones pueden ser tablas o figuras que representan las tendencias de los diferentes algoritmos en función de las variables que intervienen en el problema.

En el Anexo C se puede encontrar un ejemplo del código generado dónde se implementan, para el análisis descrito en la sección 4.2.2, las cuatro fases descritas arriba. A lo largo de los problemas, se ha intentado que el código fuese lo más abierto posible, es decir, que las condiciones iniciales del problema que se utilizan se puedan variar con gran facilidad para poder evaluar diferentes escenarios sin implicar grandes cambios en el código. Esto puede verse también en el ejemplo de código del Anexo C, donde se toman como condiciones iniciales: las dimensiones en las que se va a analizar el desempeño de los métodos, el rango del exponente k, el rango de distancias y el número de puntos de cada cluster en función de la dimensión. Todos estos valores están definidos mediante vectores al inicio del código, lo que permite ejecutar el mismo código sobre diferentes condiciones únicamente modificando los vectores que definen las condiciones iniciales.

En cuanto a la estructura de las pruebas, éstas siguen un orden de menor a mayor complejidad. Se comienza con problemas donde solo interviene como variable la distancia de separación o solape de los clusters. Después, se introducen más variables en los problemas a analizar como: el tipo de función de densidad de probabilidad que genera los clusters, el desbalanceo en la cantidad de puntos de un cluster con respecto a otro o la dimensión de los datos analizados. De este modo se puede observar un comportamiento más global del método y se pueden contrastar los resultados obtenidos en el nuevo problema con los obtenidos previamente de un modo crítico, comprobando su coherencia.

4 Pruebas y resultados en problemas sintéticos

4.1 Introducción

Una vez introducidos: los conceptos clave para entender este proyecto, el planteamiento seguido para desarrollar los diferentes casos de prueba y el orden seguido en la ejecución de las pruebas, pasamos a explicar una a una y detalladamente las pruebas realizadas con problemas sintéticos, que son aquellos que han sido generados de manera artificial. Con este tipo de problemas conseguimos analizar el desempeño de los métodos en un entorno controlado, lo que simplifica el análisis por conocer las condiciones iniciales y el resultado esperado.

4.2 Análisis de problemas

4.2.1 Análisis en función de la separación de los clusters

4.2.1.1 Descripción del problema

El objetivo de esta prueba es ser un punto de partida del proceso, y para ello se intenta replicar el problema planteado en el artículo (16), cuyo objetivo es identificar el comportamiento de cada uno de los métodos a analizar, AIC, BIC y ΔJ, en función de la distancia que separa dos clusters. En este problema, a diferencia del problema del artículo, la cantidad de puntos de cada cluster es de 100 en lugar de 1000.

Para ello, se han generado un conjunto de problemas en dimensión dos, con dos clusters por problema. En estos problemas se varía la forma de los clusters y la distancia que los separa.

Para variar la forma, se han generado problemas cuyos clusters se obtienen mediante cuatro diferentes funciones de densidad de probabilidad, pero todos con la matriz de covarianzas igual a la identidad, $\Sigma = \mathrm{I}$. Los tipos de funciones utilizadas han sido las siguientes:

1. Distribución normal.
2. Distribución normal truncada. Donde los puntos que superan la distancia a la media en 4 veces la desviación estándar son descartados.

3. Distribución gamma-uniforme. Sigue una distribución gamma en el radio con un valor del parámetro de forma igual a 2 y una distribución uniforme en el ángulo.
4. Distribución uniforme en un círculo.

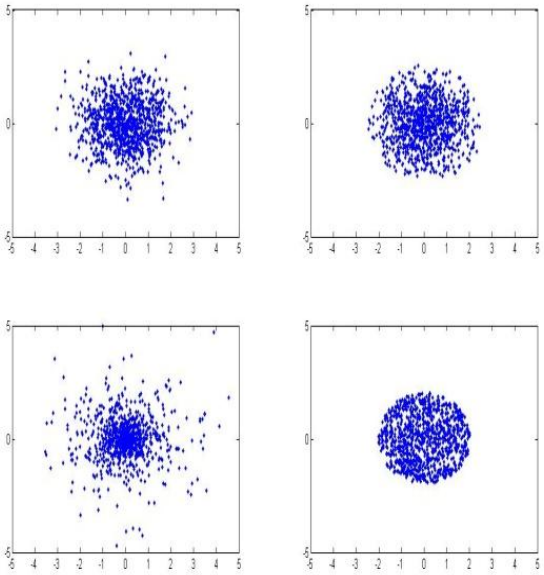


Figura 5. Ejemplos de los diferentes clusters utilizados en este problema. De izquierda a derecha y de arriba abajo: un cluster generado con una distribución normal, con una normal truncada, con una gamma-uniforme y con una uniforme en un círculo.

Cabe destacar que por cada dupla distancia-forma de cluster, se han generado 40 problemas diferentes. Cada uno de estos problemas se ha ajustado a una mezcla de $nc = \{1, 2, 3, 4, 5\}$ componentes gaussianas aplicando el método EM sobre los datos, y este proceso se ha repetido 10 veces por cada valor de n_c. Con estas condiciones iniciales se aplica el proceso descrito en la sección 3, obteniéndose por cada problema y por cada método, el número de clusters que contiene el modelo que cada método ha considerado óptimo de entre todos los devueltos.

Con los datos que se obtienen en todo el proceso, se generan gráficas que representan el número medio de clusters escogidos por cada índice a una determinada distancia. Hay 160 problemas diferentes por distancia, es decir 40 problemas por dupla distancia-forma de cluster, y 4 formas diferentes de clusters.
Además de las gráficas, se generan también unas tablas que contienen la media y la desviación estándar de los resultados por cada dupla distancia-método.
También se procesan los datos haciendo una distinción por la forma de los clusters del problema.
En la siguiente sección se encuentra todo el material aquí descrito.

4.2.1.2 Resultados y conclusiones

En la Figura 6 se representa el número medio de clusters en la mejor partición seleccionada por cada uno de los diferentes métodos: AIC, BIC, ΔJ_B, ΔJ_U, ΔJ_{US} y ΔJ_{UG}.

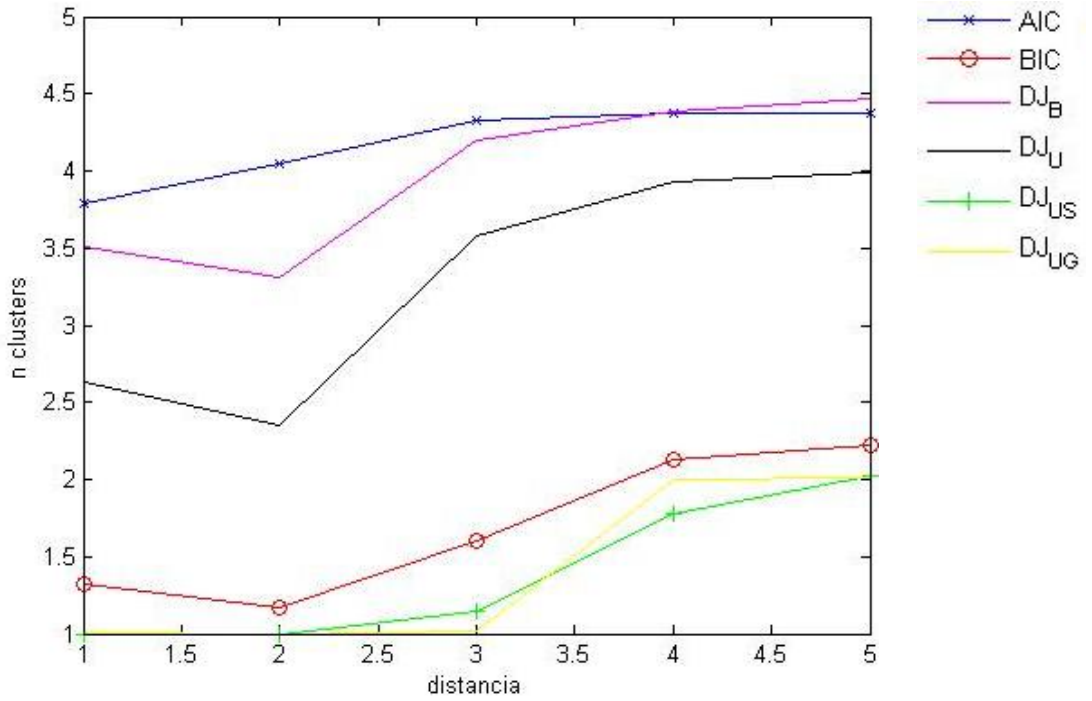


Figura 6. Media obtenida del número de clusters seleccionado por cada índice de validación para una determinada distancia.

Además de esta gráfica, es interesante mostrar la tabla con las medias y las desviaciones estándar de cada uno de los métodos, lo que nos da una idea más acertada del funcionamiento de cada uno.

Método\ Distancia	**1**	**2**	**3**	**4**	**5**
AIC	3.79 ± 1.40	4.04 ± 1.36	4.33 ± 0.93	4.37 ± 0.87	4.38 ± 0.96
BIC	1.32 ± 0.49	1.17 ± 0.42	1.61 ± 0.84	2.13 ± 0.36	2.22 ± 0.46
$\boldsymbol{\Delta J_B}$	3.51 ± 1.67	3.31 ± 1.76	4.20 ± 1.22	4.39 ± 0.86	4.47 ± 0.78
$\boldsymbol{\Delta J_U}$	2.63 ± 1.69	2.35 ± 1.70	3.58 ± 1.56	3.93 ± 1.16	3.99 ± 1.10
$\boldsymbol{\Delta J_{US}}$	1.00 ± 0.00	1.00 ± 0.00	1.14 ± 0.42	1.77 ± 0.45	2.02 ± 0.14
$\boldsymbol{\Delta J_{UG}}$	1.01 ± 0.08	1.00 ± 0.00	1.02 ± 0.14	2.00 ± 0.00	2.01 ± 0.11

Tabla 1. El primer valor de cada celda representa la media de clusters obtenida a partir los diferentes resultados obtenidos por uno de los métodos para una distancia dada, el segundo es la desviación estándar de los mismos resultados.

Si se observan los resultados de la Figura 6 y la Tabla 1, se pueden sacar las siguientes conclusiones:

1. Tanto AIC como ΔJ_B tienen un comportamiento muy similar, sobrestimando el número de clusters para todas las distancias.

2. ΔJ_U tiene un desempeño algo mejor que los métodos anteriores pero también tiende a sobrestimar el número de clusters.

3. DJ_{US}y DJ_{UG} subestima para distancias desde 1 hasta 3, es decir, cuando existe un mayor solape entre los clusters. Pero su desempeño mejora enormemente cuando supera esa distancia, llegando en el caso de ΔJ_{UG} a acertar en la mayoría de los problemas. Esto se ve por la desviación estándar tan pequeña que presenta para distancias 4 y 5.

4. BIC al igual que los dos métodos anteriores, comienza subestimando para distancias de 1 a 3. Y a partir de 3, aunque se aproxima mucho a la solución, tiene tendencia a sobrestimar el número de clusters, obteniendo un desempeño peor que ΔJ_{UG}.

Este resultado sirve para ver el comportamiento global cuando entre los problemas hay clusters generados por diferentes distribuciones, aportando una idea general del desempeño de cada uno de los métodos en función de la distancia.

Para ver el comportamiento que tiene cada uno de los diferentes métodos en función del tipo de cluster y la distancia, se van a presentar ahora los mismos resultados realizando sobre ellos un tratamiento distinto al anterior. El enfoque a seguir es muy similar pero se muestran los resultados separados en función de la forma de los clusters que componen el problema. Este modo de proceder permite comprobar el desempeño de los diferentes métodos en función de la forma de los clusters y de la distancia de separación entre ellos, permitiendo identificar la idoneidad de cada método en función de la forma de los clusters que contiene el problema a analizar.

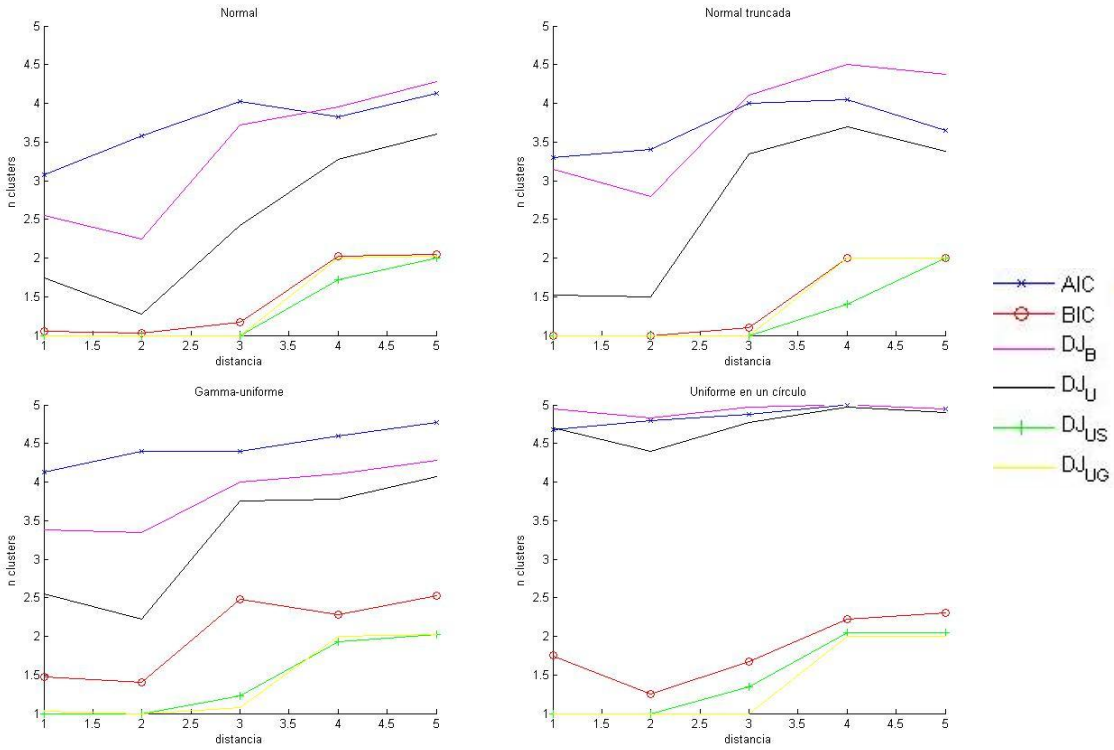


Figura 7. Resultados de los diferentes métodos en función a la forma de los clusters que conforman el problema.

Normal

Método\ Distancia	1	2	3	4	5
AIC	3.07 ± 1.55	3.57 ± 1.59	4.02 ± 1.02	3.82 ± 0.95	4.12 ± 1.01
BIC	1.05 ± 0.31	1.02 ± 0.15	1.17 ± 0.38	2.02 ± 0.15	2.05 ± 0.22
$\mathbf{\Delta J_B}$	2.55 ± 1.66	2.25 ± 1.58	3.73 ± 1.47	3.95 ± 1.01	4.28 ± 0.85
$\mathbf{\Delta J_U}$	1.75 ± 1.26	1.27 ± 0.72	2.42 ± 1.55	3.27 ± 1.18	3.60 ± 1.15
$\mathbf{\Delta J_{US}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.73 ± 0.45	2.00 ± 0.00
$\mathbf{\Delta J_{UG}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	2.02 ± 0.16

Tabla 2. Media y desviación estándar del número de clusters escogidos por cada método para una distancia determinada cuando los clusters tienen una distribución normal.

Normal Truncada

Método\ Distancia	1	2	3	4	5
AIC	3.30 ± 1.57	3.40 ± 1.67	4.00 ± 1.08	4.05 ± 0.93	3.65 ± 1.16
BIC	1.00 ± 0.00	1.00 ± 0.00	1.10 ± 0.30	2.00 ± 0.00	2.00 ± 0.00
$\mathbf{\Delta J_B}$	3.15 ± 1.86	2.80 ± 1.84	4.10 ± 1.32	4.50 ± 0.75	4.38 ± 0.81
$\mathbf{\Delta J_U}$	1.52 ± 1.18	1.50 ± 1.22	3.35 ± 1.72	3.70 ± 1.20	3.38 ± 1.10
$\mathbf{\Delta J_{US}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.40 ± 0.50	2.00 ± 0.00
$\mathbf{\Delta J_{UG}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	2.00 ± 0.00

Tabla 3. Media y desviación estándar del número de clusters escogidos por cada método para una distancia determinada cuando los clusters tienen una distribución normal truncada.

Gamma-uniforme

Método\ Distancia	1	2	3	4	5
AIC	4.12 ± 1.06	4.40 ± 0.77	4.40 ± 0.81	4.60 ± 0.67	4.77 ± 0.47
BIC	1.47 ± 0.50	1.40 ± 0.63	2.47 ± 0.84	2.27 ± 0.45	2.52 ± 0.50
$\mathbf{\Delta J_B}$	3.38 ± 1.37	3.35 ± 1.64	4.00 ± 1.13	4.10 ± 0.87	4.28 ± 0.85
$\mathbf{\Delta J_U}$	2.55 ± 1.32	2.23 ± 1.54	3.75 ± 1.19	3.77 ± 0.97	4.08 ± 0.92
$\mathbf{\Delta J_{US}}$	1.00 ± 0.00	1.00 ± 0.00	1.23 ± 0.48	1.93 ± 0.27	2.02 ± 0.16
$\mathbf{\Delta J_{UG}}$	1.02 ± 0.16	1.00 ± 0.00	1.08 ± 0.27	2.00 ± 0.00	2.02 ± 0.16

Tabla 4. Media y desviación estándar del número de clusters escogidos por cada método para una distancia determinada cuando los clusters tienen una distribución gamma-uniforme.

Uniforme en un círculo

Método\ Distancia	**1**	**2**	**3**	**4**	**5**
AIC	4.67 ± 0.52	4.80 ± 0.40	4.87 ± 0.33	5.00 ± 0.00	4.95 ± 0.22
BIC	1.75 ± 0.49	1.25 ± 0.43	1.67 ± 0.82	2.22 ± 0.47	2.30 ± 0.60
$\mathbf{\Delta J_B}$	4.95 ± 0.22	4.83 ± 0.50	4.97 ± 0.16	5.00 ± 0.00	4.95 ± 0.22
$\mathbf{\Delta J_U}$	4.70 ± 0.69	4.40 ± 1.06	4.78 ± 0.48	4.97 ± 0.16	4.90 ± 0.38
$\mathbf{\Delta J_{US}}$	1.00 ± 0.00	1.00 ± 0.00	1.35 ± 0.62	2.05 ± 0.22	2.05 ± 0.22
$\mathbf{\Delta J_{UG}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	2.00 ± 0.00	2.00 ± 0.00

Tabla 5. Media y desviación estándar del número de clusters escogidos por cada método para una distancia determinada cuando los clusters tienen una distribución uniforme en un círculo.

Como se puede observar en la Figura 7 y en las tablas 2, 3, 4 y 5, todos los métodos excepto BIC se comportan de manera muy similar, independientemente de la forma del cluster que contenga el problema a validar.

Pasemos a analizar los resultados por cada método.

AIC tiende a sobrestimar el número de clusters en todos los problemas. Aunque se puede observar que para clusters normales y normales truncados tiene un comportamiento algo mejor que con respecto a clusters del tipo gamma-uniforme y uniformes en el círculo, en cuyo caso el desempeño es pésimo, sobrestimando el número de clusters en la totalidad de los problemas.

Los métodos ΔJ_B y ΔJ_U tienden a sobrestimar el número de clusters excepto en los problemas con clusters gaussianos y gaussianos truncados, en los cuales ΔJ_U subestima para distancias menores o iguales que 2. Aunque su funcionamiento queda lejos de ser el mejor de todos los métodos analizados, sí que ofrecen un desempeño marcadamente mejor que AIC para todos los tipos de problemas.

BIC tiene un funcionamiento muy positivo para los problemas en los que los clusters son gaussianos o gaussianos truncados y la distancia que los separa es mayor que 3. Para distancias entre 1 y 3 tiende a subestimar. Sin embargo, para los otros dos tipos de clusters que se han analizado, gamma-uniforme y uniforme en un círculo, no obtiene un desempeño tan positivo, lo que es de esperar pues es un método optimizado para el problema gaussiano.

ΔJ_{US} se comporta bien para el caso gaussiano, se observa como necesita de una mayor distancia de separación de la que necesita BIC o ΔJ_{UG} para tener un desempeño bueno. Hasta que la distancia no es igual a 5, el método tiende a subestimar, si bien se nota una ligera mejoría cuando la distancia es igual a 4.
Para los problemas con clusters de tipo gamma-uniforme y uniforme en un círculo tiene un desempeño muy positivo para distancias 4 y 5, mejorando lo obtenido por BIC y prácticamente igualando a ΔJ_{UG}.

ΔJ_{UG} por su parte, parece ser el método que tiene un comportamiento más estable frente a la variación de la forma del cluster del problema, pues tiene el mismo comportamiento para todos los casos, subestima hasta distancia 3 incluida, y a partir de distancia 4 comienza a tener un desempeño fabuloso.

Debido a que ΔJ_B y ΔJ_U se comportan considerablemente peor que ΔJ_{US} y ΔJ_{UG}, de ahora en adelante no se van a tener en cuenta en los análisis.

Una vez se ha analizado el comportamiento de los diferentes métodos de validación teniendo en cuenta la distancia, vamos a continuar con los análisis de los métodos elegidos haciendo un análisis más exhaustivo de su comportamiento en función de la distancia y además, se van a añadir más variables a los problemas, lo que nos permitirá tener una visión más global del comportamiento de los diferentes métodos.

4.2.2 Análisis del algoritmo frente a la variación de la distancia de separación entre clusters y el grado de normalidad de los mismos.

4.2.2.1 Introducción

En el análisis anterior, se han sacado algunas conclusiones con respecto al comportamiento de los métodos en función de la distancia, pero únicamente se estaba analizando a cinco distancias diferentes lo que no nos permitía observar la frontera real a la cual cada método pasaba de tener un desempeño negativo a uno positivo. En este análisis vamos a hacer un análisis más en profundidad del comportamiento de los métodos frente a la distancia. Además, también se va a analizar el comportamiento de los métodos cuando analizan clusters en un rango de formas mucho más amplio que en el anterior. Por lo tanto, este problema se puede tomar como una ampliación del problema previo, añadiendo además el análisis en diferentes dimensiones.

4.2.2.2 Descripción del problema

Para conseguir lo descrito anteriormente en la introducción, se ha generado un problema *d*-dimensional de dos clusters, donde cada punto de los clusters se extrae de una función de densidad de probabilidad que varía desde una delta hasta una uniforme, pasando por el caso gaussiano.
Para ello se utiliza un proceso en el que la función de densidad de probabilidad depende de lo que llamamos exponente k, que será además el indicador que tomaremos para cuantificar el grado de normalidad de los clusters.

Para entender mejor el significado del exponente k, a continuación se explica el proceso de generación de los clusters:

1. Se generan los n puntos mediante una distribución normal en dimensión d y se normalizan para obtener puntos que estén uniformemente distribuidos en una esfera *d-1*.

2. Se genera para cada punto su radio de acuerdo a una distribución que tiene la siguiente forma

$$f(r) = Cr^{d-1}exp\,(-r^k)$$

Donde r es el radio, d la dimensión de los datos, C una constante tal que cumple que la integral de $f(r)$ sea la unidad y k es el único parámetro libre, el cual hará que nuestra función densidad de probabilidad total tenga una u otra forma.

3. Finalmente se aplica la transformación de Whitening (23) para conseguir que la matriz de covarianza de estos datos sea la identidad $\Sigma = I$.

Se puede demostrar que en función del exponente k, la función de densidad de probabilidad que genera los clusters varía entre una delta cuando $k \to 0$, y una distribución uniforme cuando $k \to \infty$, pasando por una distribución normal para el caso de $k = 2$.
También puede demostrarse que para el caso de $k = 1$, la función densidad de probabilidad que genera el cluster es una gamma con parámetro de forma igual a la dimensión d, y parámetro de escala igual a 1. Es decir, exactamente igual que la función gamma-uniforme utilizada en el problema anterior, pero únicamente para el caso $d = 2$.
Esto nos permitirá contrastar los resultados de este problema con el anterior y comprobar si son consistentes.

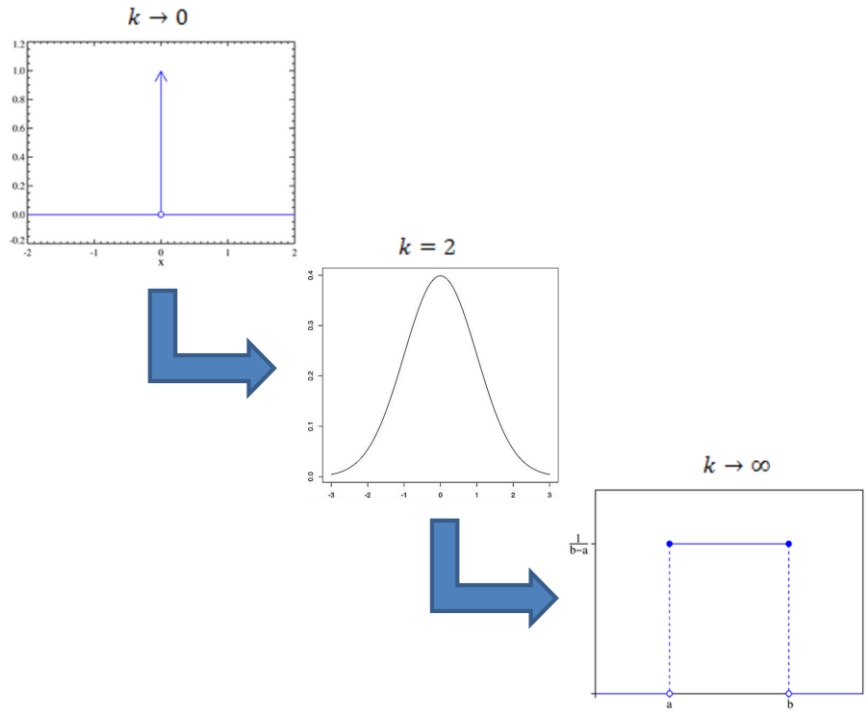


Figura 8. Progresión en la forma de la fdp generada en función del exponente ***k***

Mediante este exponente k podemos definir lo que llamamos *grado de normalidad*, pues cuanto más cerca se encuentre el exponente del valor 2, más normal será la función. En la Figura 8 se puede observar la progresión en la forma de la fdp generada por la función según varía el exponente k, y en la Figura 9 la distribución de los puntos generados para diferentes valores de k en el rango que se van a encontrar en este análisis.

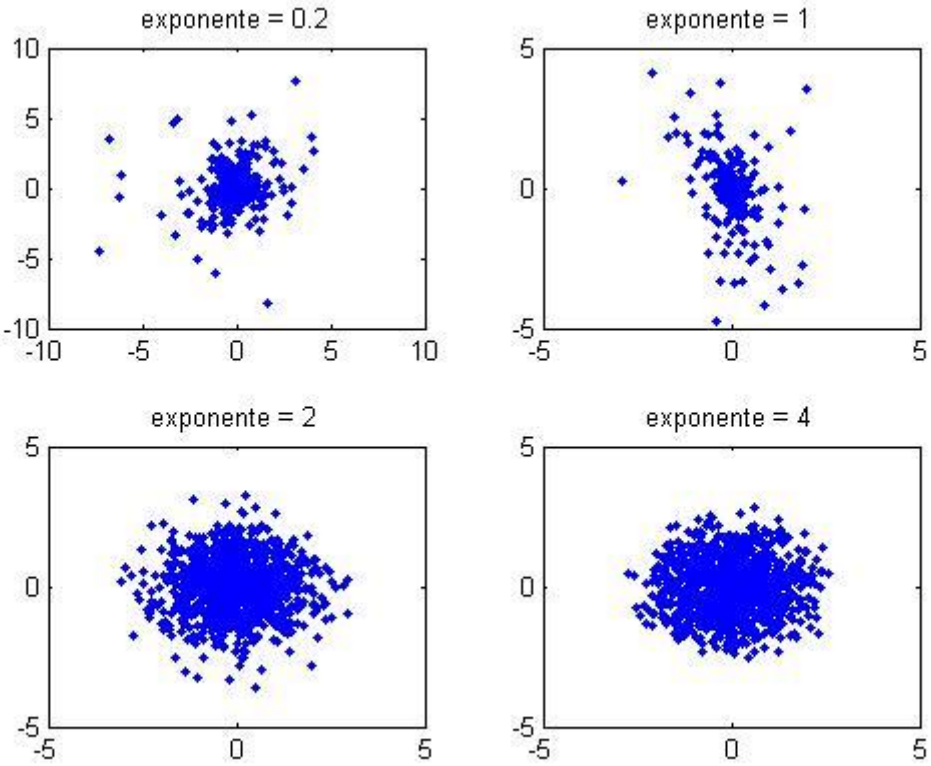


Figura 9. Distribución de los puntos de un cluster en dimensión 2 y en función del exponente $\boldsymbol{k}$ o grado de normalidad.

Una vez se tienen los clusters generados, se separan los puntos del segundo cluster una distancia a, quedando la media del cluster 1 en $c_1 = (0, \dots, 0)$ y para el cluster 2 en $c_2 = (a, 0, \dots, 0)$.

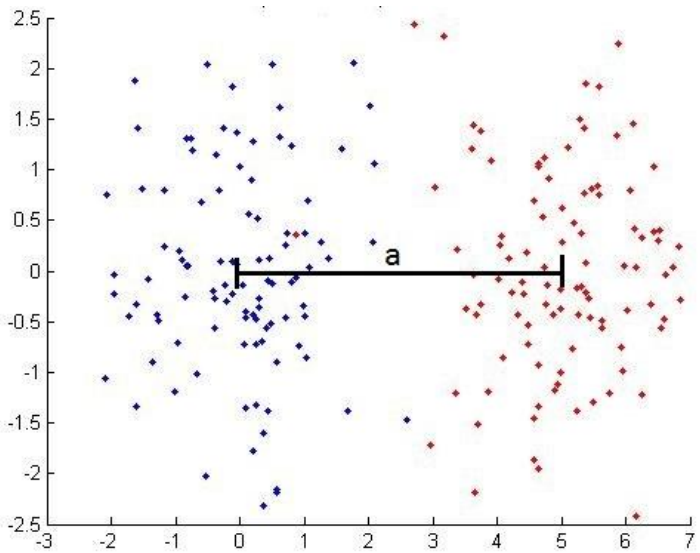


Figura 10. Ejemplo de dos clusters en dimensión 2, con una separación $\boldsymbol{a = 5}$. $\boldsymbol{c_1 = (0, 0)}$ y $\boldsymbol{c_2 = (5, 0)}$

Variando estos dos parámetros, a y k, se ha generado una batería de diferentes problemas por cada dimensión d = {2, 3, 4, 10}. La distancia a se ha variado desde 0 hasta 5 en intervalos de 0.2 y el exponente k desde 0.2 hasta 4 en intervalos de 0.2.

Hay que especificar también que el número de puntos por cluster para cada dimensión varía según la dimensión, siendo 100 para $d = 2$, 1000 para $d = 3$ y 10000 para $d = 4$ y $d = 10$.

Al de ajustar los datos a mezclas de distribuciones mediante el método EM, se ha hecho con $n_c = \{1, 2, 3, 4\}$. Se ha eliminado el ajuste a 5 componentes, pues por los resultados que se han obtenido en el problema anterior no cabe esperar que este ajuste tenga mucha relevancia en los resultados, además de que reduce el coste computacional considerablemente. Por cada ajuste a un n_c distinto, se ha ejecutado 10 veces el método EM.

Hay que destacar que por cada dupla exponente-distancia se han generado 20 problemas diferentes, por lo tanto, el proceso descrito en la sección 3 se ejecutará sobre cada uno de estos problemas.

Con los datos obtenidos se realizan gráficas en tres dimensiones: en el eje x se representa lo que hemos llamado exponente k, que da el grado de normalidad de la fdp que genera los puntos de los clusters; en el eje y se representa la distancia que separa los dos clusters; y el tercer eje es una escala de color, que representa el número medio de clusters que ha detectado el método para una determinada dupla. Esta dupla se corresponde con una coordenada (x, y). El valor para esta coordenada es el número medio de clusters que ha escogido cada método en los 20 problemas diferentes que contiene una dupla distancia-exponente.

4.2.2.3 Resultados

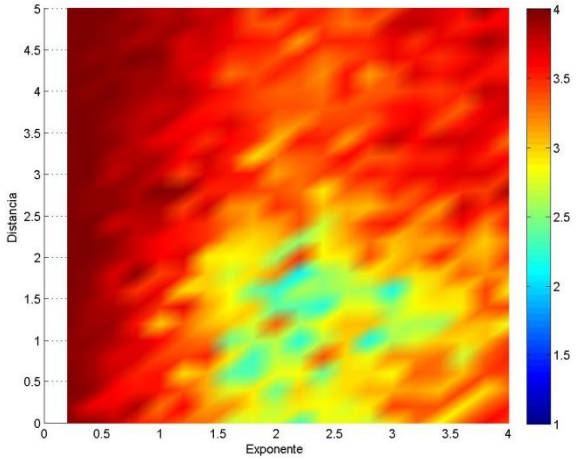
A continuación se muestran las figuras generadas para las diferentes dimensiones a partir de los resultados obtenidos con los métodos AIC, BIC y ΔJ_{UG}. De los métodos basados en el incremento de negentropía de las particiones solo se muestra el enfoque ΔJ_{UG}, pues es el que mejor desempeño ha obtenido y de esta forma se presenta solo la información más relevante facilitando su análisis. Se pueden consultar los resultados obtenidos para ΔJ_{US} en el anexo A.

Los resultados se van a mostrar por dimensión, así, cada una de las siguientes figuras se corresponde con el resultado de un determinado índice de validación en una dimensión dada.

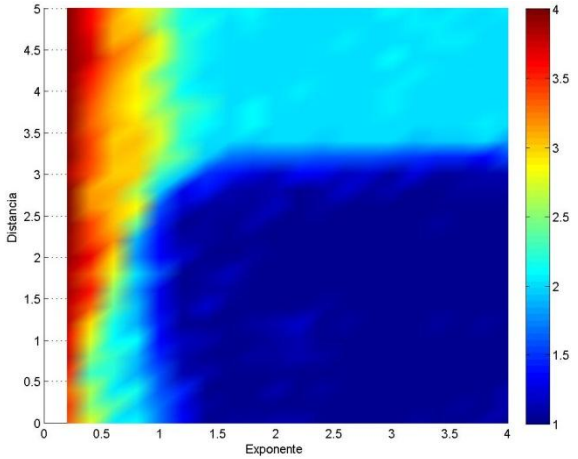
Hay que destacar que el corte que se obtiene en las gráficas para la dimensión $d = 2$, cuando el exponente $k = 1$ debe corresponderse con las gráficas del análisis anterior para los problemas generados mediante la función gamma–uniforme. Y el corte para $k = 2$ debe corresponderse con las gráficas del análisis previo para problemas generados mediante la distribución normal.

Dimensión 2

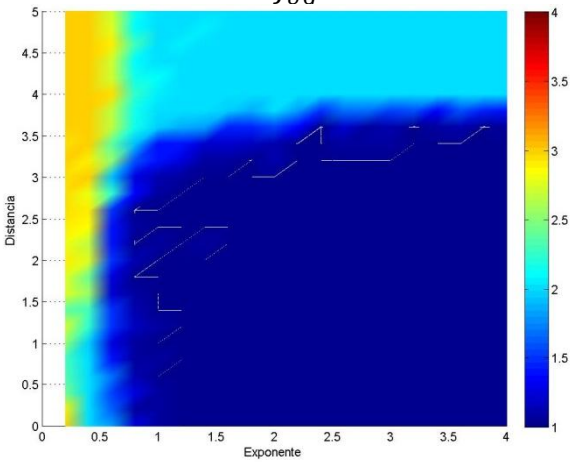
AIC



BIC

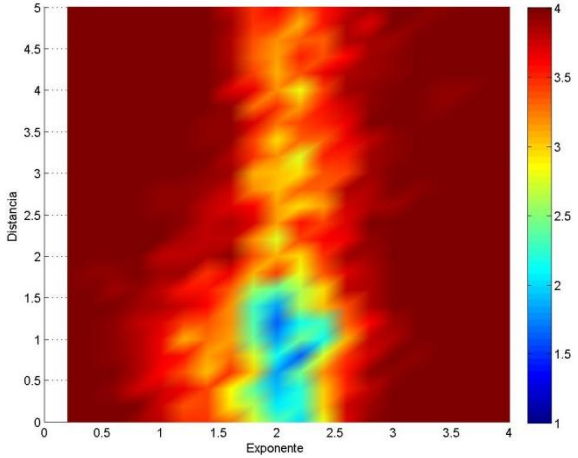


ΔJ_{UG}

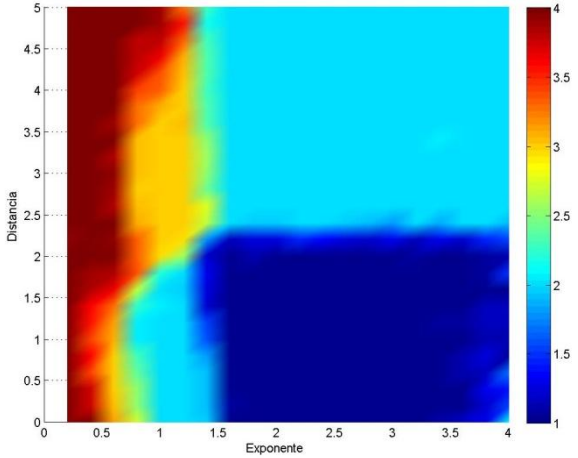


Dimensión 3

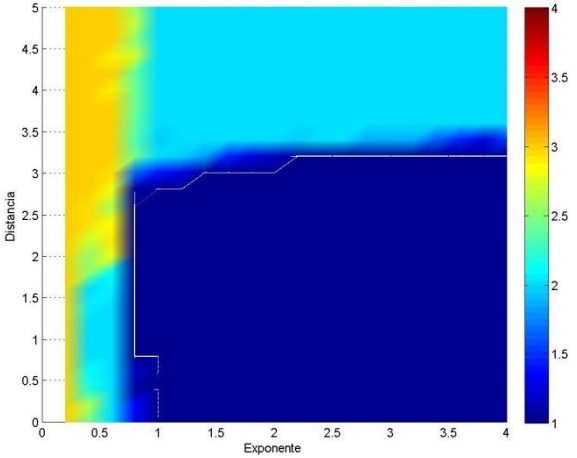
AIC



BIC

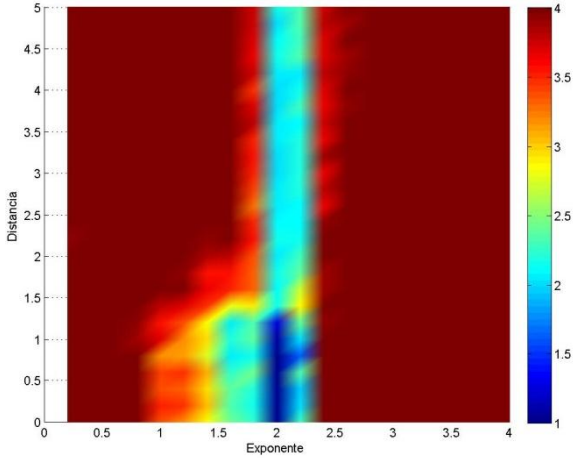


ΔJ_{UG}

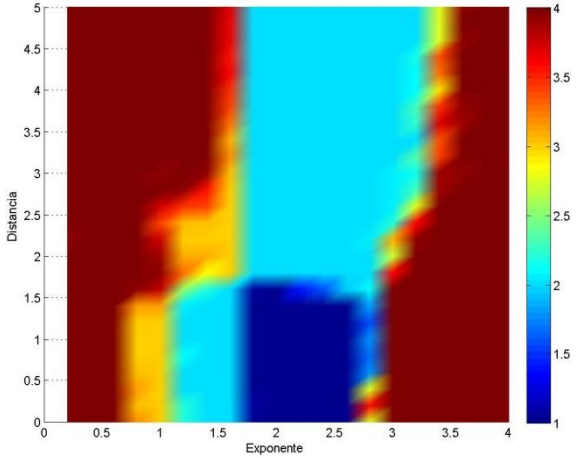


Dimensión 4

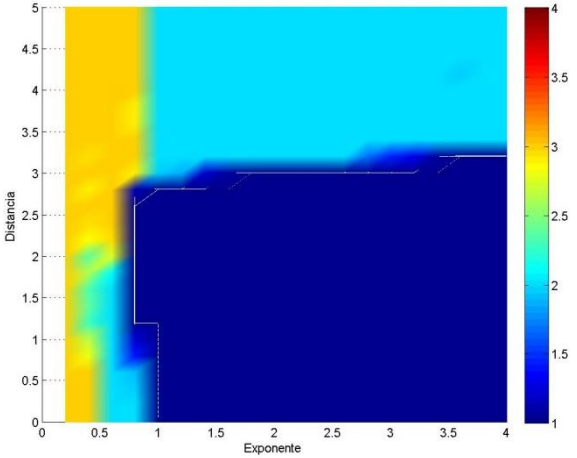
AIC



BIC

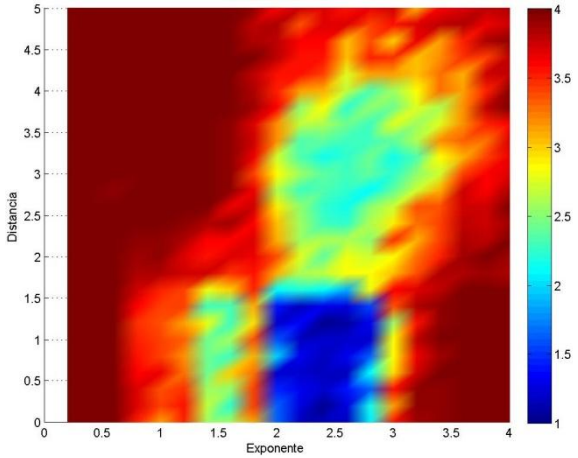


ΔJ_{UG}

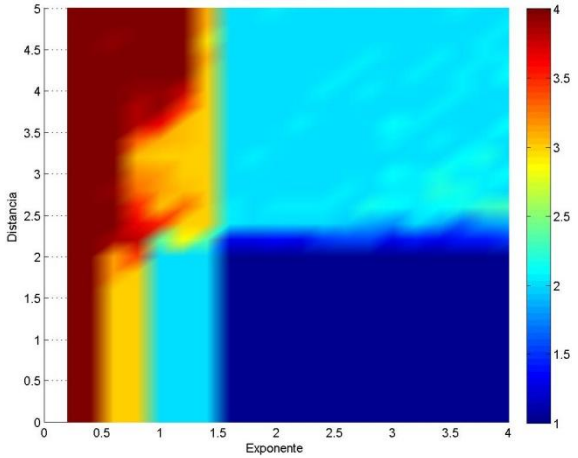


Dimensión 10

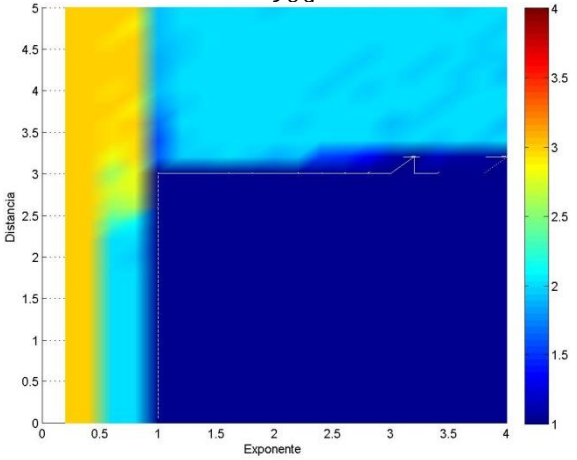
AIC



BIC



ΔJ_{UG}



4.2.2.4 Conclusiones

Al igual que en el análisis anterior, AIC tiende a escoger la solución con el mayor número de clusters, es decir, tiende a sobrestimar, aunque se nota una cierta mejoría cuando el exponente está cercano a 2, que se corresponde con clusters gaussianos. Según aumenta la dimensión hasta 4, se aprecia cómo para exponentes muy cercanos a 2, el método se va comportando mejor.

BIC y ΔJ_{UG} tienen un rango de acierto similar para todas las dimensiones. Se puede observar como BIC tiene una tolerancia mayor que ΔJ_{UG} al solape, mientras que por su parte ΔJ_{UG} tiene mayor tolerancia al grado de normalidad de la fdp que genera los datos. Hay que destacar que para el rango del exponente donde BIC y ΔJ_{UG} sobrestiman el número de clusters, ΔJ_{UG} tiende a escoger ajustes con un número de componentes más cercano a la solución de lo que lo hace BIC.

Centrando la atención en el comportamiento frente a la dimensión, parece que al aumentarla se mejora el umbral de distancia de separación entre clusters al cual los métodos comienzan a acertar. Sin embargo, un aumento en la dimensión parece empeorar el rango de exponente que acepta cada método sobrestimando en estos puntos, este hecho se acentúa más en BIC que en ΔJ_{UG}.

Se puede observar en AIC y BIC, aunque en BIC se hace más notorio, como para dimensión 4 cuando el exponente está cercano a 2 tienen un mejor desempeño. Sin embargo, cuando el exponente se aleja de la normalidad comienza a sobrestimar el número de clusters.

En dimensión 10 se puede ver como la sobrestimación en el número de clusters que se observa para dimensión 4 no se produce. Esto se debe a las características del problema: se ha saltado de dimensión 4 a dimensión 10, manteniéndose el número de puntos constante en 10000. A continuación se explica más en detalle.

Recordemos que BIC, definido por la expresión [13], disminuye con la verosimilitud y aumenta con el término de penalización $c \ln(n)$. Teniendo en cuenta que para escoger entre un conjunto de modelos usando BIC, se escogerá siempre el modelo cuyo valor del índice sea menor, vamos a ver cómo afecta cada variable al índice para poder sacar la conclusión de lo que está sucediendo.

- Un aumento de puntos se traduce en un aumento de la verosimilitud, pues a mayor número de muestras, el valor de los parámetros estimados de cada componente es más cercano al real. En nuestro caso, el número de puntos se mantiene constante, lo que no aporta mejora a la verosimilitud.

- Un aumento de dimensión provoca que haya que estimar más parámetros por cada componente. Por ejemplo, si en dimensión 4 hay que estimar 4 medias por componente, en dimensión 10 habrá que estimar 10 medias por componente. Por lo tanto, este aumento de dimensión provoca un aumento en el término de penalización de BIC, pues depende del número de parámetros a estimar c, y una disminución en la verosimilitud, pues se necesitan más puntos porque se tienen más parámetros a estimar.

- Un aumento del número de componentes de ajuste se traduce: por un lado en una mayor verosimilitud pues cuantas más componentes se usen mejor será el ajuste; y por otro lado, en un aumento en el término de penalización pues es necesario calcular los parámetros de cada una de las componentes.

Ahora dejando la dimensión constante y el número de puntos constante que es el caso que nos ocupa, el valor de BIC vendrá determinado por el número de componentes:

- Un aumento en el término verosimilitud se produce por un aumento en el número de componentes de ajuste.
- Un aumento del término de penalización se produce por un aumento en el número de componentes de ajuste.

Lo que ha sucedido en dimensión 10 teniendo 10000 puntos por cluster, es que se ha disminuido el ratio número de puntos frente a dimensión con respecto a dimensión 4 debido al aumento de la dimensión. Esto ha provocado que para obtener un valor de BIC bajo sea necesario escoger el modelo en el que se ha ajustado con pocas componentes o clusters, pues introducen una menor penalización.

Así, es como se explica que en dimensión 10 esté acertando para valores del exponente donde en dimensión 4 estaba fallando, pues en dimensión 4 al tener que estimar menos parámetros por ajuste que en dimensión 10, el valor de BIC es menor cuando se ajusta con más componentes, pues la mejora que produce este aumento en la verosimilitud afecta más al índice de lo que lo hace la penalización introducida. En dimensión 10 sin embargo, se produce el efecto contrario, al aumentar el número de componentes el término de penalización se ve más afectado de lo que lo hace el término verosimilitud, lo que se traduce en valores de BIC menores a menor número de componentes.

4.2.3 Análisis del algoritmo frente a la variación del número de puntos de los clusters y el grado de normalidad de los mismos.

4.2.3.1 Introducción

Este nuevo problema que se plantea es muy similar al planteado en el punto anterior, con la diferencia de que solo se van a analizar tres distancias diferentes. Estas distancias se dejarán fijas a lo largo de todas las pruebas para una misma dimensión.
Los parámetros que vamos a variar son el número de puntos de uno de los clusters y el grado de normalidad de los clusters.

Para generar los clusters se va a utilizar la misma función que se utiliza en el punto anterior, dependiendo su grado de normalidad del exponente k. Además, la cantidad de puntos de uno de los clusters es un porcentaje de la cantidad del otro, lo que nos proporciona problemas con diferente grado de desbalanceo entre los clusters.

4.2.3.2 Descripción del problema

Basándonos en los resultados del punto anterior, hemos elegido tres distancias de separación de los clusters con las cuales realizar el análisis. Estas distancias varían con la dimensión, la cual toma los mismos valores que en el punto anterior, $d = \{2, 3, 4, 10\}$, y coinciden para cada dimensión con:

- Distancia 1: la distancia donde BIC falla
- Distancia 2: la frontera donde BIC pasa de fallar a acertar
- Distancia 3: la distancia donde BIC acierta

Se puede ver gráficamente en la Figura 11 lo que estas distancias representan.

Los valores de estas distancias escogidos para cada dimensión están representados en la siguiente tabla:

Dimensión \ punto	**Distancia 1**	**Distancia 2**	**Distancia 3**
2	3	3.5	4
3	2	2.5	3.5
4	1.5	2	3.5
10	1.5	2.5	3.5

Tabla 6. Representa las tres distancias escogidas en función de la dimensión del problema. Distancia 1 es la distancia donde BIC falla, distancia 2 es la frontera entre el fallo y el acierto de BIC, y distancia 3 es la distancia a la cual BIC acierta.

La cantidad de puntos de uno de los clusters se escoge en función de un porcentaje de la cantidad del otro, y para todas las dimensiones ese porcentaje es el mismo. Se comienza con el 100% de los puntos, y va descendiendo hasta el 10% de los puntos en pasos de 10%, esto es: $\%\ de\ puntos = \{100\%, 90\%, \dots, 20\%, 10\%\}$.

La cantidad de puntos para cada dimensión del cluster con número de puntos constante es idéntica al problema anterior. Es decir, para dimensión 2 está compuesto por 100 puntos, para dimensión 3 por 1000 y para dimensión 4 y dimensión 10 por 10000. Esto hace que la cantidad de puntos del cluster de tamaño variable para dimensión 2 varíe desde 100 puntos hasta 10 puntos; en dimensión 3 desde 1000 puntos hasta 100 puntos, y en dimensiones 4 y 10 desde 10000 hasta 1000 puntos.

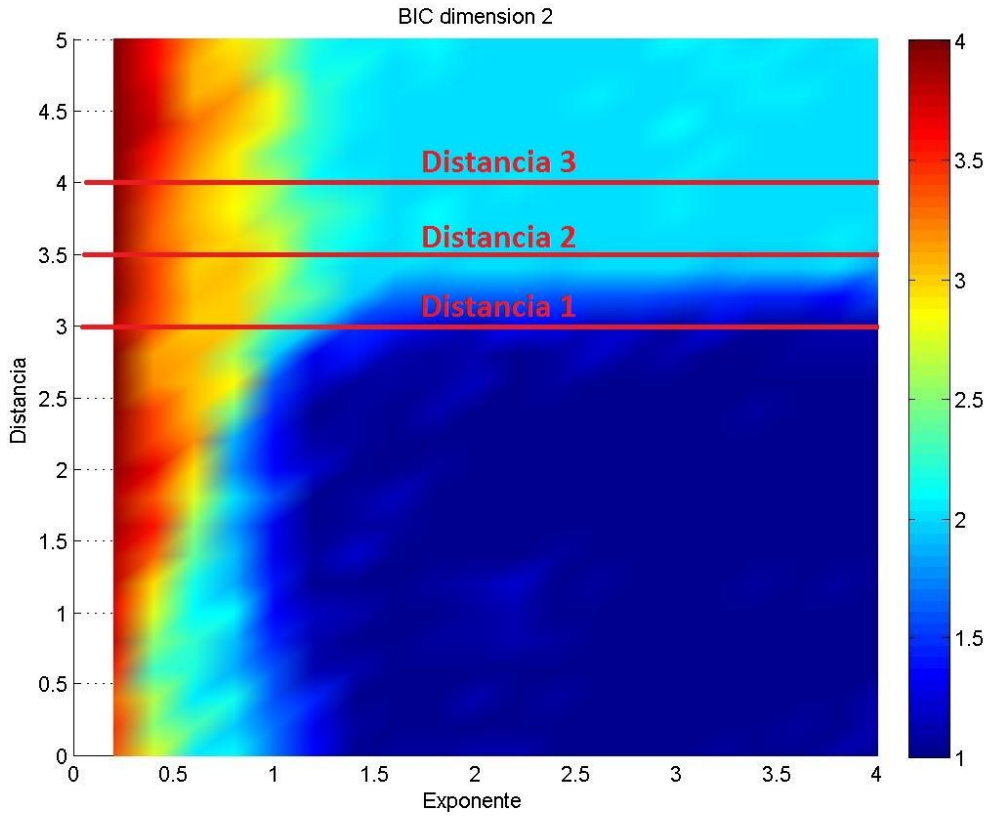


Figura 11. Ejemplo de método de selección de distancias para dimensión 2. De arriba abajo sería: distancia donde BIC falla (distancia 1), distancia frontera entre acierto y fallo (distancia 2) y distancia a la cual BIC acierta (distancia 3)

A la hora de ajustar los datos a mezclas de distribuciones mediante el método EM se ha seguido el mismo enfoque que en el problema anterior, $n_c = \{1, 2, 3, 4\}$. Por cada ajuste a un n_c distinto, se ha ejecutado 10 veces el método EM.

Hay que destacar que por cada dupla exponente-desbalanceo se han generado 20 problemas diferentes, por lo tanto, el proceso descrito en la sección 3 se ejecutará sobre cada uno de estos problemas.

Con los datos obtenidos se realizan unas gráficas en tres dimensiones muy similares a las del punto anterior: en el eje x se representa el exponente; en el eje y se representa la proporción de puntos que tiene el cluster de tamaño variable con respecto al fijo (desbalanceo); y el tercer eje es una escala de color que representa el número medio de clusters que ha detectado el método para una determinada dupla exponente-desbalanceo. Esta dupla se corresponde con una coordenada (x, y). El valor para esta coordenada es el número medio de clusters que se ha detectado en los 20 problemas diferentes que contiene una dupla exponente-desbalanceo para una dimensión y una distancia dentro de esa dimensión.

4.2.3.3 Resultados

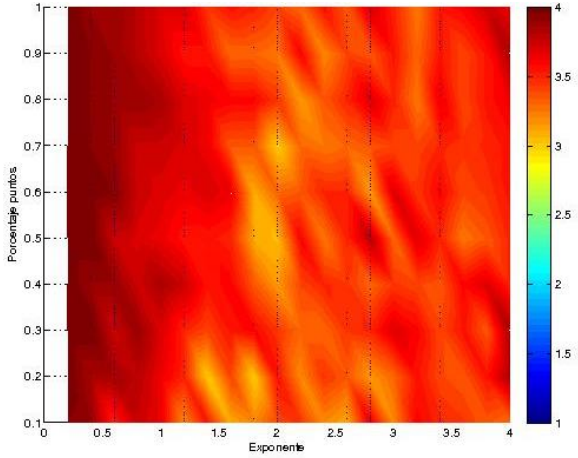
A continuación se muestran las figuras generadas para las diferentes dimensiones a partir de los resultados obtenidos con los métodos AIC, BIC y ΔJ_{UG}. Al igual que en el problema anterior, de los métodos de validación basados en la negentropía de las particiones solo se muestra el resultado obtenido para el enfoque ΔJ_{UG}, pues los resultados para ΔJ_{US} son peores y entorpecerían el análisis. Estos resultados se pueden consultar en el anexo B.

Los resultados se van a mostrar por dimensión, y dentro de cada dimensión se mostrarán por método de validación. Así, cada una de las siguientes figuras se corresponde con el resultado de un determinado índice de validación, a una distancia determinada, en una dimensión dada.

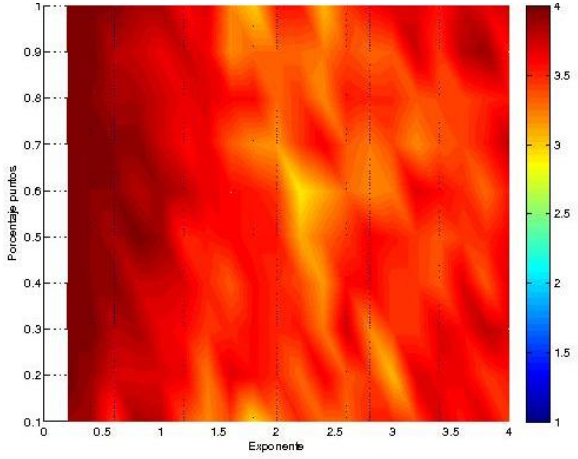
Hay que destacar que el corte de cada una de estas figuras cuando el número de puntos del cluster de tamaño variable es el 100% del fijo, debe corresponderse con el corte a cada una de las distancias de las figuras del problema anterior.

Dimensión 2, AIC

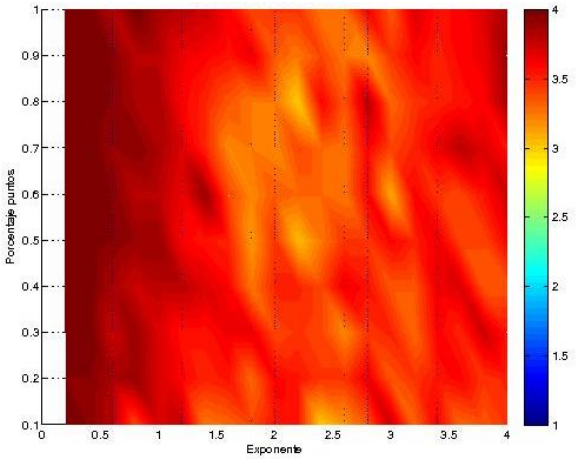
Distancia 1



Distancia 2

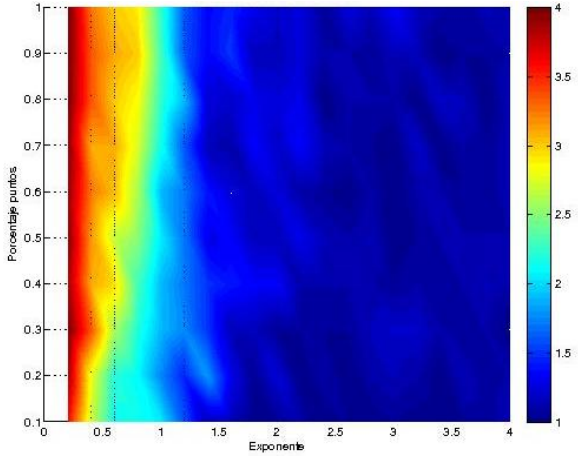


Distancia 3

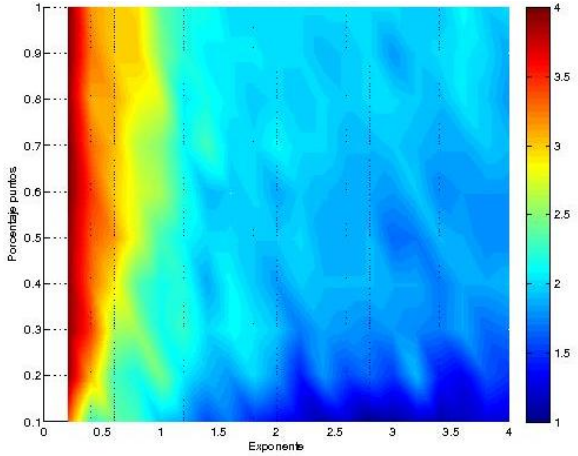


Dimensión 2, BIC

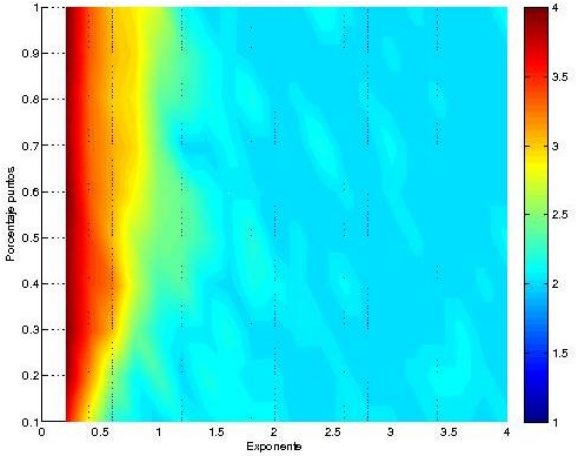
Distancia 1



Distancia 2

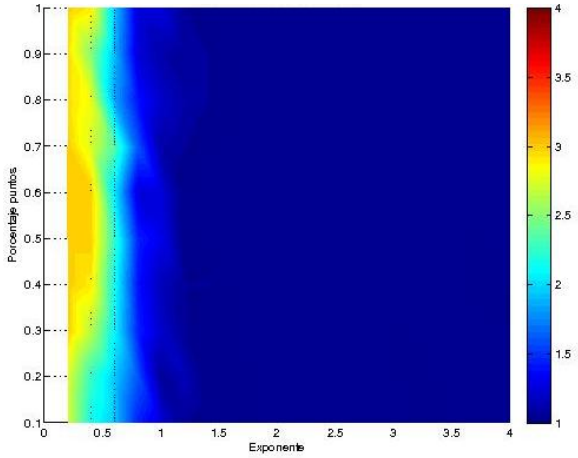


Distancia 3

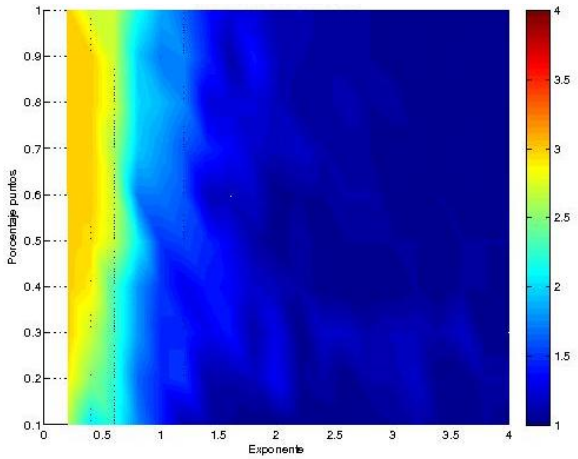


Dimensión 2, ΔJ_{UG}

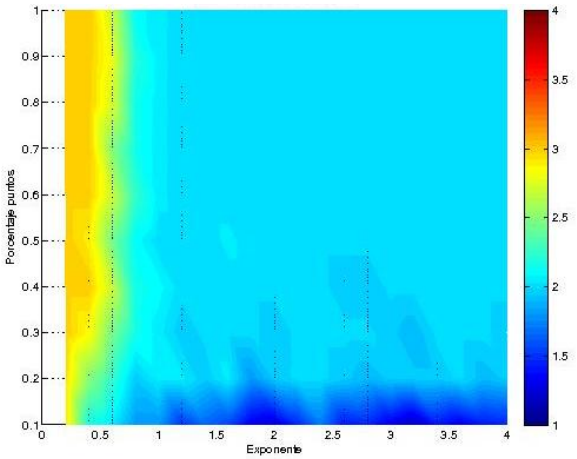
Distancia 1



Distancia 2

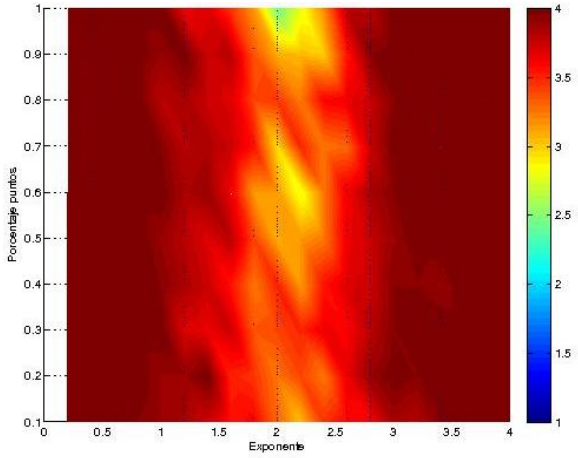


Distancia 3

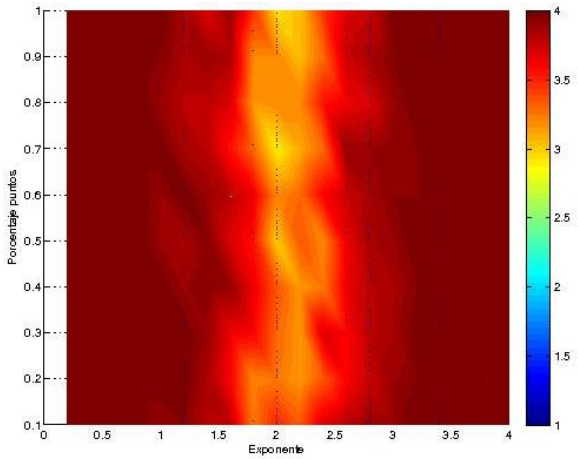


Dimensión 3, AIC

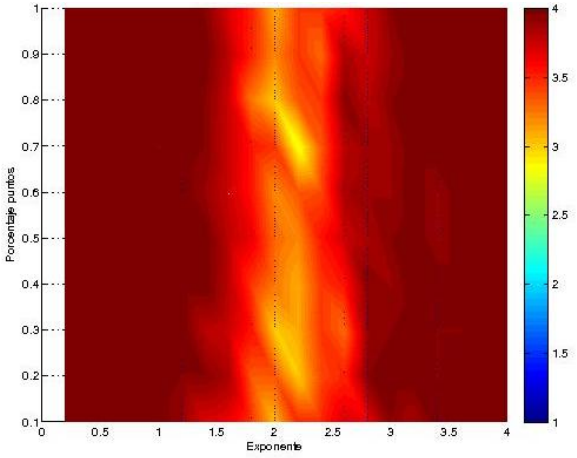
Distancia 1



Distancia 2

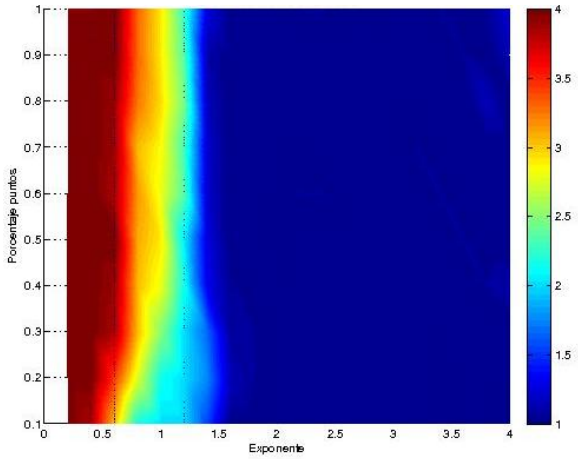


Distancia 3

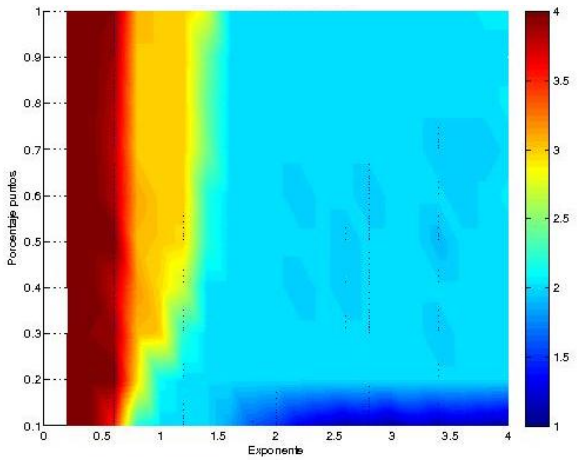


Dimensión 3, BIC

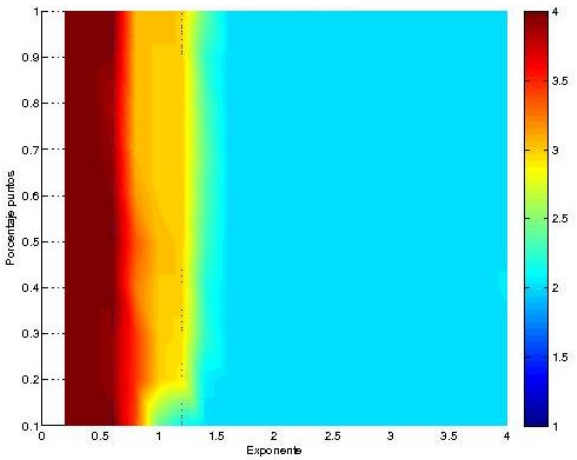
Distancia 1



Distancia 2

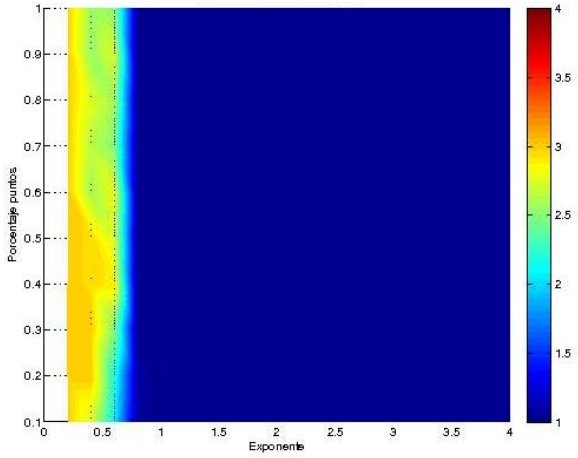


Distancia 3

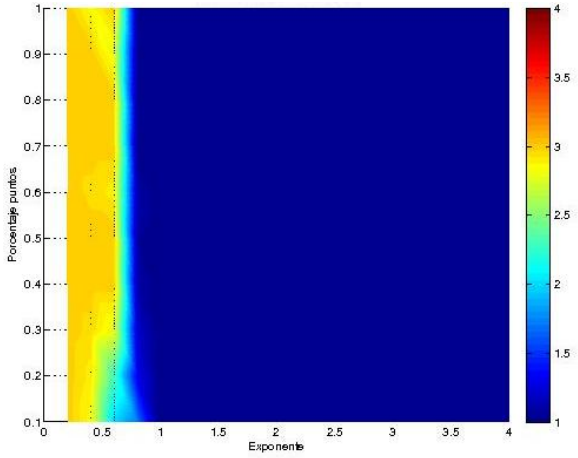


Dimensión 3, ΔJ_{UG}

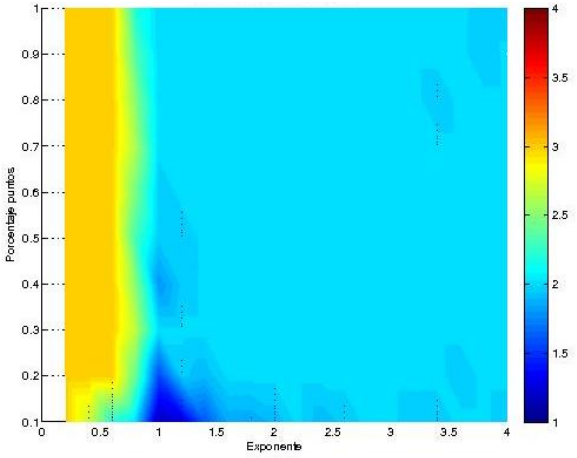
Distancia 1



Distancia 2

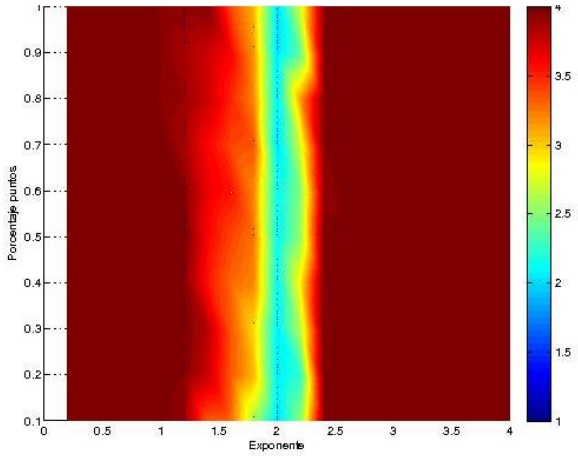


Distancia 3

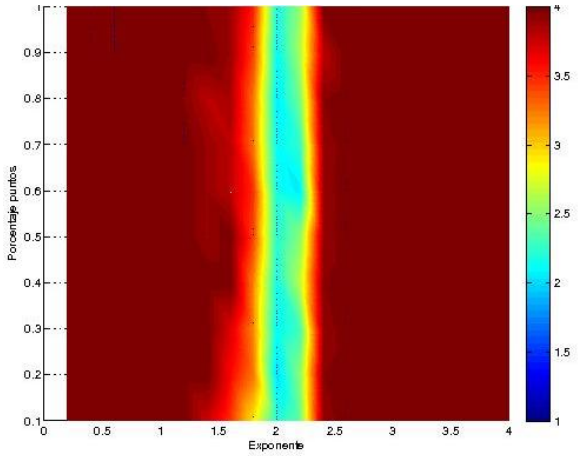


Dimensión 4, AIC

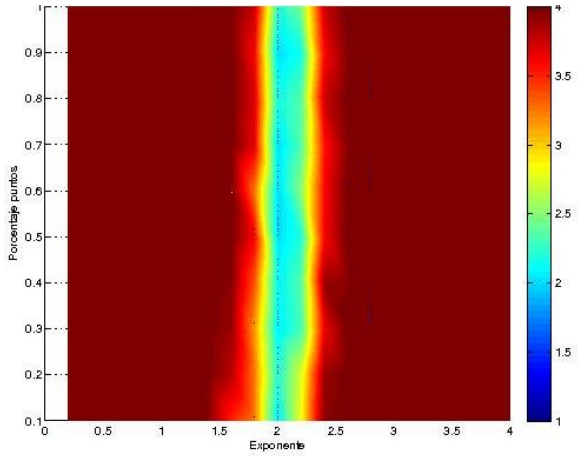
Distancia 1



Distancia 2

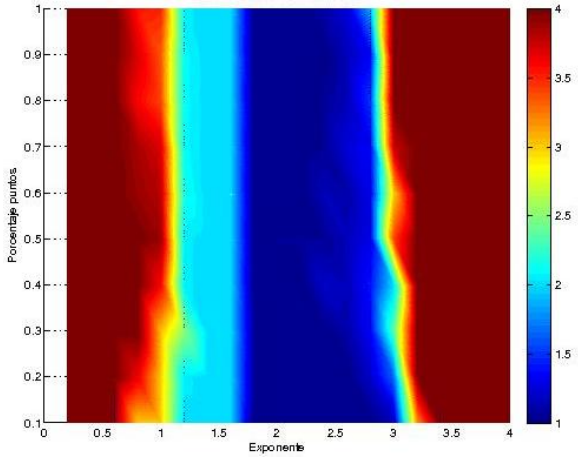


Distancia 3

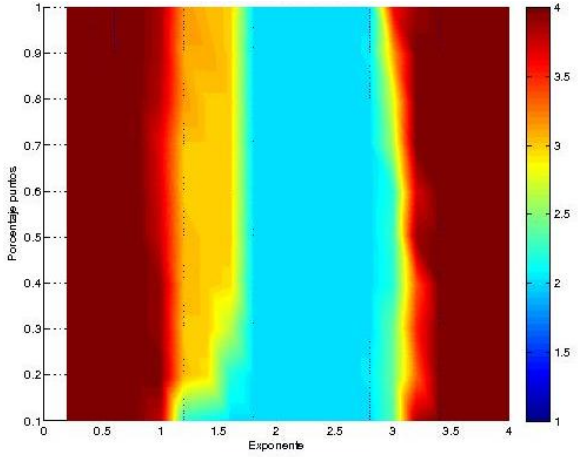


Dimensión 4, BIC

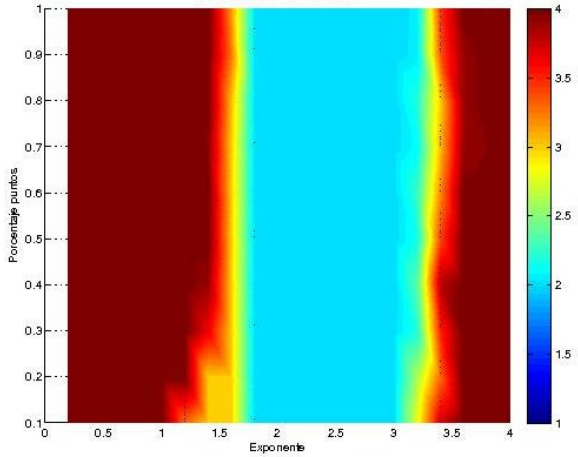
Distancia 1



Distancia 2

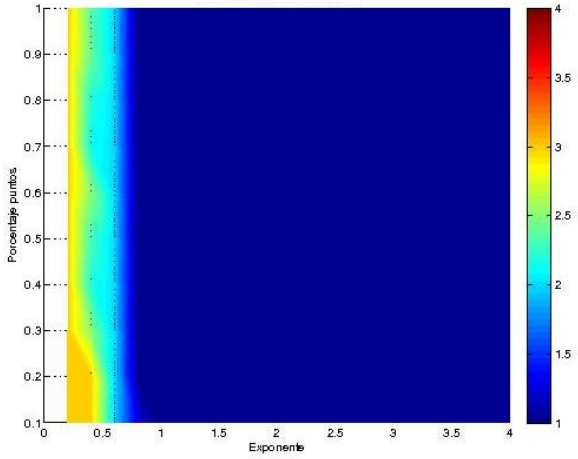


Distancia 3

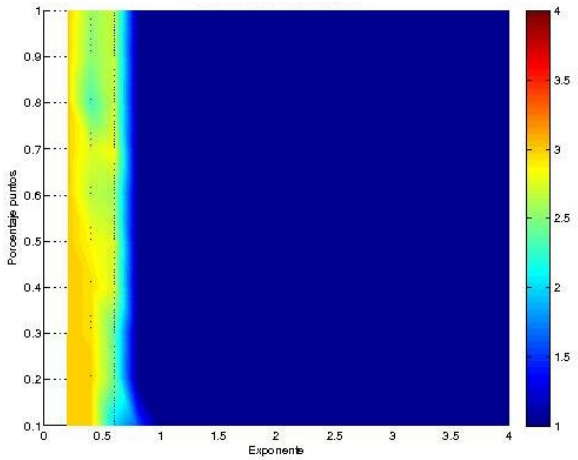


Dimensión 4, ΔJ_{UG}

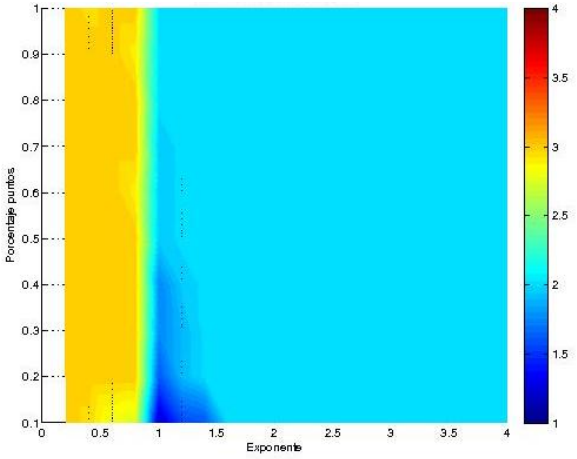
Distancia 1



Distancia 2

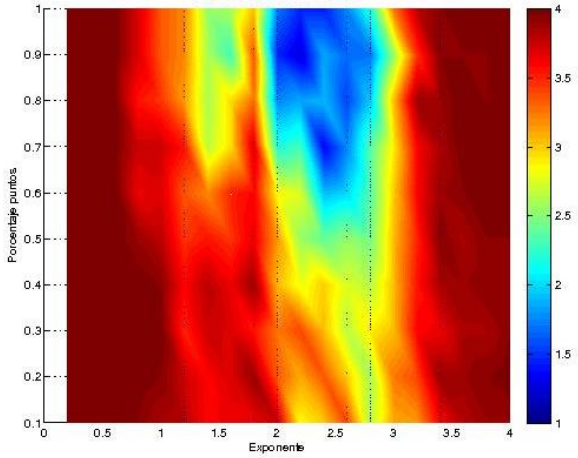


Distancia 3

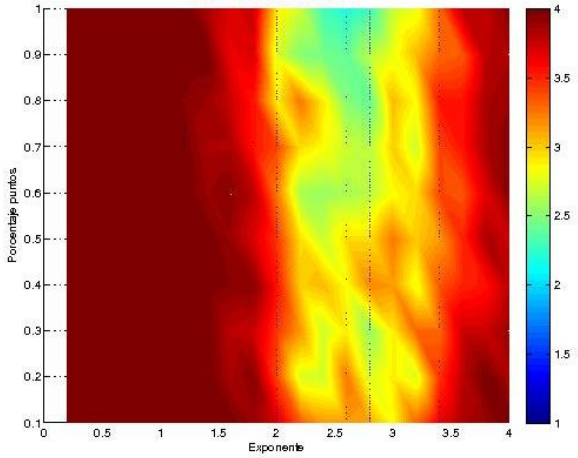


Dimensión 10, AIC

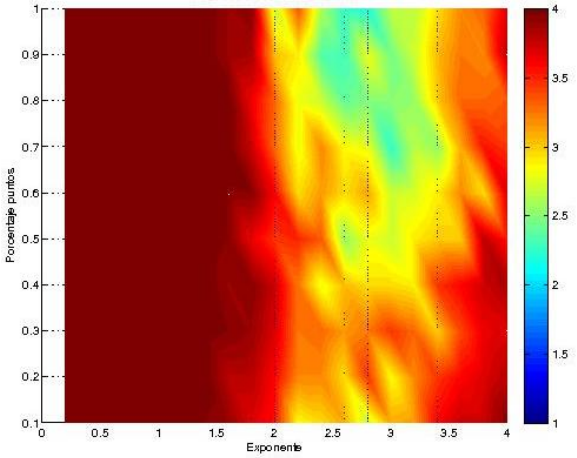
Distancia 1



Distancia 2

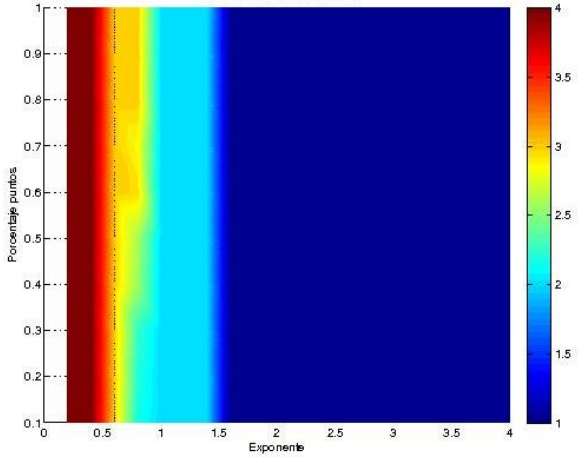


Distancia 3

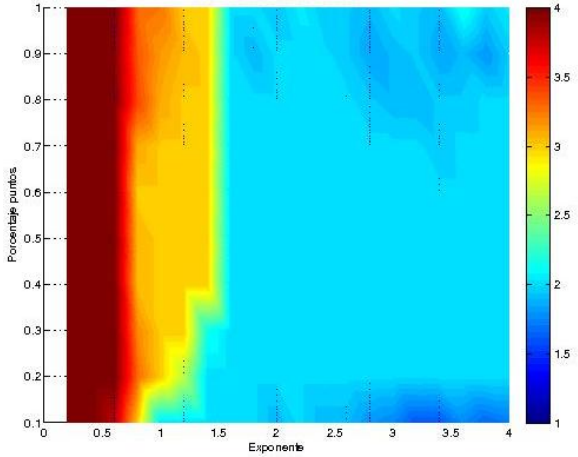


Dimensión 10, BIC

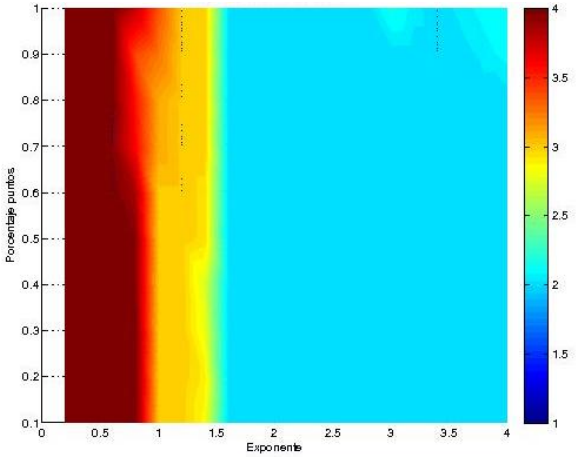
Distancia 1



Distancia 2

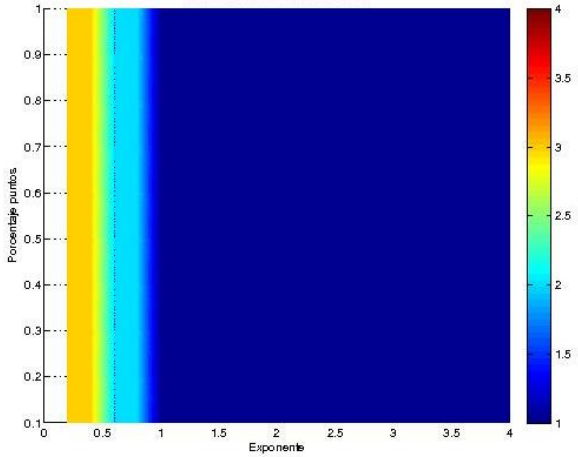


Distancia 3

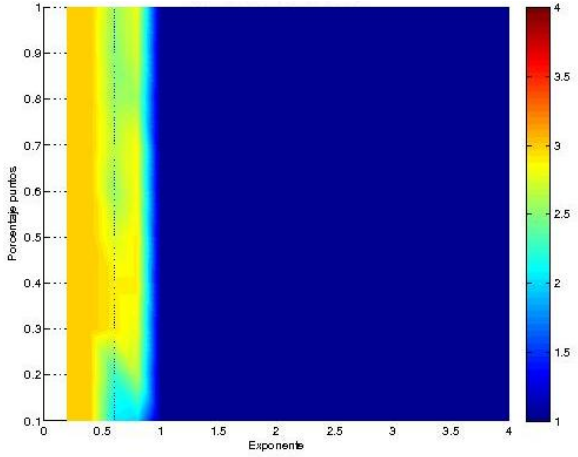


Dimensión 10, ΔJ_{UG}

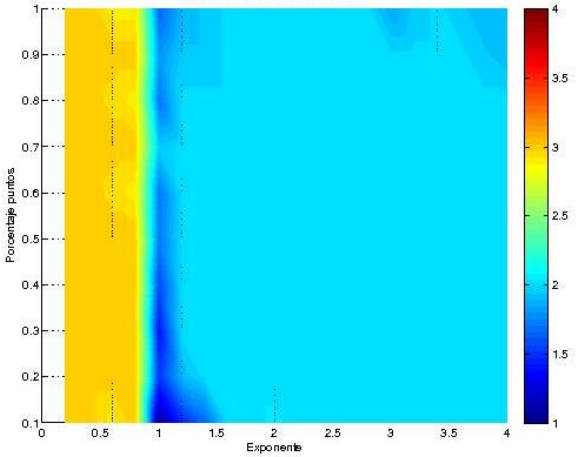
Distancia 1



Distancia 2



Distancia 3



4.2.3.1 Conclusiones

Como se puede apreciar en las figuras y es un resultado que se puede esperar a partir de los resultados que se han obtenido en los análisis previos, AIC sobrestima el número de clusters para prácticamente la totalidad de las situaciones. En general, siempre se comporta mejor en torno al exponente $k = 2$.

También se hace notar que aunque BIC, como ya sabíamos de las pruebas anteriores, permite un mayor solape entre los clusters que ΔJ_{UG}, una vez que se está trabajando dentro del rango de distancias entre clusters donde ΔJ_{UG} acierta, se puede apreciar que esté tiene mayor tolerancia al grado de normalidad.
Además, también se puede observar que ΔJ_{UG}, al igual que en el problema anterior, sobrestima menos que BIC para valores de k pequeños.

Por el mismo motivo que en el problema anterior, en dimensión 10 tanto BIC como AIC mejoran con respecto a dimensión 4.

En general se puede decir que el número de puntos de un cluster con respecto al otro prácticamente no afecta al desempeño de los métodos, solo se nota en el caso de un desbalanceo muy acusado, es decir, cuando uno de los clusters contiene el 20% o menos del número de puntos del otro.

Continuando un poco más en profundidad con el análisis en función del desbalanceo, se puede apreciar en la Figura 12 como en dimensiones pequeñas ΔJ_{UG} tiende a ajustar a un cluster cuando está muy descompensado, cuando el número de puntos de uno de los clusters está sobre el 20% de los puntos del otro. Pasado ese umbral, ambos algoritmos se comportan de manera similar y de manera coherente con los resultados obtenidos hasta el momento.

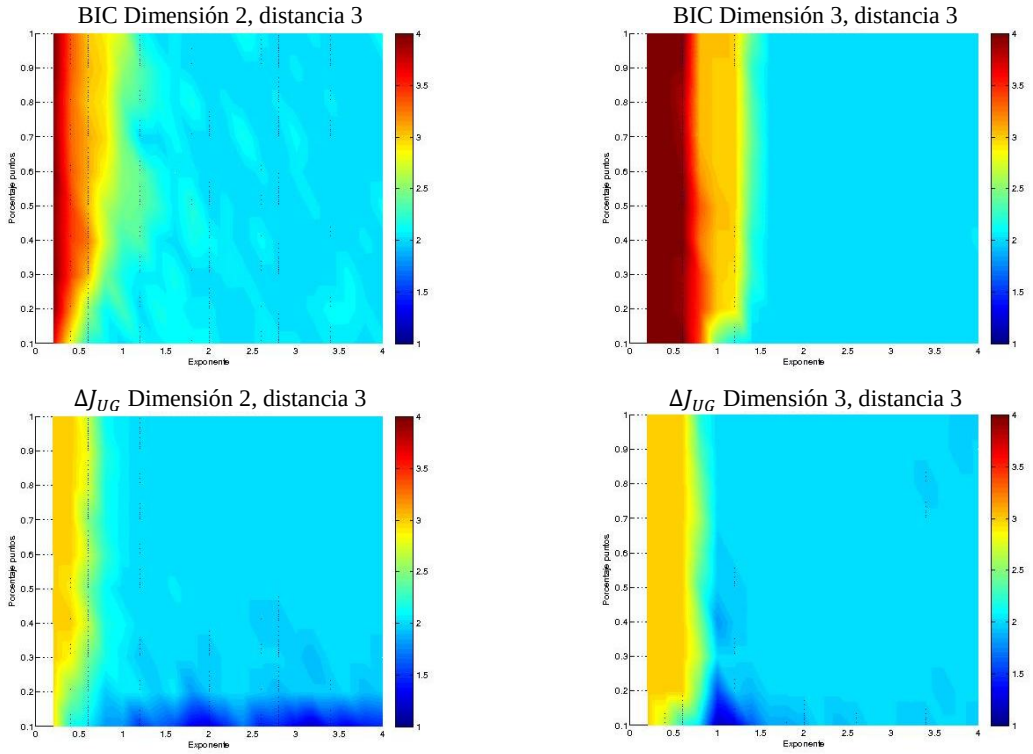


Figura 12. Comparativa BIC y $\boldsymbol{\Delta J_{UG}}$ para distancia 3 y dimensiones pequeñas, 2D y 3D.

Sin embargo, para dimensiones más altas, se puede observar como aun para un desbalanceo muy acusado, ΔJ_{UG} tiene un desempeño mucho mejor que BIC, pues desaparece esa franja entre el 10% y el 20% donde ΔJ_{UG} subestimaba para dimensiones pequeñas. Esto se puede observar en la siguiente figura.

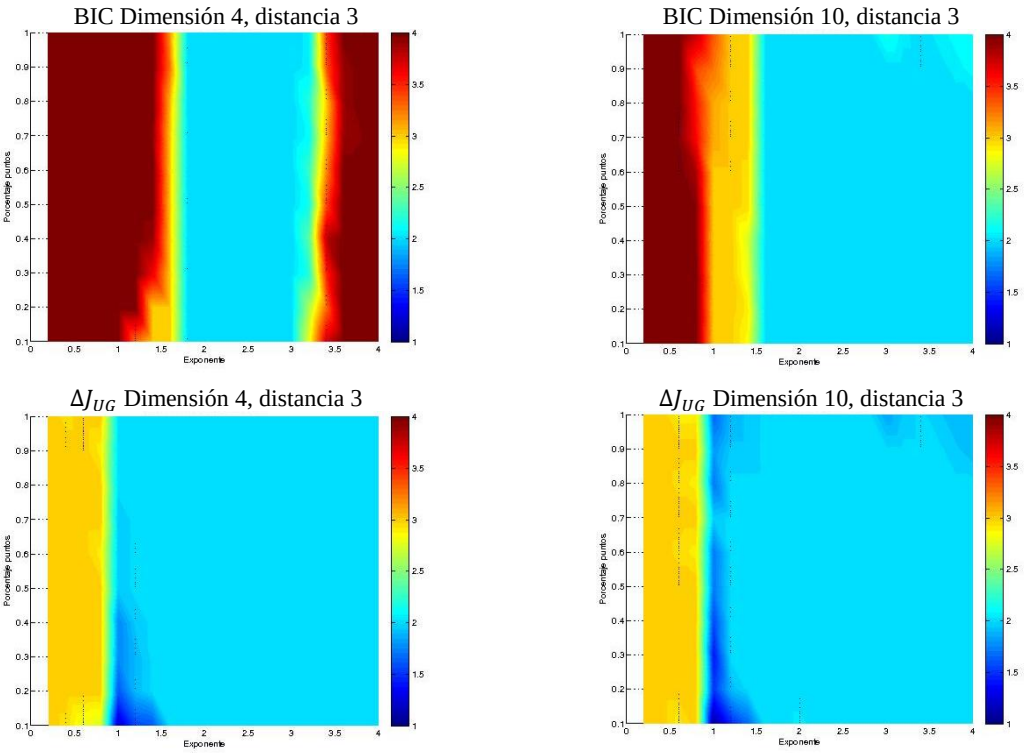


Figura 13. Comparativa BIC y $\boldsymbol{\Delta J_{UG}}$ para distancia 3 y dimensiones grandes, 4D y 10D.

4.2.4 Análisis en función de la dimensión.

4.2.4.1 Introducción

En esta ocasión se propone un análisis más exhaustivo del comportamiento de los diferentes métodos en función de la dimensión cuando los clusters son gaussianos, pues es el caso donde se puede observar que los métodos tienen un mejor desempeño. Para ello se plantean dos problemas diferentes:

- **Twonorm** (2): se basa principalmente en dos clusters generados mediante dos distribuciones normales. En el segundo cluster se va a ir reduciendo el número de puntos de manera similar a como hizo en el análisis anterior, y se van a analizar los problemas para un conjunto más extenso de dimensiones.

- **Threenorm** (2): este problema consta de tres clusters ya desbalanceados de por si en una proporción $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$. El objetivo que se persigue con este problema es analizar cómo afecta la dimensión al desempeño de los métodos cuando se aumenta el número de clusters en los problemas, pasarán de tener 2 clusters a tener 3.

4.2.4.2 Twonorm

4.2.4.2.1 Descripción del problema

Twonorm es un problema sintético diseñado inicialmente para clasificación, pero puesto que las clases de los problemas son conocidas, es un buen problema para analizar algoritmos de validación de clustering.
El problema consta de dos clusters con una dimensión d variable entre 2 y 20 con incrementos de 2, es decir: dimensión $d = \{2, 4, 6, \dots, 20\}$.

Cada cluster es extraído de una distribución normal con $\Sigma = I$ y media situada en $c_1 = (a, a, .., a)$ para el primer cluster, y $c_2 = (-a, -a, .., -a)$ para el segundo cluster, donde $a = 2/\sqrt{d}$.

El problema está diseñado de tal forma la complejidad es siempre constante independientemente de la dimensión, o lo que es lo mismo, el error de Bayes es constante, y con valor aproximadamente de 0.023 en todos los casos.

El número de puntos del primer cluster es de 500, mientras que el del segundo se genera con un tanto por ciento de los puntos del primer cluster. $\%\, de\ puntos = \{100\%, 90\%, \dots, 20\%, 10\%\}$, por lo que el número de puntos del segundo cluster variará desde 500 puntos hasta 50 puntos. Esto provoca un desbalanceo del número de puntos entre el primer y el segundo cluster.

El ajuste mediante EM, se ha hecho para $n_c = \{1,2,3,4\}$. Y en este caso por cada ajuste a un n_c distinto el algoritmo EM se ha ejecutado 20 veces.

Hay que destacar que por cada dupla dimensión-desbalanceo se han generado 100 problemas diferentes, por lo tanto, el proceso descrito en la sección 3 se ejecutará sobre cada uno de estos problemas.

Con los datos obtenidos, se ha generado una tabla por método de validación en la que se representa el número de aciertos por cada dupla dimensión-desbalanceo. Esta tabla se ha representado mediante una gráfica en la que el eje x es el número de puntos del segundo cluster, el eje y es la dimensión y el tercer eje es una escala de grises que representa el número total de aciertos por cada dupla dimensión-desbalanceo.
Además se obtienen unas tablas por cada método donde se representa por cada dupla dimensión-desbalanceo, la media del número de clusters detectados y su desviación estándar.

4.2.4.2.2 Resultados

A continuación se muestran los resultados obtenidos mediante el procedimiento previo.

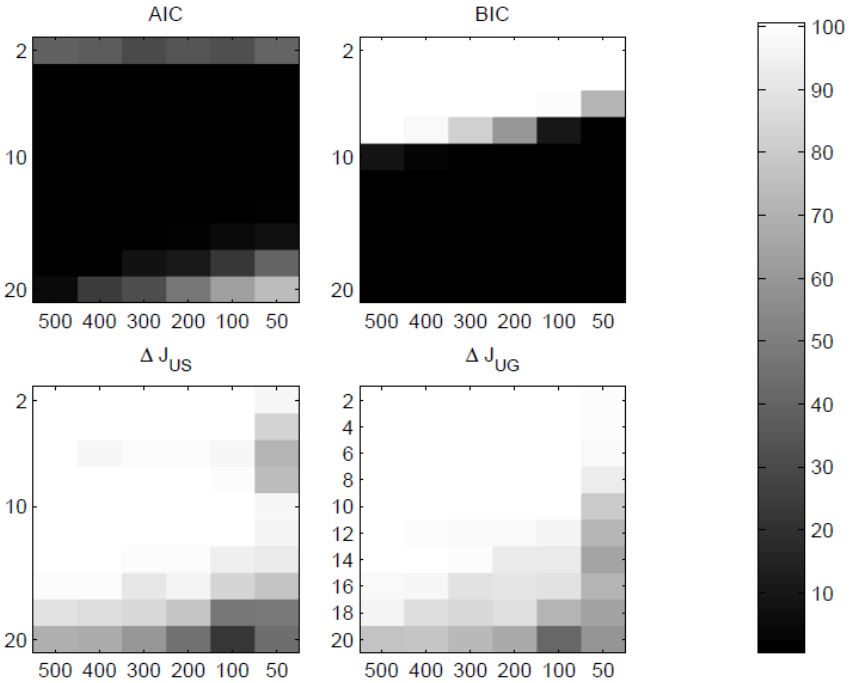


Figura 14. Número total de aciertos por cada dupla dimensión-número de puntos del segundo cluster. Cada una de las cuatro gráficas representa los resultados para cada uno de los métodos de validación

AIC

Dimensión \ Núm. Puntos	500	400	300	200	100	50
2	3.0 ± 0.9	3.0 ± 0.9	3.1 ± 0.8	3.0 ± 0.8	3.1 ± 0.9	3.0 ± 0.9
4	3.7 ± 0.5	3.7 ± 0.5	3.8 ± 0.4	3.7 ± 0.5	3.7 ± 0.5	3.6 ± 0.5
6	3.8 ± 0.4	3.8 ± 0.4	3.8 ± 0.4	3.8 ± 0.4	3.8 ± 0.4	3.8 ± 0.4
8	3.9 ± 0.3	3.8 ± 0.4	3.8 ± 0.4	3.8 ± 0.4	3.9 ± 0.3	3.8 ± 0.4
10	3.8 ± 0.4	3.8 ± 0.4	3.9 ± 0.4	3.8 ± 0.4	3.8 ± 0.4	3.7 ± 0.5
12	3.8 ± 0.4	3.8 ± 0.4	3.8 ± 0.4	3.6 ± 0.5	3.7 ± 0.5	3.7 ± 0.5
14	3.7 ± 0.4	3.7 ± 0.4	3.6 ± 0.5	3.6 ± 0.5	3.6 ± 0.5	3.6 ± 0.5
16	3.6 ± 0.5	3.6 ± 0.5	3.6 ± 0.5	3.4 ± 0.5	3.4 ± 0.6	3.3 ± 0.6
18	3.6 ± 0.5	3.4 ± 0.5	3.2 ± 0.6	3.2 ± 0.6	3.0 ± 0.7	2.8 ± 0.7
20	3.2 ± 0.5	2.9 ± 0.6	2.8 ± 0.6	2.6 ± 0.6	2.5 ± 0.6	2.3 ± 0.5

Tabla 7. AIC. Media y desviación estándar por cada dupla dimensión-número de puntos para el método AIC.

BIC

Dimensión \ Núm. Puntos	500	400	300	200	100	50
2	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
4	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
6	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.1	1.7 ± 0.5
8	2.0 ± 0.0	2.0 ± 0.1	1.8 ± 0.4	1.6 ± 0.5	1.1 ± 0.3	1.0 ± 0.0
10	1.1 ± 0.3	1.0 ± 0.2	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
12	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
14	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
16	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
18	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
20	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0

Tabla 8. BIC. Media y desviación estándar por cada dupla dimensión-número de puntos para el método BIC.

ΔJ_{US}

Dimensión \ Núm. Puntos	500	400	300	200	100	50
2	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.2
4	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	1.8 ± 0.4
6	2.0 ± 0.0	2.0 ± 0.2	2.0 ± 0.1	2.0 ± 0.1	2.0 ± 0.2	1.7 ± 0.5
8	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.1	1.7 ± 0.4
10	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.2
12	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.2
14	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.1	2.0 ± 0.1	2.1 ± 0.2	2.1 ± 0.3
16	2.0 ± 0.1	2.0 ± 0.1	2.1 ± 0.3	2.0 ± 0.2	2.2 ± 0.4	2.2 ± 0.4
18	2.1 ± 0.3	2.1 ± 0.3	2.1 ± 0.4	2.2 ± 0.4	2.5 ± 0.5	2.5 ± 0.5
20	2.3 ± 0.5	2.3 ± 0.5	2.4 ± 0.5	2.5 ± 0.5	2.8 ± 0.4	2.6 ± 0.5

Tabla 9. ΔJ_{US}. Media y desviación estándar por cada dupla dimensión-número de puntos para el método ΔJ_{US}.

ΔJ_{UG}

Dimensión \ Núm. Puntos	500	400	300	200	100	50
2	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.1
4	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.1
6	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.1
8	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	1.9 ± 0.3
10	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	1.8 ± 0.4
12	2.0 ± 0.0	2.0 ± 0.1	2.0 ± 0.1	2.0 ± 0.1	2.0 ± 0.2	1.7 ± 0.5
14	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.1	2.0 ± 0.3	1.9 ± 0.3	1.7 ± 0.5
16	2.0 ± 0.1	2.0 ± 0.2	2.0 ± 0.3	2.0 ± 0.3	1.9 ± 0.3	1.9 ± 0.5
18	2.0 ± 0.2	2.0 ± 0.4	2.0 ± 0.4	2.0 ± 0.3	2.1 ± 0.5	2.2 ± 0.6
20	2.2 ± 0.4	2.2 ± 0.4	2.2 ± 0.5	2.3 ± 0.5	2.5 ± 0.5	2.4 ± 0.5

Tabla 10. ΔJ_{UG}. Media y desviación estándar por cada dupla dimensión-número de puntos para el método ΔJ_{UG}.

4.2.4.2.3 Conclusiones

Como se puede apreciar tanto en la Figura 14 como en la Tabla 7, el porcentaje de aciertos de AIC es prácticamente nulo, tendiendo a sobrestimar el número de clusters en todas las pruebas.

Si comprobamos ahora el comportamiento de BIC, como era de esperar pues se vio en los análisis previos, tiene un desempeño muy bueno para dimensiones pequeñas, sin que se vea afectado prácticamente por el desbalanceo de los clusters hasta la dimensión 6, a la cual, cuando el desbalanceo es muy acusado comienza a sufrirlo y a provocar que falle.

Cuando ya sobrepasamos la dimensión 6, el desempeño del método desciende drásticamente y comienza a fallar el 100% de los problemas. Esto se debe al mismo motivo que se explicó previamente, cuanto mayor es la dimensión, mayor es el número de parámetros a estimar, lo que produce que si no se varía el número de puntos tienda a subestimar el número de clusters.

Los dos enfoques de ΔJ por su parte, tienen un comportamiento muy similar, consiguiendo un desempeño realmente positivo hasta la dimensión 16. A partir de dimensión 16 su tasa de aciertos baja, aunque continúa por encima del 50% hasta dimensión 20.

4.2.4.3 Threenorm

4.2.4.3.1 Descripción del problema

En este problema se añade un tercer cluster y el número de puntos de los problemas permanece constante, estando conformados por un total de 1000 puntos. La proporción de puntos de los clusters es de ¼, ¼, ½, es decir, contienen aproximadamente 250 puntos el primer y segundo cluster, y aproximadamente 500 puntos el tercer cluster.
Los clusters se encuentran centrados en $c_1 = (a, a, \dots, a)$, $c_2 = (-a, -a, \dots, -a), y\ c_3 = (a, -a, a, -a, \dots, -a)$. El valor de $a = \sqrt{\frac{8}{d}}$, y proporciona una complejidad a los problemas constante a lo largo de la dimensión, es decir, el error de Bayes es constante y aproximadamente 0.023, al igual que en el problema Twonorm.

La dimensión d varía y toma valores entre 2 y 20 en intervalos de 2. Esto es, $d = \{2, 4, 6, \dots, 20\}$.

El ajuste mediante EM, se ha hecho para $n_c = \{1, 2, 3, 4\}$. Y por cada ajuste a un número de componentes determinado, n_c, el algoritmo EM se ha ejecutado 20 veces.

Hay que destacar que por cada dimensión se han generado 100 problemas diferentes, por lo que el proceso descrito en la sección 3 se ejecutará sobre cada uno de estos problemas.

Con los datos obtenidos se ha generado una gráfica en escala de grises similar a las obtenidas en el problema Twonorm, en la que cada cuadrícula representa el número de aciertos totales que realiza cada método por cada dimensión. Además se ha generado una tabla donde se representa por cada método y por cada dimensión, el número medio de clusters detectados y su desviación estándar

4.2.4.3.2 Resultados

Los resultados obtenidos son los siguientes.

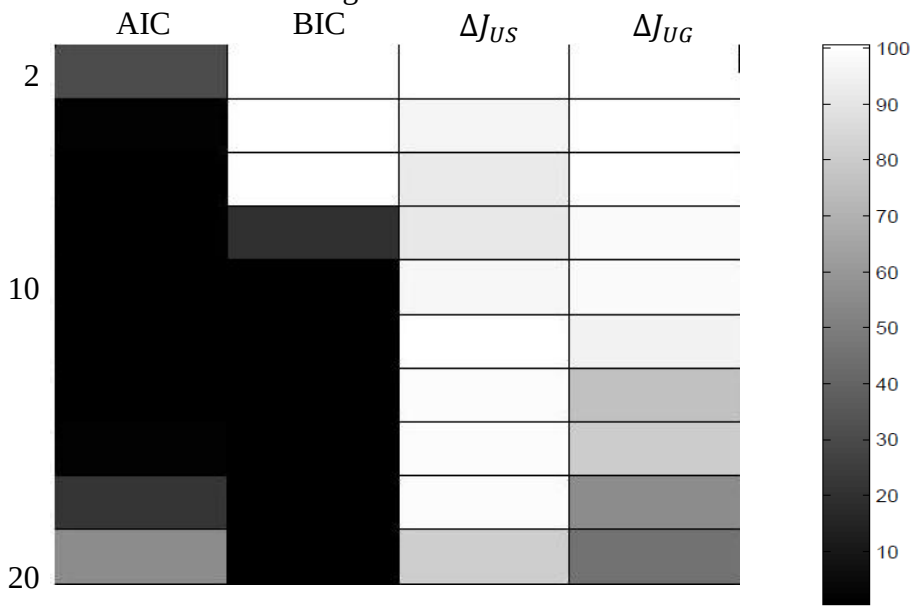


Figura 15. Número total de aciertos por cada dimensión y método de validación.

Dimensión \ Método	AIC	BIC	ΔJ_{US}	ΔJ_{UG}
2	4.1 ± 0.8	3.0 ± 0.0	3.0 ± 0.0	3.0 ± 0.0
4	4.8 ± 0.5	3.0 ± 0.0	3.0 ± 0.2	3.0 ± 0.0
6	4.8 ± 0.4	3.0 ± 0.0	2.9 ± 0.3	3.0 ± 0.0
8	4.8 ± 0.4	2.2 ± 0.4	2.9 ± 0.3	3.0 ± 0.1
10	4.8 ± 0.4	2.0 ± 0.0	3.0 ± 0.2	3.0 ± 0.1
12	4.8 ± 0.4	1.8 ± 0.4	3.0 ± 0.0	3.0 ± 0.2
14	4.7 ± 0.5	1.0 ± 0.0	3.0 ± 0.1	2.8 ± 0.4
16	4.5 ± 0.5	1.0 ± 0.0	3.0 ± 0.1	2.8 ± 0.4
18	4.1 ± 0.7	1.0 ± 0.0	3.0 ± 0.1	2.6 ± 0.5
20	3.6 ± 0.7	1.0 ± 0.0	3.2 ± 0.4	2.5 ± 0.5

Tabla 11. Media y desviación estándar sobre los ajustes elegidos de 100 problemas por cada dimensión y método de validación.

4.2.4.3.3 Conclusiones

El resultado para AIC, como en todos los demás análisis, es que sobrestima el número de clusters para cualquier dimensión.

BIC es un algoritmo cuyo comportamiento se ve afectado por el ratio puntos/dimensión en mayor medida que los dos métodos ΔJ que se están analizando. Así, en este problema era de esperar que funcionase correctamente para dimensiones pequeñas y según aumenta la dimensión, comenzase a subestimar el número de clusters.
Como se puede ver en la Figura 15 y en la Tabla 11, tiene un desempeño óptimo cuando la dimensión está comprendida entre 2 y 6, y comienza a fallar para dimensiones mayores que 6, tendiendo según aumenta la dimensión a escoger ajustes con un solo cluster. Es decir, los resultados que se obtienen para BIC se corresponden con lo esperado.

Por el contrario ambos métodos ΔJ no se ve tan afectado y consigue un mejor desempeño a lo largo de todas las dimensiones.
Hay que destacar, que en este caso se puede observar como ΔJ_{UG} consigue un desempeño mejor que ΔJ_{US} para dimensiones bajas, pero sin embargo, para dimensiones altas ΔJ_{US} obtiene número de aciertos mayor que ΔJ_{UG}.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

El objetivo principal de este proyecto era hacer un análisis del desempeño de métodos de validación basados en la negentropía, Negentropy-based Validation (ΔJ), en problemas sintéticos, así como su comparación con otros métodos más tradicionales, AIC y BIC. Para ello se han planteado varios problemas que se han resuelto de manera sistemática. En cada nuevo problema a estudiar, o bien se ha introducido una nueva variable o se han combinado diferentes variables para observar el comportamiento del método en función de los valores que toman. Las variables que se han utilizado han sido: el solape entre clusters (apartados 4.2.1 y 4.2.2), el grado de normalidad (apartados 4.2.2 y 4.2.3), el grado de desbalanceo entre clusters (4.2.3 y 4.2.4 Twonorm), la dimensión (apartados 4.2.4 Twonorm y Threenorm) y por último, el número de clusters que contenía el problema (apartado 4.2.4 Threenorm).

Para todos los análisis exceptuando Threenorm, se han utilizado problemas d-dimensionales con dos clusters en los que se hace variar el solape entre ellos, el grado de normalidad o el desbalanceo del número de puntos de un cluster con respecto al otro. Todos estos problemas a excepción de uno se han realizado para diferentes dimensiones, lo que nos ha permitido tener una idea general de la tendencia de los diferentes métodos con la dimensión además de con las variables implicadas. Finalmente con el objetivo de observar el comportamiento de los diferentes métodos de validación en función de la dimensión de los datos, se han realizado las pruebas con los problemas Twonorm y Threenorm, en los cuales la dimensión varía desde 2 hasta 20 en pasos de 2. En Twonorm se hace un análisis más exhaustivo del desempeño de los métodos en función de la dimensión de los datos y el desbalanceo del segundo cluster. En Threenorm se hace el análisis sobre las mismas dimensiones que en Twonorm pero con un cluster más y con problemas en los que los clusters están desbalanceados pero siempre en la misma proporción: ½, ¼, ¼.

A lo largo de todos los análisis se ha podido observar como AIC ha obtenido unos resultados muy negativos proporcionando siempre una fuerte sobrestimación en el número de clusters. En ninguno de los análisis ha obtenido unos resultados mínimamente satisfactorios que poder resaltar.

BIC sin embargo tiene un buen desempeño en problemas donde el ratio número de puntos frente a dimensión es alto y además la función que genera la distribución de puntos es gaussiana. En estos casos presenta mayor tolerancia al solape que ΔJ, lo cual es de esperar pues el funcionamiento óptimo de BIC se da para clusters generados con distribuciones normales.

ΔJ, el método objeto de este proyecto, como se ha podido comprobar a lo largo del mismo aporta una mayor tolerancia al tipo de distribución que genera los clusters cuando no están muy solapados y a variaciones en el ratio número de puntos frente a dimensión.

Todo parece indicar que para problemas reales, donde suele ser común que el ratio número de puntos frente a la dimensión sea bajo y la distribución que genera los datos no tiene

necesariamente que ser gaussiana, la elección de ΔJ como algoritmo de validación puede ser una opción mucho más acertada que BIC y por supuesto que AIC.

5.2 Trabajo futuro

En este proyecto se ha tratado de cubrir una amplia variedad de situaciones mediante el análisis de problemas sintéticos. Comenzar los análisis con estos problemas nos ha permitido cubrir diferentes casos y hacer un análisis exhaustivo del desempeño de los métodos en todos ellos cuando el entorno está muy controlado, es decir, se conoce el proceso que generó el problema.
Pero el objetivo final es dar una aplicación real al método de validación ΔJ, donde pueda ser de utilidad bien porque mejore el rendimiento de la aplicación con respecto a la utilización de otros métodos, o bien porque sea el único método capaz de aportar un rendimiento válido para dicha aplicación. En estas situaciones donde no se tiene un conocimiento tan amplio sobre el proceso que generó los datos, el análisis se complica, aunque tener como base este proyecto puede facilitar el trabajo.

Como se mencionó en la introducción, uno de los usos de los métodos de clustering es la clasificación automática de imágenes, y ésta sería una posible aplicación del método de validación de clustering analizado.

Al finalizar el análisis con datos sintéticos, se ha hecho una prueba preliminar con una base de datos real. Esta base de datos contiene un conjunto de puntos SIFT (24) obtenidos a partir de un conjunto de imágenes. Estas imágenes forman parte de una base de datos muy conocida, Feret (25), la cual contiene multitud de imágenes de caras de personas con rasgos muy variadas, algunas tienen gafas, otras barba, otras el pelo largo, unas son varones y otras mujeres, etc.

La prueba se ha basado en un análisis de este conjunto de puntos SIFT, obteniéndose resultados prometedores.
El proceso que se ha seguido para la obtención de los resultados ha sido el siguiente:
Sobre el conjunto de puntos SIFT, se ha aplicado el algoritmo PCA para reducir la dimensión, pues cada dato consta de 128 componentes o lo que es lo mismo, $d = 128$. Sobre el conjunto de datos obtenidos con la nueva dimensión tras aplicar PCA, $l = 10$, se ha realizado varios ajustes a mezcla de gaussianas mediante el método EM con n_c desde 1 hasta 15. Por cada ajuste a un n_c diferente se ha ejecutado 10 veces el método EM.

Como resultado de aplicar el método ΔJ_{UG} a los modelos devueltos por el método EM, se ha escogido una partición que contiene dos clusters.

Estos datos se han representado de la siguiente manera:

1. Sobre cada imagen se ha dibujado en un color diferente los puntos pertenecientes a clusters diferentes.
2. Al lado de cada imagen, se ha representado un histograma donde cada barra representa el porcentaje de puntos del cluster que está representando frente al total de puntos de la imagen.

Como se puede apreciar en la Figura 16, cuando la imagen es de una mujer de color, ésta contiene significativamente más puntos de un cluster que del otro.

Estos resultados nos hacen pensar que continuar analizando este método más en profundidad pueda llevar a encontrar una aplicación real en esta u otras áreas.

Puesto que mediante los análisis realizados a lo largo de este proyecto ha quedado evidenciado que ΔJ_{UG} tiene un mejor desempeño que BIC trabajando con datos sintéticos, y que en la prueba preliminar del método ΔJ_{UG} con datos reales se han obtenido unos resultados que permiten pensar que este método de validación tiene grandes posibilidades de ser usado en aplicaciones reales, la propuesta para continuar este proyecto es la siguiente:
Dar el salto a analizar el método de validación con datos reales, otorgándole una aplicación real en la que pueda ayudar a mejorar algún proceso ya existente o implementar una nueva aplicación para la resolución de problemas más complejos.

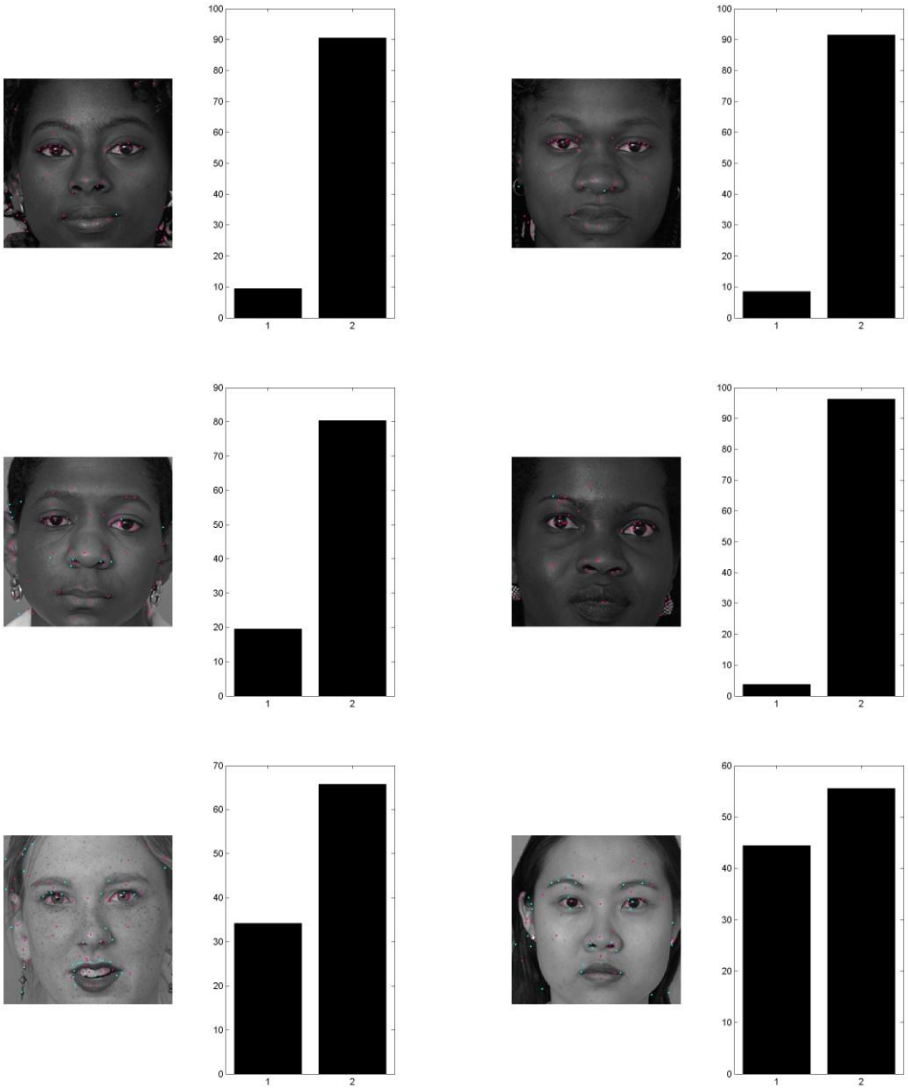


Figura 16. Ejemplos de imágenes de Feret con sus histogramas asociados, donde cada barra representa el porcentaje de puntos del cluster que está representando frente al total de puntos de la imagen.

6 Bibliografía

1. *Cluster validation in problems with increasing dimensionality and unbalanced clusters.* **Luis F. Lago-Fernández, Jesús Aragón, Gonzalo Martínez-Muñoz, Ana González Marcos, Manuel A. Sánchez-Montañés.** 2014, Neurocomputing, Vol. 123, págs. 33-39.

2. **Breiman, Leo.** *Bias, variance, and arcing classifiers.* Statistics Deparment, University of California, Berkeley, CA, USA : Technical Report 460, 1996.

3. *An optimal graph theoric Approach to Data Clustering: Theory and Its Application to Image Segmentation.* **Zhenyu Wu, Richard Leahy.** 11, 1993, IEEE Transactions on Pattern Analysis and Machine, Vol. 15, págs. 1101-1113.

4. *On image classification: City images vs. landscapes.* **Aditya Vailaya, Anil Jain, Hong Jiang Zhang.** 12, 1998, Pattern Recognition, Vol. 31, págs. 1921-1935.

5. **Rebecca Nugent, Marina Meila.** *An Overview of Clustering Applied to Molecular Biology.* 2010. págs. 369-404. Vol. 620.

6. *Clustering of lifestyle risk factors in a general adult population.* **A.Jantine Schuit, A.Jeanne M. van Loon, Marja Tijhuis., Marga C. Ocké.** 3, 2002, Prev Med, Vol. 35, págs. 219-224.

7. *Cluster Analysis in Marketing Research: Review and Suggestions for Application.* **Stewart, Girish Punj and David W.** 2, 1983, Journal of Marketing Research, Vol. 20, págs. 134-148.

8. *Automatic clustering of business processes in business systems planning.* **Lee, Hsuan-Shih.** 2, 1999, European Journal of Operational Research, Vol. 144, págs. 354-362.

9. *Model-based clustering for social networks.* **Mark S. Handcock, Adrian E. Raftery, Jeremy M. Tantrum.** 2, 2007, Journal of the Royal Statistical Society, Vol. 170, págs. 301-354.

10. *Maximum Likelihood from Incomplete Data via the EM Algorithm.* **Dempster, A.P., Laird, N.M., Rubin, D.B.** 1977, J Royal Statistical Soc. B 39, págs. 1-38.

11. *Some Methods for classification and Analysis of Multivariate Observations.* **MacQueen, J.** 1967, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press, págs. 281–297.

12. **Trevor Hastie, Robert Tibshirani, Jerome Friedman.** 14.3.12 Hierarchical clustering. *The Elements of Statistical Learning.* s.l. : Springer-Verlag, 2009, págs. 520-528.

13. *A new look at the statistical model identification.* **Akaike, Hirotugu.** 1974, IEEE Trans. Automatic, Vol. 19, págs. 716-723.

14. *Estimating the Dimension of a Model.* **Schwartz, G.** 6, 1978, Annals of Statistics, págs. 461–464.

15. **Luis F. Lago-Fernández, Fernando Corbacho.** Using the Negentropy Increment to Determine The Number of Clusters. *Bio-Inspired Systems: Computational and Ambient Intelligence.* 2009, págs. 448-455.

16. **Luis F. Lago-Fernandez, Manuel Sanchez-Montañes, Fernando Corbacho.** Fuzzy Cluster Validation Using the Partition Negentropy Criterion. *Artificial Neural Networks – ICANN 2009.* 2009, Vol. 5769, págs. 235-244.

17. *Normality-based validation for crisp clustering.* **Luis F. Lago-Fernando, Fernando Corbacho.** 2010, Pattern Recognition, Vol. 43, págs. 782-795.

18. *The effect of low number of points in clustering validation via the negentrophy increment.* **Luis F. Lago-Fernandez, Manuel Sánchez-Montañes, Fernando Corbacho.** 16, 2011, Neurocomputing, Vol. 74, págs. 2657-2664.

19. *Estimation of the entropy of a multivariate normal distribution.* **Neeraj Misra, Harshinder Singh, Eugene Demchuk.** 2, 2005, Journal of Multivariate Analysis, Vol. 92, págs. 324-342.
20. **Milton Abramowitz, Irene A. Stegun.** *Handbook of mathematical functions.* New York : Dover, 1965.
21. **I.T., Jolliffe.** *Principal Component Analysis, Second Edition.* s.l. : Springer, 2002.
22. **Bellman, Richard E.** *Dynamic Programming.* s.l. : Princeton University Press, 1957.
23. **Bishop, Christopher.** *Pattern Recognition and Machine Learning.* s.l. : Springer, 2006.
24. *Distinctive Image Features from Scale-Invariant Keypoints.* **Lowe, David G.** 2, s.l. : International Journal of Computer vision, 2004, Vol. 60, págs. 91-110.
25. *The FERET database and evaluation procedure for face-recognition algorithms.* **P.Jonathon Phillips, Harry Wechsler, Jeffery Huang, Patrick J. Rauss.** 5, s.l. : Image and Vision Computing, 1998, Vol. 16, págs. 295-306.
26. Wikipedia. [En línea] [Citado el: 02 de 02 de 2015.] http://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales.

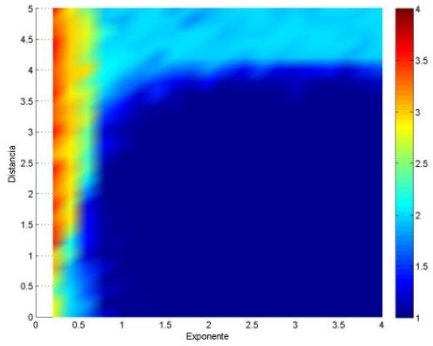
Glosario

AIC	Akaike's Information Criterion
BIC	Bayesian Inference Criterion
EM	Expectation-Maximization
PCA	Principal Component Analysis
fda	función de densidad acumulada
fdp	función de densidad de probabilidad
$\boldsymbol{\Delta J}$	Negentropy-based validation

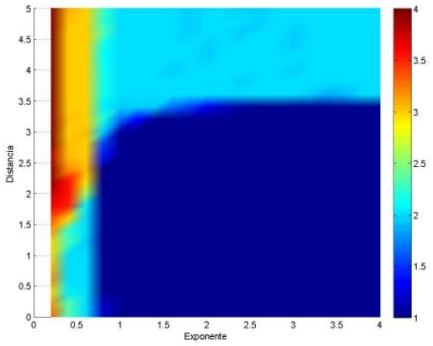
Anexos

A Figuras obtenidas en el punto 4.2.2 para ΔJ_{US}

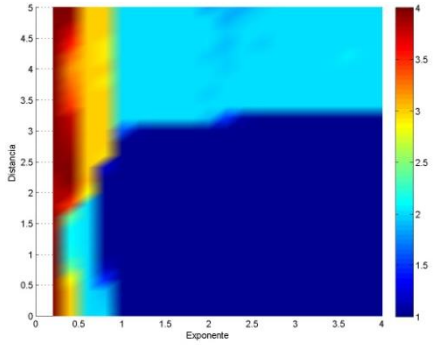
Dimensión 2



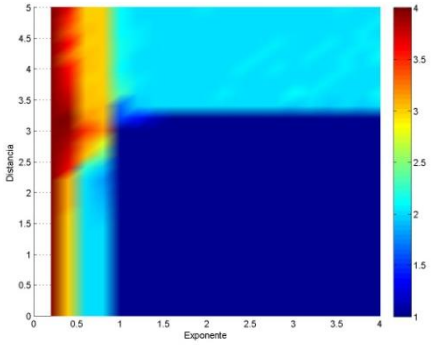
Dimensión 3



Dimensión 4



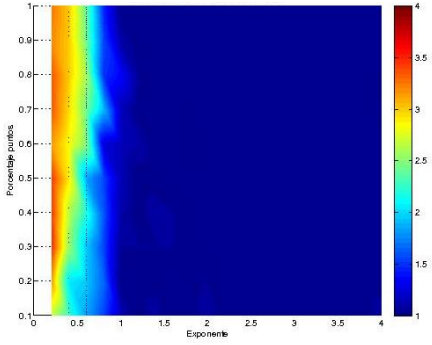
Dimensión 10



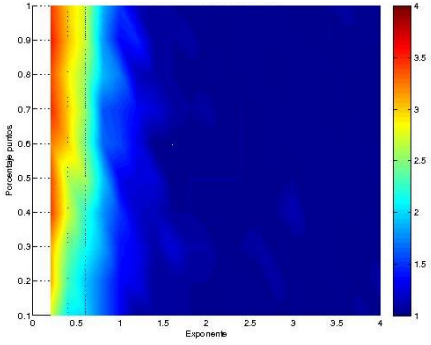
B Figuras obtenidas en el punto 4.2.3 para ΔJ_{US}

Dimensión 2

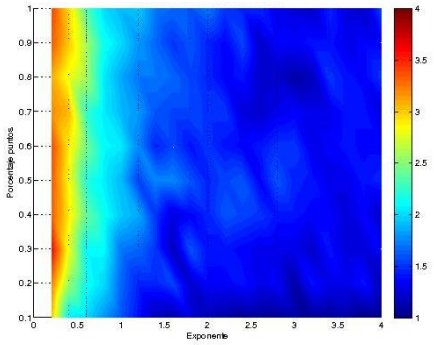
Distancia 1



Distancia 2

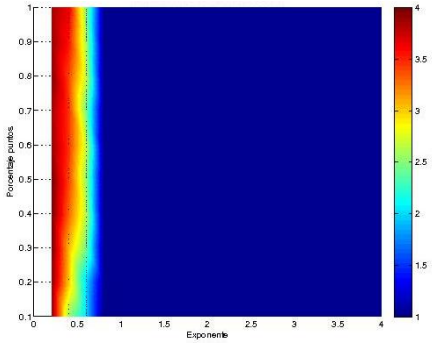


Distancia 3

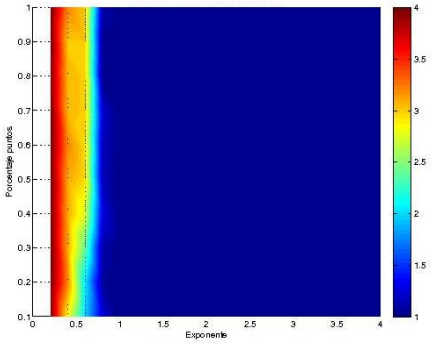


Dimensión 3

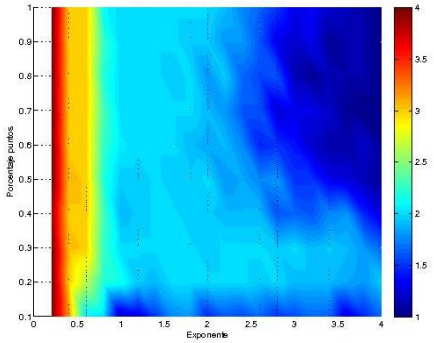
Distancia 1



Distancia 2

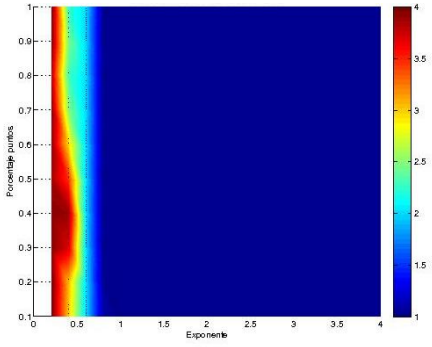


Distancia 3

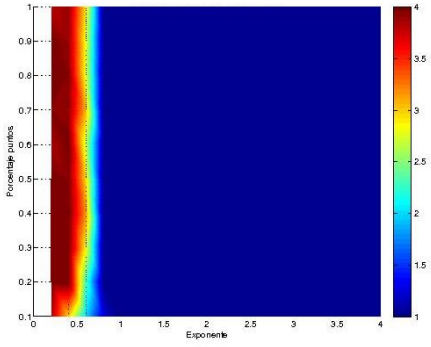


Dimensión 4

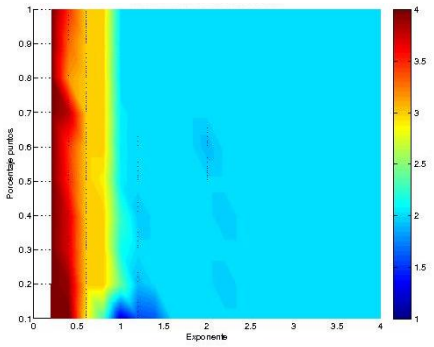
Distancia 1



Distancia 2

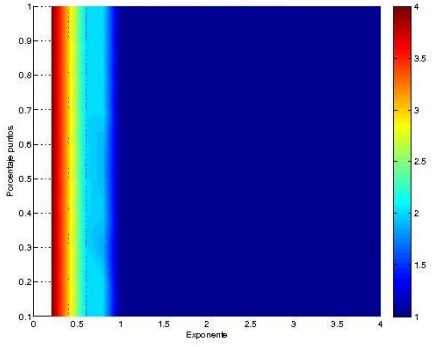


Distancia 3

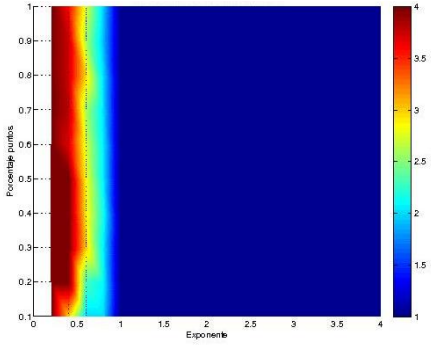


Dimensión 10

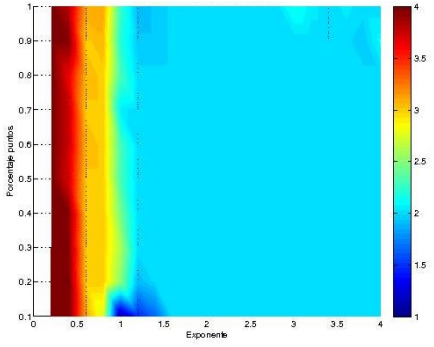
Distancia 1



Distancia 2



Distancia 3



C Manual del programador

Aquí se expone el ejemplo con el ejemplo de código utilizado para la sección 4.2.2

```
%%%%%%%%%%%%%%%%%%%%
% generarProblemas %
%%%%%%%%%%%%%%%%%%%%
close all
clear all
clc

matlabpool close force
matlabpool

dimension = [2 3 4 10];
exponente = .2:.2:4;
distancia = 0:.2:5;
num_puntos = [100 1000 10000 10000]


for dim = 1:length(dimension)
    for a = 1:length(exponente)
        for dist = 1:length(distancia)
            parfor problema = 1:20
                X{problema} =
generaClusterFinal(num_puntos(dim),exponente(a),dimension(dim),[1 1], false);
                X2{problema} =
generaClusterFinal(num_puntos(dim),exponente(a),dimension(dim),[1,1],false);
                X2{problema}(:,1) = X2{problema}(:,1) + distancia(dist);
                X{problema} = [X{problema}; X2{problema}];

                y{problema} = [-1*ones(num_puntos(dim),1)
ones(num_puntos(dim),1)];
            end
            save(sprintf('../data/clusters-%d-%dD-a%.1f-dist%.1f.mat',
num_puntos(dim), dimension(dim), exponente(a), distancia(dist)), 'X', 'y');
        end
    end
```

```
%%%%%%%%%%%%%%%%%%%%%%
% generaClusterFinal %
%%%%%%%%%%%%%%%%%%%%%%

function data = generaClusterFinal(npuntos,a,dim,avrange,verb)

%
% Genera un cluster al azar con n puntos que se distribuyen de manera
% uniforme en la orientacion y segun una distribucion xxx en el radio. La
% coordenada radial depende del exponente a (ver mis notas), y se genera
% llamando a la funcion generaRadio, a la que hay que pasarle los
% parametros dim y a.
%
% npuntos = numero de puntos en el cluster
% a = exponente, tiene que ser un numero real mayor que 0
% dim = dimension
% avrange = rango para los autovalores
```

```
% verb = flag true/false para mostrar info
%
% Nota: si el exponente a es 2, la distribucion generada es gausiana
%

% 1. Genero los puntos segun distribucion normal, esto lo voy a usar solo
% para la orientacion, que debe ser uniforme:
x = randn(npuntos,dim);
norma = sqrt(sum(x.*x,2));
x = x ./ repmat(norma,1,dim);

% 2. Genero la coordenada radial:
r = generaRadio(dim,a,npuntos,1);

% 3. Multiplico cada punto por su radio:
x = x.*repmat(r,1,dim);

% 4. Centrado:
x = x - repmat(mean(x),npuntos,1);

% 5. Whitening:
c = cov(x);
[v d] = eig(c);
lambda = diag(1./sqrt(diag(d)));
x = x*v*lambda;

% 6. Reescalo segun matriz de autovalores aleatoria:
l = diag(avrange(1) + rand(dim,1)*(avrange(2) - avrange(1)));
x = x*sqrt(l);

% 7. Roto con matriz aleatoria:
r = generaRot(dim);
x = x*r;
data = x;

if verb == true

    % Plot si dim = 2:
    if dim == 2
        figure(1); %clf;
        plot(x(:,1),x(:,2),'.');
        axis([-6 6 -6 6]);
        grid on;
    end

    fprintf('-------------------------------------------\n');
    fprintf('Autovalores deseados:')
    l
    fprintf('-------------------------------------------\n');
    fprintf('Valor medio de x:')
    mean(x)
    fprintf('-------------------------------------------\n');
    fprintf('Matriz de covarianzas:')
    cov(x)
    [v d] = eig(cov(x));
    fprintf('-------------------------------------------\n');
    fprintf('Autovalores:')
    d
    fprintf('-------------------------------------------\n');
```

```
    fprintf('Autovectores:')
    v
    fprintf('-----------------------------------------\n');

end
```

```
%%%%%%%%%%%%%%
% generaRadio %
%%%%%%%%%%%%%%

function r = generaRadio(d,a,ny,nx)

% Genera los radios de la distribucion. Los parametro son la dimension a y
% el exponente a (ver notas). El parametro de forma s se supone igual a 1.
% Devuelve una matriz de dimensiones ny x nx en la que cada elemento es una
% observacion del radio.

y = rand(ny,nx);
r = gammaincinv(y,d/a).^(1/a);
```

```
%%%%%%%%%%%%
% generaRot %
%%%%%%%%%%%%

function rot = generaRot(d)
% Genera una matriz de rotacion aleatoria en d dimensiones
% Para ello genera puntos segun una dimension normal y calcula la matriz de
% covarianzas de esos puntos. Si no hay sesgos en la funcion randn de
% matlab, dicha matriz de covarianzas tendra una orientacion completamente
% aleatoria. Tomamos entonces como matriz de rotacion la matriz de
% autovectores.
% El numero de puntos generados es 2*(d+1) para garantizar que la matriz de
% covarianzas tenga sentido.
x = randn(2*(d+1),d);
cm = cov(x);
[rot, ~] = eig(cm);
```

```
%%%%%%%%%%%%%%%%%
% generarObjetos %
%%%%%%%%%%%%%%%%%

% Ajusta los clusters de todos los problemas, genera unos objetos devueltos por la
funcion gmdistribution y los almacenamos en ./objetos_EM

close all;
clear all;
clc;
matlabpool close force;
matlabpool;



options = statset('MaxIter', 700);
```

```
dimension = [2 3 4 10];
distancia = 0:.2:5;
exponente = .2:.2:4;
num_problemas = 20;
ptos = [100 1000 10000 10000]; %puntos con los que vamos a realizar las pruebas
para 2D, 3D y 4D respectivamente

for dim = 1:length(dimension)


    for a = 1:length(exponente)
        for dist = 1:length(distancia)
            fprintf('\n\nDim %d exponente %g distancia %g\n\n',dimension(dim),
exponente(a), distancia(dist));
            % realizo para cada grado de libertad num_problemas, y almaceno como
valor
            % para ese grado el promedio de todos ellos.


            load(sprintf('../data/clusters-%d-%dD-a%.1f-dist%.1f.mat', ptos(dim),
dimension(dim), exponente(a), distancia(dist)));
            parfor problema = 1:num_problemas

                cont = 1;
                ir = 1;
                for k = 1:4
                    % realizo el metodo EM 10 veces para cada conjunto de datos e
                    % indice, y me quedo con el mejor resultado.
                    for i=1:10
                        flag = true;
                        while flag
                            try
                                objetos{problema}{ir} =
gmdistribution.fit(X{problema},k,'Options',options);
                                flag = false;
                                ir = ir +1
                            catch ex
                                if cont == 1
                                    cont= cont+1;
                                    fprintf('Se ha capturado una excepciÃ³n en
problema=%d, k=%d, i=%d\n',problema,k,i);
                                    disp(ex.message);
                                end
                            end
                        end
                    end

                end

            end
            save(sprintf('../objetos_EM/objetos-%dD-a%.1f-dist%.1f.mat',
dimension(dim), exponente(a), distancia(dist)), 'objetos');

        end
    end
end
matlabpool close force;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%
% calcularValoresIndices %
%%%%%%%%%%%%%%%%%%%%%%%%%

% Calcula los resultados de lpos indices AIC, BIC, DJ0, DJ1, DJ1_std, y DJ1_80 a
partir de los objetos guardados en
% "./objetos_EM". genera tablas para cada dimensiÃ³n, con estructuras en las que
se guarda el vaÃ±or medio y la desviaciá¹•on estandar de los datos. Para dibujar
el surf tomaremos las tablas generadas de valor medio
close all;
clear all;
clc;

matlabpool close force
matlabpool

% Datos inicializaciÃ³n
num_problemas = 20;
dimension = [2 3 4 10];
exponente = .2:.2:4;
distancia = 0:.2:5;
puntos = [100 1000 10000 10000];

% Cambiamos el numero maximo de iteraciones para ajustar las gaussianas a 700.
options = statset('MaxIter', 700);

% vector con los numeros de gaussianas ajustadas, como se hacen 10 pruebas con
cada ng salen 40 pruebas.
%las 10 primeras ajustando a 1 gaussiana, las 10 segundas a 2, etc
nem = [ones(1,10) 2.*ones(1,10) 3.*ones(1,10) 4.*ones(1,10)];



% Comenzamos el bucle enorme
for dim = 1:length(dimension)
    for a = 1:length(exponente)
        for dist = 1:length(distancia)
            fprintf('Dimension %d, exponente %.1f, distancia %1f\n',
dimension(dim), exponente(a), distancia(dist))
            load(sprintf('objetos_EM/objetos-%dD-a%.1f-dist%.1f.mat',
dimension(dim), exponente(a), distancia(dist)));
            load(sprintf('data/clusters-%d-%dD-a%.1f-dist%.1f.mat', puntos(dim),
dimension(dim), exponente(a), distancia(dist)));
            parfor probl = 1:num_problemas
                for ri = 1:40
                    valores(probl).objeto{ri} = objetos{probl}{ri};

                    % ValidaciÃ³n con el criterio AIC
                    valores(probl).AIC(ri) = objetos{probl}{ri}.AIC;
                    % Validacion con el criterio BIC
                    valores(probl).BIC(ri) = objetos{probl}{ri}.BIC;

                    [particion, ~] = cluster(objetos{probl}{ri} ,X{probl});

                    % Validacion con el criterio deltaJ corregido:
                    [valores(probl).dj0(ri)] =
computaDJkclusters(X{probl},particion);
                    % Validacion con el criterio deltaJ corregido:
```

```
                    [valores(probl).dj1(ri) valores(probl).std1(ri)] =
computaDJkclustersCorregido(X{probl},particion);

                    % Validacion con el criterio deltaJ corregido incluyendo HD:
                    [valores(probl).dj2(ri) valores(probl).std2(ri)] =
computaDJkclustersCorregidoConHD(X{probl},particion);

                end
            end
             save(sprintf('../resultados_indices/valores_indices-%dD-a%.1f-
dist%.1f.mat',dimension(dim), exponente(a), distancia(dist)), 'valores');
        end
    end
end

matlabpool close force
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%
% calcularSalidasIndices %
%%%%%%%%%%%%%%%%%%%%%%%%%

%Calcula las salidas de los indices AIC, BIC, DJO, DJ1, DJ1 teniendo en cuenta la
desviacion estandar y DJ1 con una confianza del 80%
% Los valores obtenidos son guardados en ./valores_indices

close all
clear all
clc


matlabpool close force
matlabpool


dimension = [2 3 4 10];
exponente = .2:.2:4;
distancia = 0:.2:5;

for dim = 1:length(dimension)
    for a = 1:length(exponente)
        parfor dist = 1:length(distancia)
            nAIC_p = zeros(1,20);
            nBIC_p = zeros(1,20);
            nDJ0_p = zeros(1,20);
            nDJ1_p = zeros(1,20);
            nDJ1_cn_std_p = zeros(1,20);
            nDJ1_80_p = zeros(1,20);

            fprintf('dimension %d, exponente %.1f, distancia %.1f\n',
dimension(dim), exponente(a), distancia(dist));
            nem = [ones(1,10) 2.*ones(1,10) 3.*ones(1,10) 4.*ones(1,10)];
            obj = load(sprintf('resultados_indices/valores_indices-%dD-a%.1f-
dist%.1f.mat', dimension(dim), exponente(a), distancia(dist)));
            for problema = 1:20
                % Validacion del criterio AIC
                [valores idx] = min(obj.valores(problema).AIC);
                nAIC_p(problema) = nem(idx);
```

```
                % Validacion del criterio BIC
                [valores idx] = min(obj.valores(problema).BIC);
                nBIC_p(problema) = nem(idx);

                % DJ0, minimo absoluto:
                [mdj0, idx] = min(obj.valores(problema).dj0);
                nDJ0_p(problema) = nem(idx);

                % DJ1, minimo absoluto:
                [mdj1, ixd] = min(obj.valores(problema).dj1);
                nDJ1_p(problema) = nem(idx);

                % DJ1, soluciones que solapan con el minimo absoluto:
                mdj1 = mdj1 + obj.valores(problema).std1(idx);
                daux = obj.valores(problema).dj1 - obj.valores(problema).std1;
                jx = find(daux < mdj1);
                nDJ1_cn_std_p(problema) = nem(jx(1));

                % DJ1, soluciones con valor de confianza 0.8
                nDJ1_80_p(problema) = getNGOptimo(obj.valores(problema).dj1,
obj.valores(problema).std1, nem, 0.8);


            end

            nAIC(a,dist) = mean(nAIC_p);
            nBIC(a,dist) = mean(nBIC_p);
            nDJ0(a,dist) = mean(nDJ0_p);
            nDJ1(a,dist) = mean(nDJ1_p);
            nDJ1_cn_std(a,dist) = mean(nDJ1_cn_std_p);
            nDJ1_80(a,dist) = mean(nDJ1_80_p);

        end

    end
    save(sprintf('../valores_figuras/valores_figuras-%dD.mat', dimension(dim)),
'nAIC', 'nBIC', 'nDJ0', 'nDJ1', 'nDJ1_cn_std', 'nDJ1_80');
end


matlabpool close force
```

```
%%%%%%%%%%%%%%%
% dibujarSurf %
%%%%%%%%%%%%%%%

% Dibuja las funciones surf

close all
clear all
clc


dimension = [2 3 4 10];
exponente = .2:.2:4;
distancia = 0:.2:5;
```

```
for dim = 1:length(dimension)
    load(sprintf('../valores_figuras/valores_figuras-%dD.mat',dimension(dim)))
    [X Y] = meshgrid(exponente,distancia);
    %AIC
    figure(1)
    surf(X,Y,nAIC');
    title(sprintf('AIC dimension %d', dimension(dim)));
    xlabel('Exponente');
    ylabel('Distancia');
    view(2)
    shading interp
    caxis([1 4]);
    colorbar
    saveas(figure(1),sprintf('../figures_axis/figure_AIC-%dD.fig',
dimension(dim)));
    saveas(figure(1),sprintf('../figures_axis/jpg/figure_AIC-
%dD.jpg',dimension(dim)),'jpg')
    %BIC
    figure(2)
    surf(X,Y,nBIC');
    title(sprintf('BIC dimension %d', dimension(dim)));
    xlabel('Exponente');
    ylabel('Distancia');
    view(2)
    shading interp
    caxis([1 4]);
    colorbar
    saveas(figure(2),sprintf('../figures_axis/figure_BIC-%dD.fig',
dimension(dim)));
    saveas(figure(2),sprintf('../figures_axis/jpg/figure_BIC-
%dD.jpg',dimension(dim)),'jpg')
    %DJ0
    figure(3)
    surf(X,Y,nDJ0');
    title(sprintf('DJ0 dimension %d', dimension(dim)));
    xlabel('Exponente');
    ylabel('Distancia');
    view(2)
    shading interp
    caxis([1 4]);
    colorbar
    saveas(figure(3),sprintf('../figures_axis/figure_DJ0-%dD.fig',
dimension(dim)));
    saveas(figure(3),sprintf('../figures_axis/jpg/figure_DJ0-
%dD.jpg',dimension(dim)),'jpg')
    %DJ1
    figure(4)
    surf(X,Y,nDJ1');
    title(sprintf('DJ1 dimension %d', dimension(dim)));
    xlabel('Exponente');
    ylabel('Distancia');
    view(2)
    shading interp
    colorbar
    saveas(figure(4),sprintf('../figures_axis/figure_DJ1-%dD.fig',
dimension(dim)));
    saveas(figure(4),sprintf('../figures_axis/jpg/figure_DJ1-
%dD.jpg',dimension(dim)),'jpg')
```

```
    %DJ1_cn_std
    figure(5)
    surf(X,Y,nDJ1_cn_std');
    title(sprintf('DJ1 con std dimension %d', dimension(dim)));
    xlabel('Exponente');
    ylabel('Distancia');
    view(2)
    shading interp
    caxis([1 4]);
    colorbar
    saveas(figure(5),sprintf('../figures_axis/figure_DJ1_cn_std-%dD.fig',
dimension(dim)));
    saveas(figure(5),sprintf('../figures_axis/jpg/figure_DJ1_cn_std-
%dD.jpg',dimension(dim)),'jpg')
    %DJ1_80
    figure(6)
    surf(X,Y,nDJ1_80');
    title(sprintf('DJ1 80 dimension %d', dimension(dim)));
    xlabel('Exponente');
    ylabel('Distancia');
    view(2)
    shading interp
    caxis([1 4]);
    colorbar
    saveas(figure(6),sprintf('../figures_axis/figure_DJ1_80-%dD.fig',
dimension(dim)));
    saveas(figure(6),sprintf('../figures_axis/jpg/figure_DJ1_80-
%dD.jpg',dimension(dim)),'jpg')
end
```

PRESUPUESTO

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... ..... 2.000 €
- Alquiler de impresora láser durante 6 meses ..... 60 €
- Material de oficina ..... 150 €
- Total de ejecución material ..... 2.300 €

2) Gastos generales

- 16 % sobre Ejecución Material ..... 368 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material ..... 138 €

4) Honorarios Proyecto

- 1000 horas a 15 € / hora ..... 15000 €

5) Material fungible

- Gastos de impresión ..... 90 €
- Encuadernación ..... 20 €

6) Subtotal del presupuesto

- Subtotal Presupuesto ..... 20126 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto ..... 4226.4 €

8) Total presupuesto

- Total Presupuesto ..... 13989,4 €

Madrid, Abril de 2015

El Ingeniero Jefe de Proyecto

Fdo.: Jesús Aragón Novo
Ingeniero de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un análisis de métodos de validación de clustering basado en la negentropía de las particiones. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es

obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.