Alternativas en Clustering Espectral

Santiago Neira Hernández

Universidad de los Andes

17 de Febrero 2022

Contenido

¿Por qué son relevantes los algoritmos de clustering?

- En la literatura de Machine Learning (ML) exite una rama de investigación conocida como el Aprendizaje no Supervisado, que a su vez tiene una sub-rama encargada de estudiar los algoritmos de clustering.

¿Por qué son relevantes los algoritmos de clustering?

- En la literatura de Machine Learning (ML) exite una rama de investigación conocida como el Aprendizaje no Supervisado, que a su vez tiene una sub-rama encargada de estudiar los algoritmos de clustering.
- Estos algoritmos sirven para clasificar los datos en subconjuntos significativos y manejables, donde cada grupo (llamado clúster) consiste en datos que son similares entre sí, y diferentes en comparación con los datos de otros grupos. [Everitt et al., 2011].

¿Por qué son relevantes los algoritmos de clustering?

- En la literatura de Machine Learning (ML) exite una rama de investigación conocida como el Aprendizaje no Supervisado, que a su vez tiene una sub-rama encargada de estudiar los algoritmos de clustering.
- Estos algoritmos sirven para clasificar los datos en subconjuntos significativos y manejables, donde cada grupo (llamado clúster) consiste en datos que son similares entre sí, y diferentes en comparación con los datos de otros grupos. [Everitt et al., 2011].
- Esta asignación en clusters refleja de alguna manera la estructura subyacente de las características que identifican y representan dichos datos.

¿Por qué son relevantes los algoritmos de clustering?

- En la literatura de Machine Learning (ML) exite una rama de investigación conocida como el Aprendizaje no Supervisado, que a su vez tiene una sub-rama encargada de estudiar los algoritmos de clustering.
- Estos algoritmos sirven para clasificar los datos en subconjuntos significativos y manejables, donde cada grupo (llamado clúster) consiste en datos que son similares entre sí, y diferentes en comparación con los datos de otros grupos. [Everitt et al., 2011].
- Esta asignación en clusters refleja de alguna manera la estructura subyacente de las características que identifican y representan dichos datos.
- Permiten plantear modelos que explican fenómenos relacionados con los datos.

Introducción

Como los algoritmos tienen varios parámetros, operan en espacios de gran dimensión y tienen que hacer frente a datos ruidosos e incompletos, su rendimiento varía para diferentes aplicaciones y tipos de datos.

Introducción

Como los algoritmos tienen varios parámetros, operan en espacios de gran dimensión y tienen que hacer frente a datos ruidosos e incompletos, su rendimiento varía para diferentes aplicaciones y tipos de datos.
Dichos algoritmos pueden ser aplicados a temas tan diversos como:

Introducción

Como los algoritmos tienen varios parámetros, operan en espacios de gran dimensión y tienen que hacer frente a datos ruidosos e incompletos, su rendimiento varía para diferentes aplicaciones y tipos de datos.
Dichos algoritmos pueden ser aplicados a temas tan diversos como:

- Los diagnósticos médicos [Alashwal et al., 2019].

Introducción

Como los algoritmos tienen varios parámetros, operan en espacios de gran dimensión y tienen que hacer frente a datos ruidosos e incompletos, su rendimiento varía para diferentes aplicaciones y tipos de datos.
Dichos algoritmos pueden ser aplicados a temas tan diversos como:

- Los diagnósticos médicos [Alashwal et al., 2019].
- La educación [Pavithra and Dhanaraj, 2019].

Introducción

Como los algoritmos tienen varios parámetros, operan en espacios de gran dimensión y tienen que hacer frente a datos ruidosos e incompletos, su rendimiento varía para diferentes aplicaciones y tipos de datos.
Dichos algoritmos pueden ser aplicados a temas tan diversos como:

- Los diagnósticos médicos [Alashwal et al., 2019].
- La educación [Pavithra and Dhanaraj, 2019].
- El análisis de documentos [Mohd ariff et al., 2018].

Introducción

Como los algoritmos tienen varios parámetros, operan en espacios de gran dimensión y tienen que hacer frente a datos ruidosos e incompletos, su rendimiento varía para diferentes aplicaciones y tipos de datos.
Dichos algoritmos pueden ser aplicados a temas tan diversos como:

- Los diagnósticos médicos [Alashwal et al., 2019].
- La educación [Pavithra and Dhanaraj, 2019].
- El análisis de documentos [Mohd ariff et al., 2018].
- El filtrado de spam [Sharma and Rastogi, 2014].

Introducción

Como los algoritmos tienen varios parámetros, operan en espacios de gran dimensión y tienen que hacer frente a datos ruidosos e incompletos, su rendimiento varía para diferentes aplicaciones y tipos de datos.
Dichos algoritmos pueden ser aplicados a temas tan diversos como:

- Los diagnósticos médicos [Alashwal et al., 2019].
- La educación [Pavithra and Dhanaraj, 2019].
- El análisis de documentos [Mohd ariff et al., 2018].
- El filtrado de spam [Sharma and Rastogi, 2014].
- La identificación de noticias falsas [Papalexakis, 2018].

Introducción

Siendo así, la tesis se enfoca en 3 grandes secciones:

Introducción

Siendo así, la tesis se enfoca en 3 grandes secciones:

1. Estudiar varios algoritmos de clustering clásicos:

Introducción

Siendo así, la tesis se enfoca en 3 grandes secciones:

1. Estudiar varios algoritmos de clustering clásicos:
   - $K-$medias.
   - Clustering Jerárquico aglomerativo
   - DBSCAN (Density Based Spatial Clustering of applications with Noise)

Introducción

Siendo así, la tesis se enfoca en 3 grandes secciones:

1. Estudiar varios algoritmos de clustering clásicos:
   - $K-$medias.
   - Clustering Jerárquico aglomerativo
   - DBSCAN (Density Based Spatial Clustering of applications with Noise)
2. Estudiar el algoritmo de clustering espectral y proponer una variación en su construcción.

Introducción

Siendo así, la tesis se enfoca en 3 grandes secciones:

1. Estudiar varios algoritmos de clustering clásicos:
   - $K-$medias.
   - Clustering Jerárquico aglomerativo
   - DBSCAN (Density Based Spatial Clustering of applications with Noise)
2. Estudiar el algoritmo de clustering espectral y proponer una variación en su construcción.
3. Comparar la implementación computacional de dichos algoritmos en diversas dimensiones y configuraciones espaciales de datos.

Preliminares

En todos los algoritmos a estudiar, se buscará entender cómo "partir" la nube de datos en subconjuntos de forma adecuada.

Preliminares

En todos los algoritmos a estudiar, se buscará entender cómo "partir" la nube de datos en subconjuntos de forma adecuada.

Definición (Partición)

Sea $\tilde{x} = (x_1, \cdots, x_n) \subseteq \mathbb{R}^d$ una nube de n datos en dimensión d. Decimos que la colección $S = \{S_1, \cdots, S_k\}$ es una partición de $\tilde{x}$ en k clusters sí:

- $\cup_{i=1}^{k} S_i = \tilde{x}$
- $S_i \cap S_j = \emptyset \quad \forall i \neq j$

Preliminares- Matrices de dispersión

Para una nube $\tilde{x} = (x_1, \cdots, x_n) \subseteq \mathbb{R}^d$ clasificadas en una partición $S = \{S_1, \cdots, S_k\}$. Consideramos las siguientes matrices de dispersión:

$$T_k = \sum_{j=1}^{k} \sum_{x_i \in S_j} (x_i - \bar{x})(x_i - \bar{x})' \qquad \text{Dispersión total} \qquad (1)$$

$$W_k = \sum_{j=1}^{k} \sum_{x_i \in S_j} (x_i - \mu_j)(x_i - \mu_j)' \qquad \text{Within (intra)} \qquad (2)$$

$$B_k = \sum_{j=1}^{k} \sum_{x_i \in S_j} |S_j|(\mu_j - \bar{x})(\mu_j - \bar{x})' \qquad \text{Between(inter)} \qquad (3)$$

Preliminares- Matrices de dispersión

Para una nube $\tilde{x} = (x_1, \cdots, x_n) \subseteq \mathbb{R}^d$ clasificadas en una partición $S = \{S_1, \cdots, S_k\}$. Consideramos las siguientes matrices de dispersión:

$$T_k = \sum_{j=1}^{k} \sum_{x_i \in S_j} (x_i - \bar{x})(x_i - \bar{x})' \qquad \text{Dispersión total} \qquad (1)$$

$$W_k = \sum_{j=1}^{k} \sum_{x_i \in S_j} (x_i - \mu_j)(x_i - \mu_j)' \qquad \text{Within (intra)} \qquad (2)$$

$$B_k = \sum_{j=1}^{k} \sum_{x_i \in S_j} |S_j|(\mu_j - \bar{x})(\mu_j - \bar{x})' \qquad \text{Between(inter)} \qquad (3)$$

Con $\bar{x} = \frac{1}{n} \sum x_i$ la media de los puntos en la nube de datos, $\mu_j = \frac{1}{|S_j|} \sum_{x \in S_j} x$ la media en cada cluster. $T_k = W_k + B_k$

Algoritmo de $K-$medias

El primer algoritmo a estudiar es el de $K-$medias. Este busca la partición de un conjunto en k clusters, tales que los clusters están conformados por datos que pertenece al clúster cuyo centroide es más cercano.

Algoritmo de $K-$medias

El primer algoritmo a estudiar es el de $K-$medias. Este busca la partición de un conjunto en k clusters, tales que los clusters están conformados por datos que pertenece al clúster cuyo centroide es más cercano.

Formalmente, para un conjunto de datos $\bar{x} = (x_1, \cdots, x_n) \subseteq \mathbb{R}^d$ el algoritmo busca una partición $S = \{S_1, \cdots, S_k\}$ tal que se minimice la varianza "within" cluster:

Algoritmo de $K-$medias

El primer algoritmo a estudiar es el de $K-$medias. Este busca la partición de un conjunto en k clusters, tales que los clusters están conformados por datos que pertenece al clúster cuyo centroide es más cercano.

Formalmente, para un conjunto de datos $\bar{x} = (x_1, \cdots, x_n) \subseteq \mathbb{R}^d$ el algoritmo busca una partición $S = \{S_1, \cdots, S_k\}$ tal que se minimice la varianza "within" cluster:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 = \arg\min_{S} Traza(W_k) \tag{4}$$

Algoritmo de K-medias ([Lloyd, 1982])

1. Escoger el número de clusters-k
2. Asignar aleatoriamente k-medias $m_1^{(1)}, \cdots, m_k^{(1)}$
3. Paso de asignación: Asignar cada observación al cluster con la media más cercana (en distancia euclídea):

$$S_i^{(t)} = \left\{ x_h : ||x_h - m_i^{(t)}||^2 \leq ||x_h - m_j^{(t)}||^2 \quad \forall j \in \{1, \cdots, k\} \right\}$$

4. Paso de actualización: Recalcular las medias (centroides) asignados en cada cluster:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_h \in S_i^{(t)}} x_h$$

5. Repetir pasos 3 y 4 hasta que no haya reasignación de centroides o hasta que $t = T_{max}$

$K-$medias. Escogencia de K

En la mayoría de casos, el número correcto de clústers no se conoce. Se pueden usar varias métodología de detección [Milligan and Cooper, 1985]:

$K-$medias. Escogencia de K

En la mayoría de casos, el número correcto de clústers no se conoce. Se pueden usar varias métodología de detección [Milligan and Cooper, 1985]:

1. ***El método del codo:*** Definiendo la Inercia como $I(k) = Traza(W_k) = \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2$. Se escoge a k como el "codo" o el punto en el cual la inercia tiende a estabilizarse

$K-$medias. Escogencia de K

En la mayoría de casos, el número correcto de clústers no se conoce. Se pueden usar varias métodología de detección [Milligan and Cooper, 1985]:

1. ***El método del codo:*** Definiendo la Inercia como $I(k) = Traza(W_k) = \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2$. Se escoge a k como el "codo" o el punto en el cual la inercia tiende a estabilizarse
2. ***Score de [Caliński and Harabasz, 1974]:*** Este método busca encontrar el k tal que:

$$k^* = \arg\max C_H(k) = \arg\max \frac{\frac{Traza(B_k)}{k-1}}{\frac{Traza(W_k)}{n-k}}$$

donde k^* maximiza la varianza between y minimiza la varianza within. Se introduce el factor $k-1$ y $n-k$ para corregir por proporcionalidad.

Discusión del método

- Una de las grandes ventajas del algoritmo es su complejidad computacional de implementación de $O(nkd)$ [Selim and Ismail, 1984].

Discusión del método

- Una de las grandes ventajas del algoritmo es su complejidad computacional de implementación de $O(nkd)$ [Selim and Ismail, 1984].
- Sin embargo, algunas desventajas del algoritmo son:
  - El algoritmo necesita una configuración esférica de los datos para funcionar correctamente, y asume que los centroides existen.

Discusión del método

- Una de las grandes ventajas del algoritmo es su complejidad computacional de implementación de $O(nkd)$ [Selim and Ismail, 1984].
- Sin embargo, algunas desventajas del algoritmo son:
  - El algoritmo necesita una configuración esférica de los datos para funcionar correctamente, y asume que los centroides existen.
  - El algoritmo no es determinista, por lo que el procedimiento puede no llevar a encontrar la partición óptima. [Hartigan and Wong, 1979]

Discusión del método

- Una de las grandes ventajas del algoritmo es su complejidad computacional de implementación de $O(nkd)$ [Selim and Ismail, 1984].
- Sin embargo, algunas desventajas del algoritmo son:
  - El algoritmo necesita una configuración esférica de los datos para funcionar correctamente, y asume que los centroides existen.
  - El algoritmo no es determinista, por lo que el procedimiento puede no llevar a encontrar la partición óptima. [Hartigan and Wong, 1979]
  - Tener el número de clusters como un híper parámetro siempre da cabida a errores a la hora de implementar el algoritmo.

Clustering Jerárquico

- El método de clustering jerárquico consiste en la costrucción de un árbol binario aglomerativo.

Clustering Jerárquico

- El método de clustering jerárquico consiste en la costrucción de un árbol binario aglomerativo.
- En dicho árbol los datos originales son las “hojas” del árbol, y el procedimiento se encarga de ir uniendo en pares aquellos sub-cojuntos que son más cercanos.

Clustering Jerárquico

- El método de clustering jerárquico consiste en la costrucción de un árbol binario aglomerativo.
- En dicho árbol los datos originales son las "hojas" del árbol, y el procedimiento se encarga de ir uniendo en pares aquellos sub-cojuntos que son más cercanos.
- Se define la distancia entre dos nodos como $d(x_i, x_j)$, y la distancia entre dos clusters como $\Delta(A, B)$, la cual se conoce como la distancia de enlace (linkage).

Clustering Jerárquico

Algunas de las construcciones más usuales para la métrica $\Delta(A, B)$ son:

$$\Delta_s(A, B) = \text{mín}\{d(x_i, x_j) : x_i \in A, x_j \in B\} \quad \text{Enlace Simple}$$

$$\Delta_c(A, B) = \text{máx}\{d(x_i, x_j) : x_i \in A, x_j \in B\} \quad \text{Enlace Completo}$$

$$\Delta_a(A, B) = \frac{1}{|A||B|} \sum_{x_i \in A, x_j \in B} d(x_i, x_j) \quad \text{Enlace promedio}$$

$$\Delta_W(A, B) = \frac{|A||B|}{|A| + |B|} ||\mu_A - \mu_B||^2 \quad \mu_A = \frac{1}{|A|} \sum_{x \in A} x \quad \text{Enlace Ward}$$

Algoritmo [Corredor and Quiroz, 2020]

1. Inicializar cada punto como un cluster: $S_i = \{x_i\}, i \leq n$
2. Asigne $t, g = n$ (t es el índice del último cluster formado, g es el número de clusters existentes)
3. Seleccione los 2 clusters que son los más cercanos, de acuerdo con la distancia Δ construida. Suponga que son S_i, S_j. Asigne $S_{t+1} = S_i \cup S_j$.
4. Actualice $t = t + 1, g = g - 1$
5. Guarde la altura del agrupamiento $(t - n)$ como $h(t - n) = \Delta(A_i, A_j)$
6. Conecte S_i, S_j a S_t en el árbol a la altura $h(t - n)$, elimine a S_j, S_i.
7. Repetir pasos 3 al 6 mientras $g > 1$.
8. Haga un corte a la algura $h(\cdot)$ tal que queden k sub-árboles por debajo.
9. Retorne la partición S

Escogencia de k

- Siguiendo a [Milligan and Cooper, 1985] y a [Everitt et al., 2011], se pueden considerar las medidas de detección del número de clusters del método del codo y el escore de Calinski- Harabasz.

Escogencia de k

- Siguiendo a [Milligan and Cooper, 1985] y a [Everitt et al., 2011], se pueden considerar las medidas de detección del número de clusters del método del codo y el escore de Calinski- Harabasz.
- Existe otra medida, propuesta por [Mojena, 1977] el cual escoge el número de clusters k cuando

$$\alpha_{k+1} > \bar{\alpha} + \kappa S_{\alpha}$$

Escogencia de k

- Siguiendo a [Milligan and Cooper, 1985] y a [Everitt et al., 2011], se pueden considerar las medidas de detección del número de clusters del método del codo y el escore de Calinski- Harabasz.
- Existe otra medida, propuesta por [Mojena, 1977] el cual escoge el número de clusters k cuando

$$\alpha_{k+1} > \bar{\alpha} + \kappa S_{\alpha}$$

donde $\alpha_0, \cdots, \alpha_{n-1}$ son las alturas de fusión correspondientes a las etapas con $n, n-1, \cdots 1$ clusters. $\bar{\alpha}$ es la media de los $k-$ésimos anteriores valores de fusión, y s_{α} hace referencia a la desviación estandar insesgada de los mismos. κ es una constante (2.75-3.50).

Discusión del algoritmo

- Una gran desventaja de usar el Enlace simple es el fenómeno de Chaining. Encadenamiento entre estructuras que son disímiles debido a presencia de datos “intermedios” entre las nubes.

Discusión del algoritmo

- Una gran desventaja de usar el Enlace simple es el fenómeno de Chaining. Encadenamiento entre estructuras que son disímiles debido a presencia de datos "intermedios" entre las nubes.
- Si la estructura del árbol es relevante, la construcción del árbol sólamente es determinista para el caso de enlace promedio.

Discusión del algoritmo

- Una gran desventaja de usar el Enlace simple es el fenómeno de Chaining. Encadenamiento entre estructuras que son disímiles debido a presencia de datos "intermedios" entre las nubes.
- Si la estructura del árbol es relevante, la construcción del árbol sólamente es determinista para el caso de enlace promedio.
- De acuerdo con [Nielsen, 2016], a pesar de que Enlace Simple tiene el fenómeno del chaining, se sabe que éste es el único algoritmo implementable en tiempo $O(n^2)$.

Discusión del algoritmo

- Una gran desventaja de usar el Enlace simple es el fenómeno de Chaining. Encadenamiento entre estructuras que son disímiles debido a presencia de datos "intermedios" entre las nubes.
- Si la estructura del árbol es relevante, la construcción del árbol sólamente es determinista para el caso de enlace promedio.
- De acuerdo con [Nielsen, 2016], a pesar de que Enlace Simple tiene el fenómeno del chaining, se sabe que éste es el único algoritmo implementable en tiempo $O(n^2)$.
- Más aún, sólo se saben resultados de consistencia para este tipo de enlace. Relacionado con la construcción del MST

Algoritmo DBSCAN

- El DBSCAN (Density-based spatial clustering of applications with noise) es un algoritmo de clusterización basado en [Ester et al., 1996].

Algoritmo DBSCAN

- El DBSCAN (Density-based spatial clustering of applications with noise) es un algoritmo de clusterización basado en [Ester et al., 1996].
- Es un algoritmo no paramétrico y es "density-based": los clusters se definen como áreas de alta concentración de puntos (con alguna medida de concentración) en comparación con el resto de la nube. Los parámetros de este algoritmo son:

Algoritmo DBSCAN

- El DBSCAN (Density-based spatial clustering of applications with noise) es un algoritmo de clusterización basado en [Ester et al., 1996].
- Es un algoritmo no paramétrico y es "density-based": los clusters se definen como áreas de alta concentración de puntos (con alguna medida de concentración) en comparación con el resto de la nube. Los parámetros de este algoritmo son:
  - $\epsilon :=$ La distancia máxima (euclídea) entre 2 puntos. Dos puntos se consideran como vecinos sí y sólo sí están separados por una distancia menor o igual a ϵ.
  - *MinPoints* $:=$ El mínimo número de puntos requerido para formar un cluster denso.

Algunas definiciones para DBSCAN

Consideramos $\bar{x} = (x_1, \cdots, x_n) \in \mathbb{R}^d$. Definimos la vecindad de x_i, $N_\epsilon(x_i) = \{x_j | d(x_i, x_j) \leq \epsilon\}$ como el conjunto de vecinos que caen en la bola cerrada de tamaño ϵ. Con esto podemos clasificar puntos de la siguiente forma:

- ***Punto nuclear:*** Un punto x_i es nuclear si $|N_\epsilon(x_i)| \geq MinPoints$

Algunas definiciones para DBSCAN

Consideramos $\bar{x} = (x_1, \cdots, x_n) \in \mathbb{R}^d$. Definimos la vecindad de x_i, $N_\epsilon(x_i) = \{x_j | d(x_i, x_j) \leq \epsilon\}$ como el conjunto de vecinos que caen en la bola cerrada de tamaño ϵ. Con esto podemos clasificar puntos de la siguiente forma:

- ***Punto nuclear:*** Un punto x_i es nuclear si $|N_\epsilon(x_i)| \geq MinPoints$
- ***Alcanzable de forma densa:*** Un punto x_j es alcanzable de forma densa a partir de un punto x_i si:
  1. $x_j \in N_\epsilon(x_i)$
  2. $|N_\epsilon(x_i)| \geq MinPoints$

Algunas definiciones para DBSCAN

Consideramos $\bar{x} = (x_1, \cdots, x_n) \in \mathbb{R}^d$. Definimos la vecindad de x_i, $N_\epsilon(x_i) = \{x_j | d(x_i, x_j) \leq \epsilon\}$ como el conjunto de vecinos que caen en la bola cerrada de tamaño ϵ. Con esto podemos clasificar puntos de la siguiente forma:

- ***Punto nuclear:*** Un punto x_i es nuclear si $|N_\epsilon(x_i)| \geq MinPoints$
- ***Alcanzable de forma densa:*** Un punto x_j es alcanzable de forma densa a partir de un punto x_i si:
  1. $x_j \in N_\epsilon(x_i)$
  2. $|N_\epsilon(x_i)| \geq MinPoints$
- ***Directamente alcanzable:*** Un punto x_j es directamente alcanzable por un punto x_i si hay una cadena de puntos $x_{j1}, \cdots, x_{jn}$ con $x_{j1} = x_j, x_{jn} = x_i$ tales que x_{jk} es alcanzable de forma densa a partir de $x_{j(k+1)}$

Definiciones

- ***Punto de ruido:*** Si un punto x_i no es directamente alcanzable por ningún otro punto se llama un punto de ruido.

Definiciones

- ***Punto de ruido:*** Si un punto x_i no es directamente alcanzable por ningún otro punto se llama un punto de ruido.
- ***Cluster:*** Un cluster S es un subcojunto de $\bar{x}$ que satisface las siguientes condiciones:

Definiciones

- ***Punto de ruido:*** Si un punto x_i no es directamente alcanzable por ningún otro punto se llama un punto de ruido.
- ***Cluster:*** Un cluster S es un subcojunto de $\bar{x}$ que satisface las siguientes condiciones:
  - $\forall x_i, x_j$: si $x_i \in S$ y x_j es alcanzable de forma densa desde x_i, entonces $x_j \in S$
  - $\forall x_i, x_j \in S \quad \exists x_k \in S$ tal que tanto x_i como x_j pueden ser directamente alcanzables por x_k

Algoritmo

1. Escoja un punto arbitrario $x_i \in \bar{x}$. Si $|N_\epsilon(x_i)| \geq \mathit{MinPoints}$ la construcción del cluster empieza. De lo contrario el punto se clasifica como punto de ruido.
2. Como x_i fue clasificado como punto nuclear, x_j será añadido al cluster $\forall x_j \in N_\epsilon(x_j)$
3. El paso se repite para todos los $x_{jk} \in N_\epsilon(x_j)$
4. El paso 3 se realiza recursivamente hasta que el cluster es completamente encontrado. (que ya no se puedan añadir más datos en el cluster)
5. Repetir puntos 1 al 4 con nuevos puntos que pueden ser parte de un nuevo cluster u originalmente clasificados como puntos de ruido.

Escogencia de ϵ y *MinPoints*

- Por regla del pulgar se approxima *MinPoints* a partir del número de dimensiones en el espacio d, procurando que $\mathit{MinPoints} \geq d + 1$. Usualmente se usa $\mathit{MinPoints} = 2 \cdot d$.

Escogencia de ϵ y *MinPoints*

- Por regla del pulgar se approxima *MinPoints* a partir del número de dimensiones en el espacio d, procurando que $MinPoints \geq d + 1$. Usualmente se usa $MinPoints = 2 \cdot d$.
- Para el valor de ϵ se puede usar una variante del método del codo, En esta variante se calcula la distancia al $k = MinPoints$ vecino más cercano de todos los puntos, se organiza de menor a mayor dicha distancia y se escoge a ϵ como aquel valor que sea el "codo".[Schubert et al., 2017].

Discusión

- Este algoritmo necesita un tiempo de $O(n^2)$ lo cual lo hace relativamente eficiente. Encuentra el número de clústers de forma endógena

Discusión

- Este algoritmo necesita un tiempo de $O(n^2)$ lo cual lo hace relativamente eficiente. Encuentra el número de clústers de forma endógena
- Sin embargo, hay que resaltar que:
  - El algoritmo no es determinístico. Puede darse el problema de chaining de enlace simple para puntos en el borde entre clústers.

Discusión

- Este algoritmo necesita un tiempo de $O(n^2)$ lo cual lo hace relativamente eficiente. Encuentra el número de clústers de forma endógena
- Sin embargo, hay que resaltar que:
  - El algoritmo no es determinístico. Puede darse el problema de chaining de enlace simple para puntos en el borde entre clústers.
  - El algoritmo no es bueno agrupando datos con diferencias en densidades,

Discusión

- Este algoritmo necesita un tiempo de $O(n^2)$ lo cual lo hace relativamente eficiente. Encuentra el número de clústers de forma endógena
- Sin embargo, hay que resaltar que:
  - El algoritmo no es determinístico. Puede darse el problema de chaining de enlace simple para puntos en el borde entre clústers.
  - El algoritmo no es bueno agrupando datos con diferencias en densidades,
  - La escogencia del ϵ puede ser considerablemente compleja cuando los datos y su escala no son bien conocidos.

Clustering Espectral- Introducción

Antes de ver el algoritmo de clustering espectral (gran parte del core del trabajo de grado), veremos algunas consideraciones teóricas relevantes. Nos basamos fuertemente en [Von Luxburg, 2007] y en la teoría de Grafos espectral.

Clustering Espectral- Introducción

Antes de ver el algoritmo de clustering espectral (gran parte del core del trabajo de grado), veremos algunas consideraciones teóricas relevantes. Nos basamos fuertemente en [Von Luxburg, 2007] y en la teoría de Grafos espectral.

- Dado un set de datos $\bar{x} = (x_1, \cdots, x_n) \subseteq \mathbb{R}^d$ y alguna noción de similaridad $s_{ij} \geq 0$, definimos el *Grafo de Similaridad* $G(V, E)$ donde $v_i \in V$ representa un vértice en el grafo y hace alusión a un punto x_i y la arista tiene un peso de s_{ij}.

Clustering Espectral- Introducción

Antes de ver el algoritmo de clustering espectral (gran parte del core del trabajo de grado), veremos algunas consideraciones teóricas relevantes. Nos basamos fuertemente en [Von Luxburg, 2007] y en la teoría de Grafos espectral.

- Dado un set de datos $\bar{x} = (x_1, \cdots, x_n) \subseteq \mathbb{R}^d$ y alguna noción de similaridad $s_{ij} \geq 0$, definimos el *Grafo de Similaridad* $G(V, E)$ donde $v_i \in V$ representa un vértice en el grafo y hace alusión a un punto x_i y la arista tiene un peso de s_{ij}.
- De esta forma el problema de clusterización puede reformularse como aquel problema que encuentra una partición del grafo tales que vértices entre diferentes grupos tengan pesos suficientemente bajos, y tales que vértices en el grupo tengan pesos altos.

Notación de Grafos- Definiciones

Sea $G(V, E)$ un grafo no dirigido (cuyas aristas representan relaciones simétricas entre vértices) con V conjunto de vértices, E conjunto de aristas y con $W = (w_{ij})_{i,j=1,\cdots,n}$ la matriz de pesos del grafo. $w_{ij} \geq 0 \ \forall v_i, v_j \in V$
Veamos algunas definiciones importantes:

Notación de Grafos- Definiciones

Sea $G(V, E)$ un grafo no dirigido (cuyas aristas representan relaciones simétricas entre vértices) con V conjunto de vértices, E conjunto de aristas y con $W = (w_{ij})_{i,j=1,\cdots,n}$ la matriz de pesos del grafo. $w_{ij} \geq 0 \ \forall v_i, v_j \in V$ Veamos algunas definiciones importantes:

- ***Matriz de grados:*** La matriz de grados D es aquella matriz diagonal cuyas entradas son $d_1, \cdots, d_n$ donde:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

Notación de Grafos- Definiciones

Sea $G(V, E)$ un grafo no dirigido (cuyas aristas representan relaciones simétricas entre vértices) con V conjunto de vértices, E conjunto de aristas y con $W = (w_{ij})_{i,j=1,\cdots,n}$ la matriz de pesos del grafo. $w_{ij} \geq 0 \ \forall v_i, v_j \in V$ Veamos algunas definiciones importantes:

- ***Matriz de grados:*** La matriz de grados D es aquella matriz diagonal cuyas entradas son $d_1, \cdots, d_n$ donde:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

- Dado un set de vértices $A \subset V$, se define a $\bar{A} := V \setminus A$

Notación de Grafos- Definiciones

Sea $G(V, E)$ un grafo no dirigido (cuyas aristas representan relaciones simétricas entre vértices) con V conjunto de vértices, E conjunto de aristas y con $W = (w_{ij})_{i,j=1,\cdots,n}$ la matriz de pesos del grafo. $w_{ij} \geq 0 \ \forall v_i, v_j \in V$ Veamos algunas definiciones importantes:

- ***Matriz de grados:*** La matriz de grados D es aquella matriz diagonal cuyas entradas son $d_1, \cdots, d_n$ donde:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

- Dado un set de vértices $A \subset V$, se define a $\bar{A} := V \setminus A$
- Se define a $\mathbb{1}_A = (f_1, \cdots, f_n)' \in \mathbb{R}^n$ como el vector cuyas entradas $f_i = 1$ si $v_i \in A$ y $f_i = 0$ de lo contrario.

Definiciones

Definiciones

- ***Conexión:*** Decimos que $A, B \subset V$ tienen una conexión sí $\exists v_i \in A, v_j \in B, w_{ij} > 0$.

Definiciones

- ***Conexión:*** Decimos que $A, B \subset V$ tienen una conexión sí $\exists v_i \in A, v_j \in B, w_{ij} > 0$.
- ***Camino:*** Sean $v_i, v_j \in V$. se dice que existe un camino entre v_i y v_j sí hay una sucesión de vértices $v_1 = v_i, \cdots v_m = v_j$ tal que $w_{ll+1} > 0 \quad \forall \quad 1 \leq l \leq m$

Definiciones

- ***Conexión:*** Decimos que $A, B \subset V$ tienen una conexión sí $\exists v_i \in A, v_j \in B, w_{ij} > 0$.
- ***Camino:*** Sean $v_i, v_j \in V$. se dice que existe un camino entre v_i y v_j sí hay una sucesión de vértices $v_1 = v_i, \cdots v_m = v_j$ tal que $w_{ll+1} > 0 \quad \forall \quad 1 \leq l \leq m$
- ***Conjunto conexo:*** Se dice que $A \subset V$ es conexo, si cualesquiera dos vértices en A pueden ser unidos por un camino tal que todos los caminos intermedios estén en A.

Definiciones

- ***Conexión:*** Decimos que $A, B \subset V$ tienen una conexión sí $\exists v_i \in A, v_j \in B, w_{ij} > 0$.
- ***Camino:*** Sean $v_i, v_j \in V$. se dice que existe un camino entre v_i y v_j sí hay una sucesión de vértices $v_1 = v_i, \cdots v_m = v_j$ tal que $w_{ll+1} > 0 \quad \forall \quad 1 \leq l \leq m$
- ***Conjunto conexo:*** Se dice que $A \subset V$ es conexo, si cualesquiera dos vértices en A pueden ser unidos por un camino tal que todos los caminos intermedios estén en A.
- ***Componente conexo:*** Se dice que $A \subset V$ es un componente conexo de V si es conexo y no hay conexiones entre A y $\bar{A}$.

Laplaciano

El laplaciano (no normalizado) de un grafo $V(G, E)$ con matriz de pesos W es definido como:

$$L = D - W$$

Laplaciano

El laplaciano (no normalizado) de un grafo $V(G, E)$ con matriz de pesos W es definido como:

$$L = D - W$$

Proposición: (Propiedades de L) La matriz L satisface las siguienes propiedades:

Laplaciano

El laplaciano (no normalizado) de un grafo $V(G, E)$ con matriz de pesos W es definido como:

$$L = D - W$$

Proposición: (Propiedades de L) La matriz L satisface las siguienes propiedades:

1. Para cualquier vector $f \in \mathbb{R}^n$:

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Laplaciano

El laplaciano (no normalizado) de un grafo $V(G, E)$ con matriz de pesos W es definido como:

$$L = D - W$$

Proposición: (Propiedades de L) La matriz L satisface las siguienes propiedades:

1. Para cualquier vector $f \in \mathbb{R}^n$:

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

2. L es simétrica y semi-definida positiva

Laplaciano

El laplaciano (no normalizado) de un grafo $V(G, E)$ con matriz de pesos W es definido como:

$$L = D - W$$

Proposición: (Propiedades de L) La matriz L satisface las siguienes propiedades:

1. Para cualquier vector $f \in \mathbb{R}^n$:

$$f' L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

2. L es simétrica y semi-definida positiva
3. El autovalor más pequeño de L es 0 correspondiente al autovector $\mathbb{1}$

Laplaciano

El laplaciano (no normalizado) de un grafo $V(G, E)$ con matriz de pesos W es definido como:

$$L = D - W$$

Proposición: (Propiedades de L) La matriz L satisface las siguienes propiedades:

1. Para cualquier vector $f \in \mathbb{R}^n$:

$$f' L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

2. L es simétrica y semi-definida positiva
3. El autovalor más pequeño de L es 0 correspondiente al autovector $\mathbb{1}$
4. L tiene n autovalores no negativos $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Demostración:

Demostración:

1. Por la definición de d_i:

$$f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\Big(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{i=1}^{n} d_j f_j^2\Big) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Demostración:

1. Por la definición de d_i:

$$f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\Big(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{i=1}^{n} d_j f_j^2\Big) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

2. Como D, W son simétricas, L debe serlo. Además, por (1) L es semidefinida positiva, puesto que $f'Lf \geq 0 \quad \forall f \in \mathbb{R}^n$

Demostración:

1. Por la definición de d_i:

$$f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\Big(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{i=1}^{n} d_j f_j^2\Big) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

2. Como D, W son simétricas, L debe serlo. Además, por (1) L es semidefinida positiva, puesto que $f'Lf \geq 0 \quad \forall f \in \mathbb{R}^n$
3. Como en cada columna la suma de entradas de L da 0, L es singular.

Demostración:

1. Por la definición de d_i:

$$f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\Big(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{i=1}^{n} d_j f_j^2\Big) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

2. Como D, W son simétricas, L debe serlo. Además, por (1) L es semidefinida positiva, puesto que $f'Lf \geq 0 \quad \forall f \in \mathbb{R}^n$
3. Como en cada columna la suma de entradas de L da 0, L es singular.
4. Consecuencia de (1)-(3) y del Teorema Espectral

□

Laplaciano

Proposición:(Número de componentes conexas y el Espectro de L) Sea G un grafo no dirigido con pesos no-negativos. Entonces la multiplicidad k del autovalor 0 de L es igual al número de componentes conexas $A_1, \cdots, A_k$ del grafo. El autoespacio del autovector 0 puede ser visto como el $span\{\mathbb{1}_{A_1}, \cdots, \mathbb{1}_{A_k}\}$.

Laplaciano

Demostración:

- $k = 1$. Suponga que f es un autovector con autovalor 0. Luego es cierto que:

$$0 = f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Laplaciano

Demostración:

- $k = 1$. Suponga que f es un autovector con autovalor 0. Luego es cierto que:

$$0 = f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Como $w_{ij} \geq 0$ la única forma que la expresión se anule es porque $f_i = f_j$. Entonces f debe ser constante para todos los vértices que pueden ser conectados por un camino en el grafo.

Laplaciano

Demostración:

- $k = 1$. Suponga que f es un autovector con autovalor 0. Luego es cierto que:

$$0 = f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Como $w_{ij} \geq 0$ la única forma que la expresión se anule es porque $f_i = f_j$. Entonces f debe ser constante para todos los vértices que pueden ser conectados por un camino en el grafo. Como todos los vértices de un componente conexo de un grafo pueden ser conectados por un camino, f es constante en todo el componente conexo.

Laplaciano

Demostración:

- $k = 1$. Suponga que f es un autovector con autovalor 0. Luego es cierto que:

$$0 = f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

Como $w_{ij} \geq 0$ la única forma que la expresión se anule es porque $f_i = f_j$. Entonces f debe ser constante para todos los vértices que pueden ser conectados por un camino en el grafo. Como todos los vértices de un componente conexo de un grafo pueden ser conectados por un camino, f es constante en todo el componente conexo. Como sólo hay un componente conexo, el autovector *debe* ser $\mathbb{1}$ cuyo autovalor es 0.

Caso de k componentes conexas: Asumiendo sin pérdida de generalidad que los vértices están ordenados de acuerdo con las componentes conexas a donde pertenecen.

Caso de k componentes conexas: Asumiendo sin pérdida de generalidad que los vértices están ordenados de acuerdo con las componentes conexas a donde pertenecen. Luego L puede verse como:

$$L = \begin{pmatrix} L_1 & 0 & \cdots & 0 \\ 0 & L_2 & \ddots & \vdots \\ \vdots & \ddots & L_{k-1} & 0 \\ 0 & \cdots & 0 & L_k \end{pmatrix}$$

Caso de k componentes conexas: Asumiendo sin pérdida de generalidad que los vértices están ordenados de acuerdo con las componentes conexas a donde pertenecen. Luego L puede verse como:

$$L = \begin{pmatrix} L_1 & 0 & \cdots & 0 \\ 0 & L_2 & \ddots & \vdots \\ \vdots & \ddots & L_{k-1} & 0 \\ 0 & \cdots & 0 & L_k \end{pmatrix}$$

Notemos que cada L_i es un laplaciano en el sub-grafo del $i-$ésimo componente conexo. Sabemos que, como L es diagonal por bloques, el espectro de L es la unión de los espectros de L_i y los autovectores de L serán los autovectores de L_i, donde se llenan de 0's las posiciones de los demás bloques.

Como cada L_i es el laplaciano de un grafo conexo, sabemos por el caso de $k = 1$ que todo L_i tendrá un autovalor 0 con multiplicidad 1, y el correspondiente autocvector es el vector de 1's en el $i-$ésimo componente conexo.

Como cada L_i es el laplaciano de un grafo conexo, sabemos por el caso de $k = 1$ que todo L_i tendrá un autovalor 0 con multiplicidad 1, y el correspondiente autocvector es el vector de 1's en el $i-$ésimo componente conexo. Luego L tiene tantos autovalores 0 como componentes conexas, y los autovectores correspondientes son los vectores indicadores de las componentes conexas. □

Algoritmo

1. Construya un grafo de similaridad S con alguna función simétrica y no negativa
2. Encuentre el Laplaciano no normalizado L
3. Encuentre los primeros k autovectores $u_1, \cdots, u_k$ de L
4. Sea $U \in \mathbb{R}^{n \times k}$ la matriz que contiene los vectores $u_1, \ldots, u_k$ como columnas
5. Para $i =, 1 \cdots, n$ asigne $y_i \in \mathbb{R}^k$ el vector correspondiente a la $i-$ésima fila de U.
6. Realice el procedimiento de clustering para $(y_i)_{i=1,\cdots,n}$ en $\mathbb{R}^k$ con el algoritmo de $k-$medias. Clasifique dichos clusters como $C_1, \cdots, C_k$
7. Retorne $A_1, \cdots, A_n$ con $A_i = \{j | y_j \in C_i\}$

Sobre la matriz de similaridad

Algunas de las matrices de similaridad más usadas son:

- **Matriz de la $\epsilon-$vecindad**:

$$s_{ij} = \begin{cases} d(x_i, x_j) & d(x_i, x_j) \leq \epsilon \\ 0 & d(x_i, x_j) > \epsilon \end{cases}$$

Sobre la matriz de similaridad

Algunas de las matrices de similaridad más usadas son:

- **Matriz de la $\epsilon-$vecindad**:

$$s_{ij} = \begin{cases} d(x_i, x_j) & d(x_i, x_j) \leq \epsilon \\ 0 & d(x_i, x_j) > \epsilon \end{cases}$$

- **Matriz de los K-vecinos más cercanos (KNN)**: En este grafo se conecta a los vértices v_i con v_j si $v_j \in NN^k(v_i)$ con

$$NN^k(v_i) := \{v_j | d(x_j, x_i) \text{ está en los mínimos k valores de } d(x'_j, x_i) \quad \forall x'_j \in \bar{x}\}$$

Este grafo se hace no dirigido cuando sólamente se piensa en las consexiones del vértice i o j.

Sobre la matriz de similaridad

Algunas de las matrices de similaridad más usadas son:

- **Matriz de la $\epsilon-$vecindad**:

$$s_{ij} = \begin{cases} d(x_i, x_j) & d(x_i, x_j) \leq \epsilon \\ 0 & d(x_i, x_j) > \epsilon \end{cases}$$

- **Matriz de los K-vecinos más cercanos (KNN)**: En este grafo se conecta a los vértices v_i con v_j si $v_j \in NN^k(v_i)$ con

$$NN^k(v_i) := \{v_j | d(x_j, x_i) \text{ está en los mínimos k valores de } d(x'_j, x_i) \quad \forall x'_j \in \bar{x}\}$$

Este grafo se hace no dirigido cuando sólamente se piensa en las consexiones del vértice i o j.

$$s_{ij} = \begin{cases} d(x_i, x_j) & v_j \in NN^k(v_i) \\ 0 & v_j \notin NN^k(v_i) \end{cases} \qquad k \simeq \lfloor \ln N \rfloor$$

Escogiendo k

Para escoger el valor de k se puede realizar:

Escogiendo k

Para escoger el valor de k se puede realizar:

- Endogeneizar el número de clusters buscando la multiplicidad algebraica del autovalor 0 de la descomposición espectral del laplaciano asociado.

Escogiendo k

Para escoger el valor de k se puede realizar:

- Endogeneizar el número de clusters buscando la multiplicidad algebraica del autovalor 0 de la descomposición espectral del laplaciano asociado.
- **Eigen-gap**: Escoger k tal que $|\lambda_k - \lambda_{k-1}|$ es relativamente grande.

Escogiendo k

Para escoger el valor de k se puede realizar:

- Endogeneizar el número de clusters buscando la multiplicidad algebraica del autovalor 0 de la descomposición espectral del laplaciano asociado.
- **Eigen-gap**: Escoger k tal que $|\lambda_k - \lambda_{k-1}|$ es relativamente grande.
- Usar los métodos del codo o de calinski-harabasz en el espacio proyectado.

Discusión del algoritmo

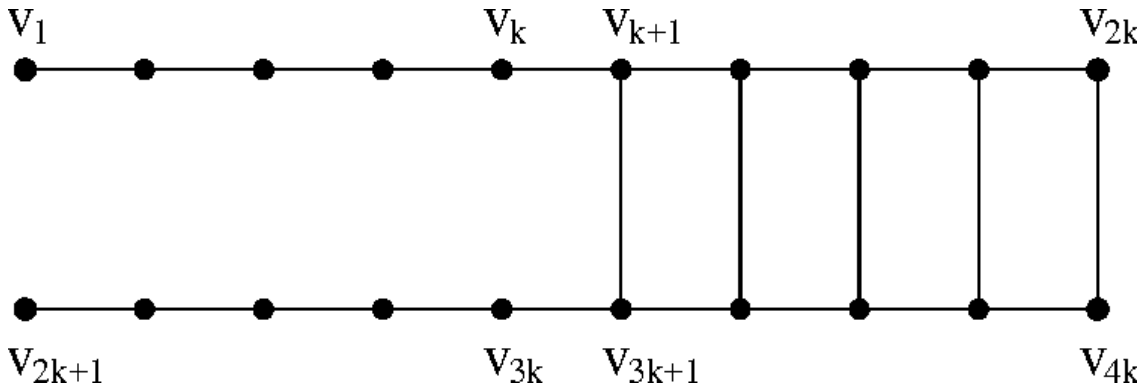


Figura 1: El grafo de la cucaracha de [Guattery and Miller, 1998]

Discusión del algoritmo

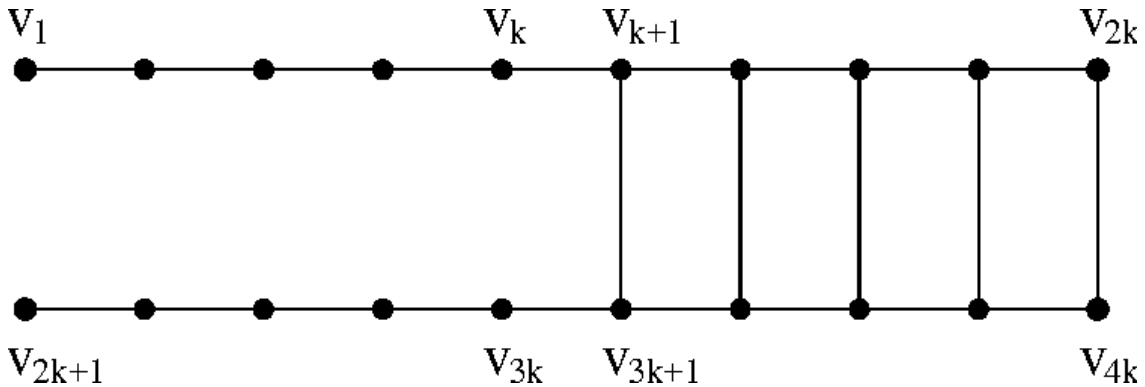


Figura 1: El grafo de la cucaracha de [Guattery and Miller, 1998]

- En este grafo, la partición óptima debería ser $A = \{v_1, \cdots, v_k, v_{2k+1}, \cdots, v_{3k}\}$ y $\bar{A} = \{v_{k+1}, \cdots, v_{2k}, v_{3k+1}, \cdots, v_{4k}\}$, sin embargo la partición inducida por el algoritmo de Clustering espectral es $B = \{v_1, \cdots, v_k, v_{k+1}, \cdots, v_{2k}\}$, $\bar{B} = \{v_{2k+1}, \cdots, v_{3k}, v_{3k+1}, \cdots, v_{4k}\}$.

Variación- Grafo de esferas de Influencia

Definimos el grafo de esferas de influencia con una matriz de pesos dada por:

$$w_{ij} = \begin{cases} \frac{\rho_k(x_i)+\rho_k(x_j)}{d(x_i,x_j)} & \rho_k(x_i) + \rho_k(x_j) \geq d(x_i, x_j), \quad x_i \neq x_j \\ 0 & c.o.p \end{cases} \tag{5}$$

Variación- Grafo de esferas de Influencia

Definimos el grafo de esferas de influencia con una matriz de pesos dada por:

$$w_{ij} = \begin{cases} \frac{\rho_k(x_i)+\rho_k(x_j)}{d(x_i,x_j)} & \rho_k(x_i) + \rho_k(x_j) \geq d(x_i, x_j), \quad x_i \neq x_j \\ 0 & c.o.p \end{cases} \tag{5}$$

Donde $\rho_k(x_i) = d(x_i, NN_k(x_i))$ y $NN_k(x_i)$ hace referencia al $k-$ésimo vecino más cercano de x_i.

Variación- Grafo de esferas de Influencia

Definimos el grafo de esferas de influencia con una matriz de pesos dada por:

$$w_{ij} = \begin{cases} \frac{\rho_k(x_i)+\rho_k(x_j)}{d(x_i,x_j)} & \rho_k(x_i) + \rho_k(x_j) \geq d(x_i, x_j), \quad x_i \neq x_j \\ 0 & c.o.p \end{cases} \tag{5}$$

Donde $\rho_k(x_i) = d(x_i, NN_k(x_i))$ y $NN_k(x_i)$ hace referencia al $k-$ésimo vecino más cercano de x_i. Este grafo da un peso positivo cuando las hiper-esferas a los vecinos más cercanos de los vértices se intersectan, y es una variante del grafo de k-vecinos mutuos de [Brito et al., 1997]

Esferas de influencia

- La construcción del grafo de esferas de influencia es más débil que la construcción de KNN al mismo k.

  Sea $SI^k(v_i) := \{v_j | \rho_k(x_i) + \rho_k(x_j) \geq d(x_i, x_j)\}$ y $NN^k(v_i)$ como se definió antes:

Esferas de influencia

- La construcción del grafo de esferas de influencia es más débil que la construcción de KNN al mismo k.

  Sea $SI^k(v_i) := \{v_j | \rho_k(x_i) + \rho_k(x_j) \geq d(x_i, x_j)\}$ y $NN^k(v_i)$ como se definió antes:
- $NN^k(v_i) \subseteq SI^k(v_i)$:

$$v_j \in NN^k(v_i) \rightarrow d(x_i, x_j) \leq \rho_k(x_i) \leq \rho_k(x_i) + \rho_k(x_j)$$
$$\rightarrow v_j \in SI^k(v_i)$$

Implementación computacional

- Vamos a comparar la implementación computacional de los algoritmos en dimensiones $d = 2, 7, 15$ para varias construcciones y haremos dos ejercicios:

Implementación computacional

- Vamos a comparar la implementación computacional de los algoritmos en dimensiones $d = 2, 7, 15$ para varias construcciones y haremos dos ejercicios:
  - Evaluar cómo se comportan los algoritmos cuando estos saben el número de clusters y están bien calibrados

Implementación computacional

- Vamos a comparar la implementación computacional de los algoritmos en dimensiones $d = 2, 7, 15$ para varias construcciones y haremos dos ejercicios:
  - Evaluar cómo se comportan los algoritmos cuando estos saben el número de clusters y están bien calibrados
  - Evaluar cuántos clusters estimarían si los algoritmos no supiesen el número de clusters.

Implementación computacional

- Vamos a comparar la implementación computacional de los algoritmos en dimensiones $d = 2, 7, 15$ para varias construcciones y haremos dos ejercicios:
  - Evaluar cómo se comportan los algoritmos cuando estos saben el número de clusters y están bien calibrados
  - Evaluar cuántos clusters estimarían si los algoritmos no supiesen el número de clusters.
- Los algoritmos a evaluar serán:

Implementación computacional

- Vamos a comparar la implementación computacional de los algoritmos en dimensiones $d = 2, 7, 15$ para varias construcciones y haremos dos ejercicios:
  - Evaluar cómo se comportan los algoritmos cuando estos saben el número de clusters y están bien calibrados
  - Evaluar cuántos clusters estimarían si los algoritmos no supiesen el número de clusters.
- Los algoritmos a evaluar serán:
  1. K-Medias (Calibración por Calinski-Harabasz)
  2. Clustering Jerárquico, enlace simple completo y ward (Calibración por Calinski-Harabasz)
  3. Clustering espectral con matriz de pesos por KNN (Calibración por autovalores o eigen-gap)
  4. Clustering espectral con matriz de pesos por esferas de influencia (Calibración por autovalores o eigen-gap)
  5. DBSCAN (Calibración por el método del codo)

Implementación computacional- Score de Fowlkes y Mallows

La comparación de la eficacia de los algoritmos se basará en el score de [Fowlkes and Mallows, 1983] que mide qué tan similar son dos particiones:

- Este score toma dos clusterings A_1, A_2 cada uno con k_1, k_2 clusters. Construye $M = (m_{ij})_{i \leq k_1, j \leq k_2}$. con m_{ij} el número de elementos en común del i-ésimo cluster de A_1 con el $j-$ésimo cluster de A_2. La medida de asociación será:

$$B_{k_1,k_2} = \frac{T_{k_1,k_2}}{\sqrt{P_{k_1,k_2} Q_{k_1,k_2}}} \quad B_{k_1,k_2} \in [0,1]$$

Implementación computacional- Score de Fowlkes y Mallows

La comparación de la eficacia de los algoritmos se basará en el score de [Fowlkes and Mallows, 1983] que mide qué tan similar son dos particiones:

- Este score toma dos clusterings A_1, A_2 cada uno con k_1, k_2 clusters. Construye $M = (m_{ij})_{i \leq k_1, j \leq k_2}$. con m_{ij} el número de elementos en común del i-ésimo cluster de A_1 con el $j-$ésimo cluster de A_2. La medida de asociación será:

$$B_{k_1,k_2} = \frac{T_{k_1,k_2}}{\sqrt{P_{k_1,k_2} Q_{k_1,k_2}}} \quad B_{k_1,k_2} \in [0,1]$$

con $T_{k_1,k_2} = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} m_{i,j}^2 - n$, $P_{k_1,k_2} = \sum_{i=1}^{k_1} (\sum_{j=1}^{k_2} m_{i,j})^2 - n$
$Q_{k_1,k_2} = \sum_{j=1}^{k_2} (\sum_{i=1}^{k_1} m_{i,j})^2 - n$

Dimensión 2

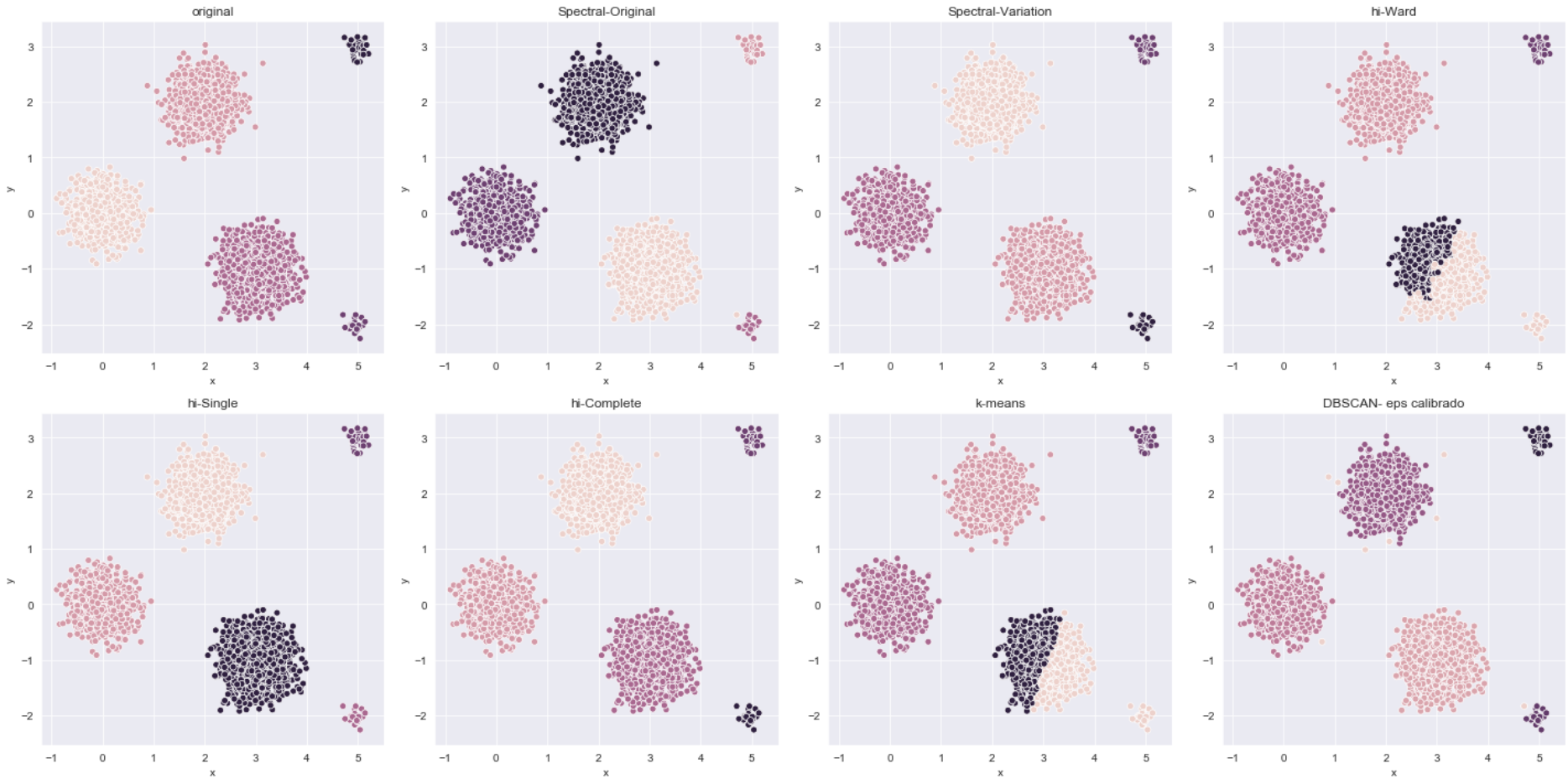


Figura 2: Datos compactos- Dimensión 2

Dimensión 2

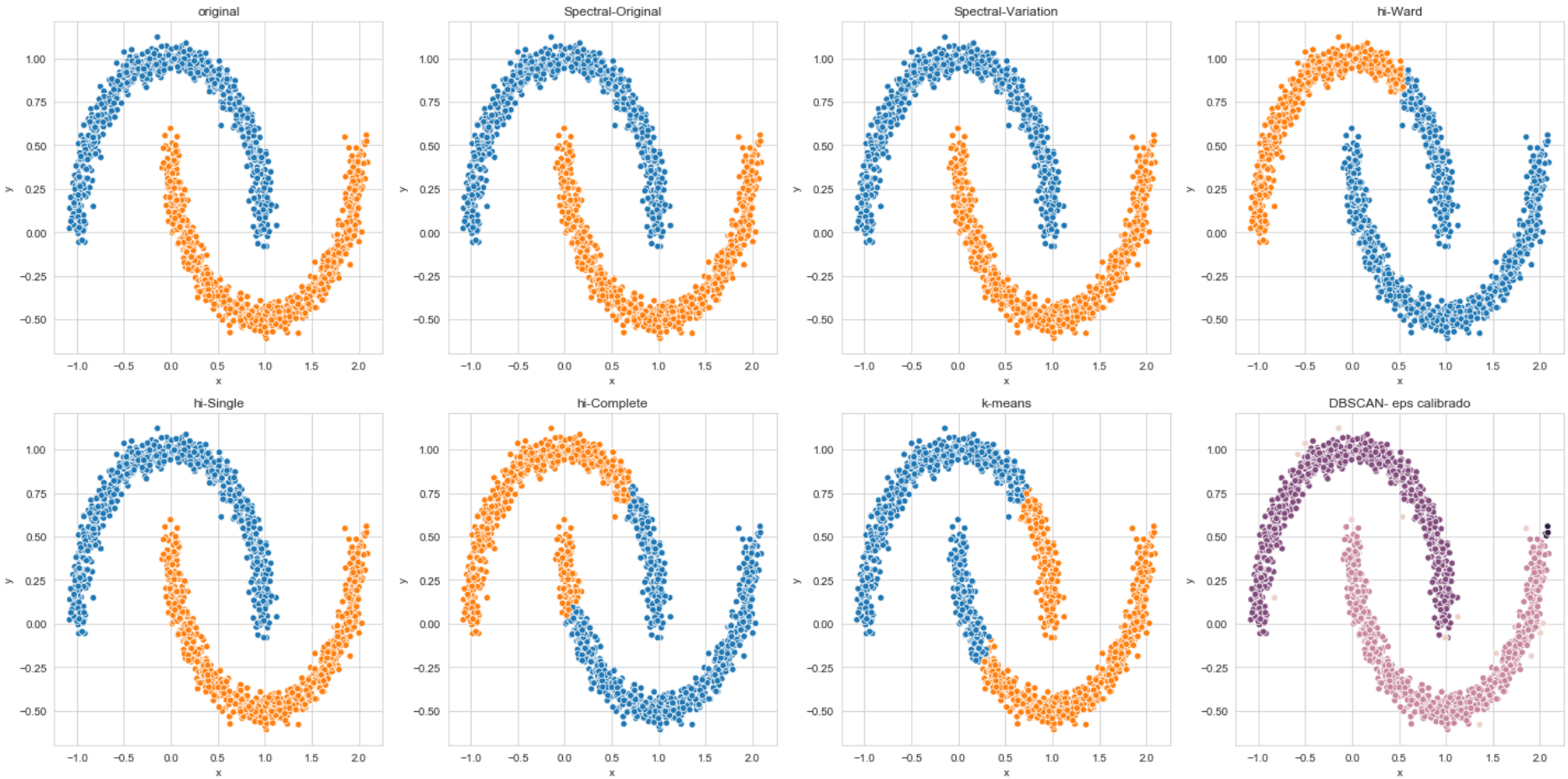


Figura 3: Semilunas- Dimensión 2

Dimensión 2

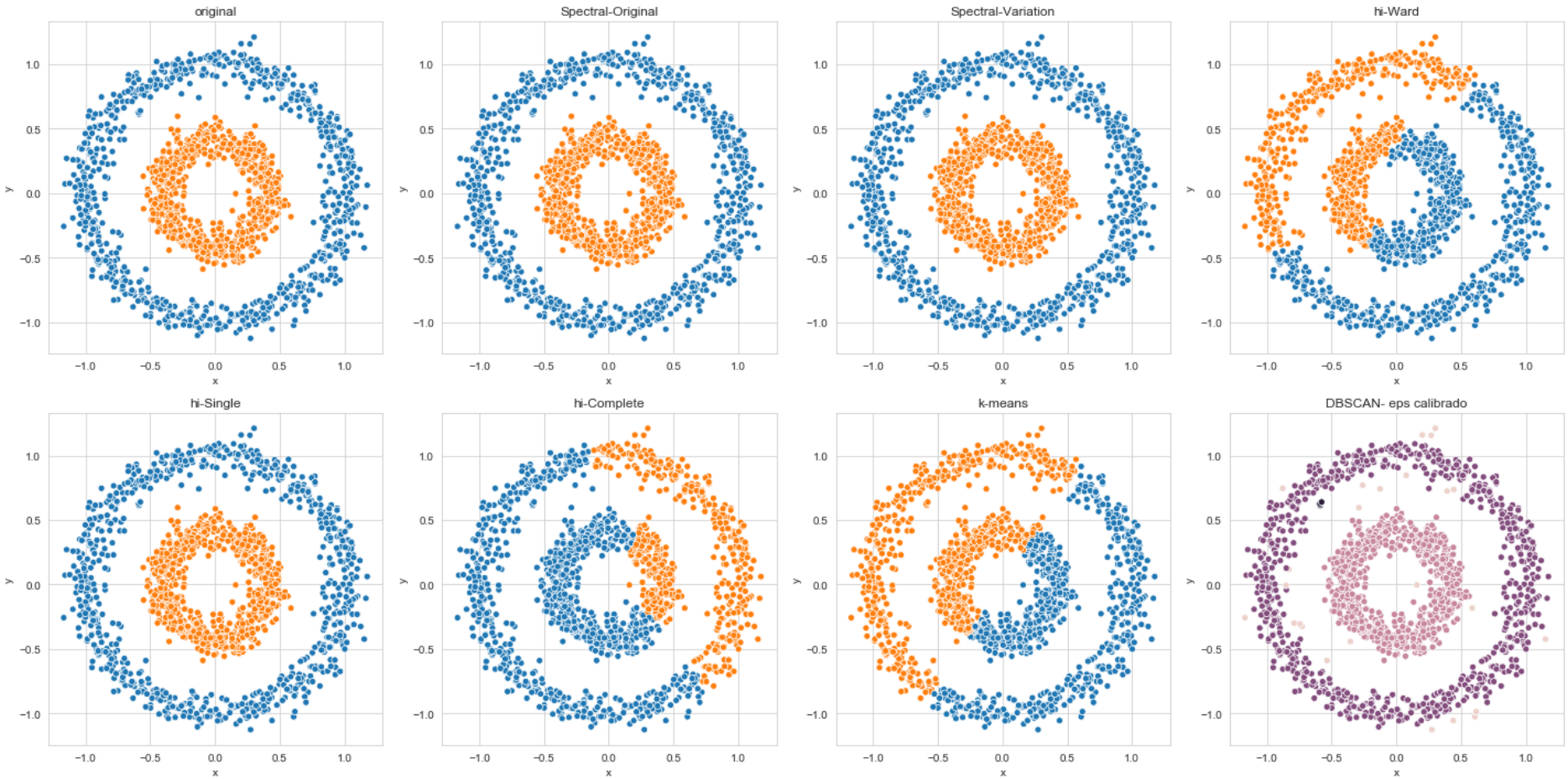


Figura 4: Círculos concéntricos- Dimensión 2

Dimensión 2

Figura 5: Datos con ruido- Dimensión 2

Dimensión 2

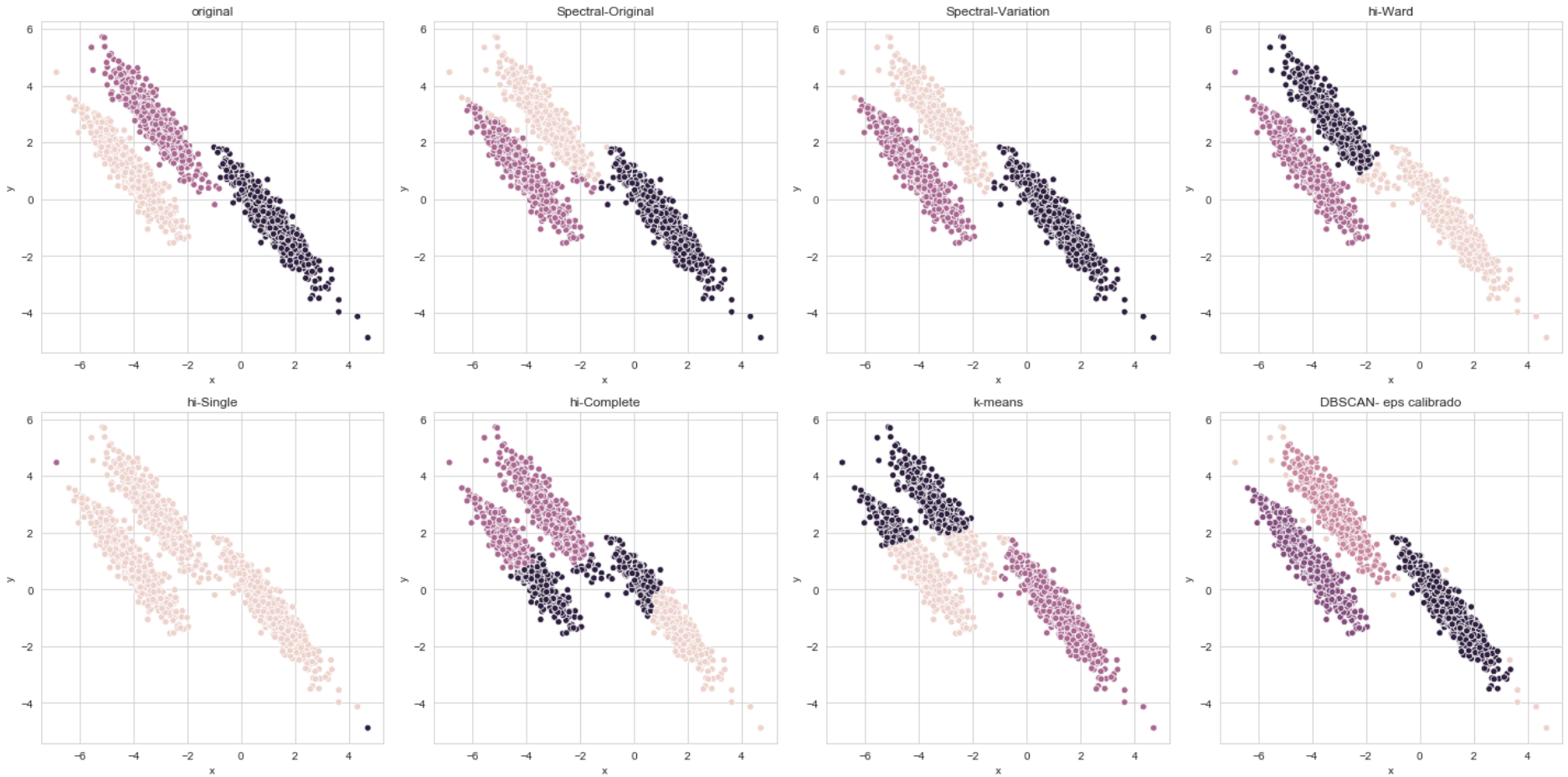


Figura 6: Datos “achatados” - Dimensión 2

Dimensión 2- Fowlkes y Mallows

	Esp-Orig	Esp-Esferas	Hi- Ward	Hi- Simple	Hi- Comp	K-Medias	DBSCAN
Compactos	1.00	1.00	0.85	1.00	1.00	0.86	0.99
Semilunas	1.00	1.00	0.78	1.00	0.74	0.62	0.97
Casquetes	1.00	1.00	0.51	0.71	0.56	0.50	0.98
Ruido	0.97	0.97	0.98	0.57	0.70	0.89	0.56
Achatados	0.99	0.99	0.97	0.57	0.62	0.76	0.99

Cuadro 1: Tabla de puntajes de Fowlkes y Mallows para las diferentes construcciones de datos y los algoritmos de clustering evaluados en su mejor configuración, dimensión 2.

Dimensión 2 - Estimación del número de clústers según cada algoritmo

	Número de clusters estimados						
	Esp-Orig	Esp-Esferas	Hi- Ward	Hi- Simple	Hi- Comp	K-Medias	DBSCAN
Compactos[5]	5	5	4	4	4	4	5
Semilunas[2]	2	2	8	2	8	8	2
Casquetes[2]	2	2	8	2	6	8	2
Ruido[3]	3	3	3	3	6	3	1
Achatados[3]	4	4	2	4	8	2	3

Cuadro 2: Tabla de calibración del número de clusters y comparación con el número verdadero de clusters para los diferentes métodos a evaluar en dimensión 2.

Dimensión alta - Nubes de datos a considerar

Sea $\underline{X} = (X_1, \cdots, X_n)' \sim \mathcal{N}(0, I_{d\times d})$ y sus propiedades relevantes:

- Invarianza ante transformaciones ortogonales.
- Descomposición de varianza con base en el espectro de transformaciones generales.

Dimensión alta - Nubes de datos a considerar

Sea $\underline{X} = (X_1, \cdots, X_n)' \sim \mathcal{N}(0, I_{d\times d})$ y sus propiedades relevantes:

- Invarianza ante transformaciones ortogonales.
- Descomposición de varianza con base en el espectro de transformaciones generales.

Se construyó una transformación $\underline{Y} = R\underline{X}$ tal que $RR' = Cov(Y) = C$ y C fuese real valuada, definida positiva y simétrica.

Dimensión alta - Nubes de datos a considerar

Sea $\underline{X} = (X_1, \cdots, X_n)' \sim \mathcal{N}(0, I_{d\times d})$ y sus propiedades relevantes:

- Invarianza ante transformaciones ortogonales.
- Descomposición de varianza con base en el espectro de transformaciones generales.

Se construyó una transformación $\underline{Y} = R\underline{X}$ tal que $RR' = Cov(Y) = C$ y C fuese real valuada, definida positiva y simétrica. Se construyó C de tal forma que fuese una transformación que inducía gran varianza en la dirección del vector $\mathbb{1}$ y poca varianza en las demás direcciones (de la base de la descomposición espectral).

Dimensión alta - Nubes de datos a considerar

Sea $\underline{X} = (X_1, \cdots, X_n)' \sim \mathcal{N}(0, I_{d\times d})$ y sus propiedades relevantes:

- Invarianza ante transformaciones ortogonales.
- Descomposición de varianza con base en el espectro de transformaciones generales.

Se construyó una transformación $\underline{Y} = R\underline{X}$ tal que $RR' = Cov(Y) = C$ y C fuese real valuada, definida positiva y simétrica. Se construyó C de tal forma que fuese una transformación que inducía gran varianza en la dirección del vector $\mathbb{1}$ y poca varianza en las demás direcciones (de la base de la descomposición espectral). $C = 10\mathbb{1}_{d\times d} + I_{d\times d}$ donde la matriz

$$\mathbb{1}_{d\times d} = \begin{pmatrix} 1 & 1 & \cdots & \cdots & 1 \\ 1 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & 1 \\ 1 & \cdots & \cdots & 1 & 1 \end{pmatrix}$$

Dimensión alta - Nubes de datos a considerar

- Con $\underline{Y} = R\underline{X}$ ya construido la nube de datos a utilizar será la nube $\underline{\tilde{Y}}$ a partir de $\underline{Y}$ haciendo una partición aleatoria del número de datos en un número de clusters dado (K) $K \leq d$:

Dimensión alta - Nubes de datos a considerar

- Con $\underline{Y} = R\underline{X}$ ya construido la nube de datos a utilizar será la nube $\underline{\tilde{Y}}$ a partir de $\underline{Y}$ haciendo una partición aleatoria del número de datos en un número de clusters dado (K) $K \leq d$:

$$\tilde{Y}_i^j = Y_i + \delta\mu_j \qquad j \in \{1, \cdots, K-1\}$$

Dimensión alta - Nubes de datos a considerar

- Con $\underline{Y} = R\underline{X}$ ya construido la nube de datos a utilizar será la nube $\underline{\tilde{Y}}$ a partir de $\underline{Y}$ haciendo una partición aleatoria del número de datos en un número de clusters dado (K) $K \leq d$:

$$\tilde{Y}_i^j = Y_i + \delta\mu_j \qquad j \in \{1, \cdots, K-1\}$$

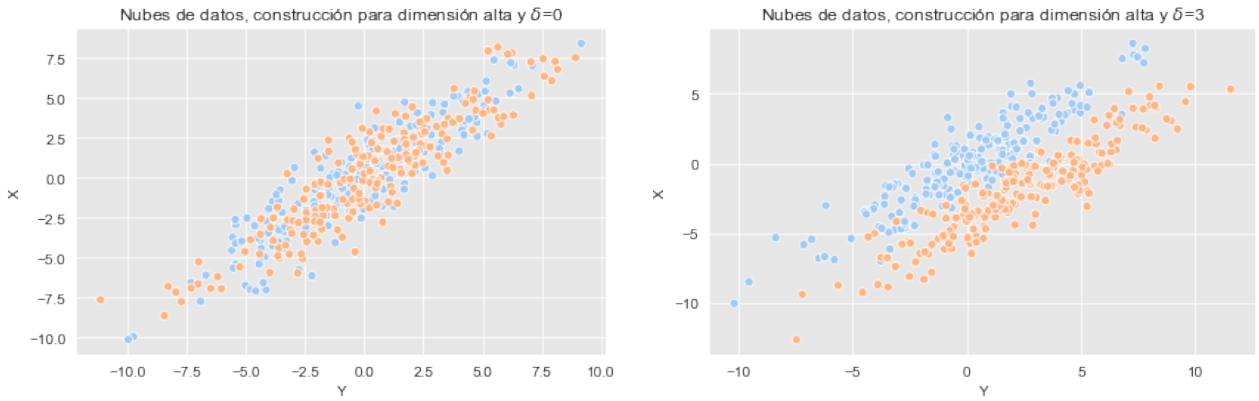
- Donde $\delta > 0$ y μ_j hace parte de los vectores propios obtenidos de la descomposición espectral de C y no es $\bar{\mathbb{1}}$.

Dimensión alta - Nubes de datos a considerar

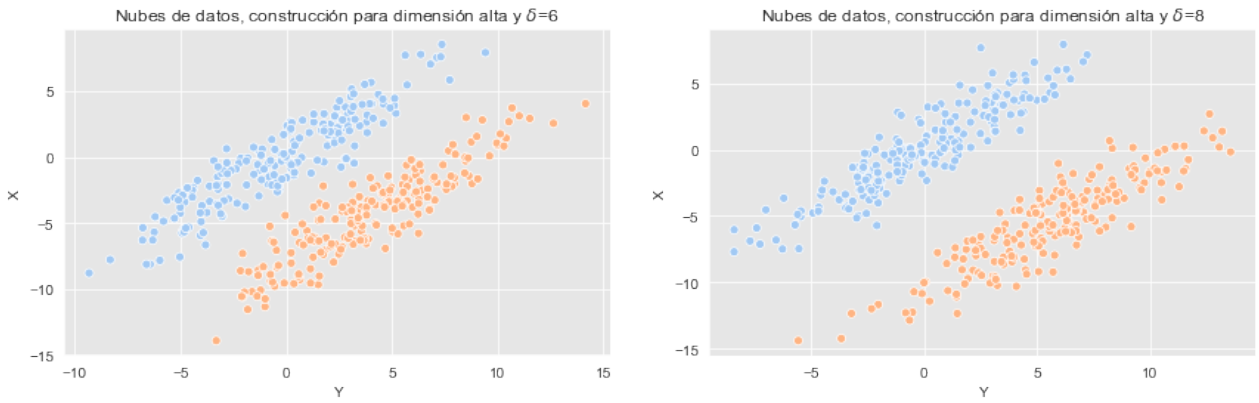
- Con $\underline{Y} = R\underline{X}$ ya construido la nube de datos a utilizar será la nube $\underline{\tilde{Y}}$ a partir de $\underline{Y}$ haciendo una partición aleatoria del número de datos en un número de clusters dado (K) $K \leq d$:

$$\tilde{Y}_i^j = Y_i + \delta\mu_j \qquad j \in \{1, \cdots, K-1\}$$

- Donde $\delta > 0$ y μ_j hace parte de los vectores propios obtenidos de la descomposición espectral de C y no es $\bar{\mathbb{1}}$.
- La idea detrás de esta nueva transformación es dividir la muestra completa en K submuestras de $\frac{n}{k}$ datos tales que estas se vayan alejando en direcciones ortogonales a la dirección de mayor varianza (que es $\bar{\mathbb{1}}$), por un factor de distancia δ.



(a) $\delta = 0$ (b) $\delta = 3$



(c) $\delta = 6$ (d) $\delta = 8$

Figura 7: Implementación de la construcción de nubes de datos en dimensión arbitraria para $d = 2$ y diferentes valores de δ.

Dimensión 7- Fowlkes y Mallows

		Esp-Orig	Esp-Esferas	Hi- Ward	Hi- Simple	Hi- Comp	K-Medias	DBSCAN
δ	5	0.54	0.37	0.25	0.44	0.25	0.27	0.44
	6	0.72	0.69	0.44	0.44	0.27	0.31	0.43
	7	0.86	0.73	0.44	0.44	0.27	0.41	0.65
	8	1.00	1.00	0.36	0.45	0.31	0.40	0.80
	9	1.00	1.00	0.52	0.54	0.34	0.42	0.96

Cuadro 3: Tabla de puntajes de Fowlkes y Mallows para algoritmos evaluados en dimensión 7 y diferentes δ.

Dimensión 7- Estimación del número de clústers según cada algoritmo

		Esp-Orig	Esp-Esferas	Hi- Ward	Hi- Simple	Hi- Comp	K-Medias	DBSCAN
δ	5	4	4	2	8	2	2	1
	6	7	7	2	2	2	2	2
	7	6	6	2	2	2	2	2
	8	5	5	2	3	2	2	3
	9	5	5	2	4	2	2	5

Cuadro 4: Tabla de calibración del número de clusters y comparación con el número verdadero de clusters para los diferentes métodos a evaluar en dimensión 7.

Dimensión 15- Fowlkes y Mallows

		Esp-Orig	Esp-Esferas	Hi- Ward	Hi- Simple	Hi- Comp	K-Medias	DBSCAN
δ	5	0.43	0.31	0.23	0.44	0.27	0.21	0.42
	6	0.64	0.34	0.25	0.44	0.25	0.21	0.41
	7	0.51	0.36	0.28	0.44	0.27	0.22	0.41
	8	0.99	0.55	0.30	0.44	0.28	0.27	0.43
	9	1.00	0.75	0.33	0.44	0.28	0.30	0.75

Cuadro 5: Tabla de puntajes de Fowlkes y Mallows para algoritmos evaluados en dimensión 15 y diferentes δ.

Dimensión 15- Estimación del número de clusters según cada algoritmo

		Esp-Orig	Esp-Esferas	Hi- Ward	Hi- Simple	Hi- Comp	K-Medias	DBSCAN
δ	5	2	2	3	2	4	2	1
	6	2	2	2	2	2	2	1
	7	5	5	2	8	3	2	1
	8	3	3	2	7	2	2	1
	9	5	5	2	2	3	2	4

Cuadro 6: Tabla de calibración del número de clusters y comparación con el número verdadero de clusters para los diferentes métodos a evaluar en dimensión 15.

Resultados Generales - Conclusiones

- En los procedimientos computacionales se puede ver que el algoritmo de clustering espectral es consistentemente robusto en la clasificación de clústers y en la detección correcta del número de clusters.

Resultados Generales - Conclusiones

- En los procedimientos computacionales se puede ver que el algoritmo de clustering espectral es consistentemente robusto en la clasificación de clústers y en la detección correcta del número de clusters.
- El algoritmo de DBSCAN es aquel que mejor se comporta ***después*** del algoritmo de clustering espectral.

Resultados Generales - Conclusiones

- En los procedimientos computacionales se puede ver que el algoritmo de clustering espectral es consistentemente robusto en la clasificación de clústers y en la detección correcta del número de clusters.
- El algoritmo de DBSCAN es aquel que mejor se comporta ***después*** del algoritmo de clustering espectral.
- Mediante las diversas construcciones salieron a la luz las desventajas de los demás algoritmos.

Resultados Generales - Conclusiones

- En los procedimientos computacionales se puede ver que el algoritmo de clustering espectral es consistentemente robusto en la clasificación de clústers y en la detección correcta del número de clusters.
- El algoritmo de DBSCAN es aquel que mejor se comporta ***después*** del algoritmo de clustering espectral.
- Mediante las diversas construcciones salieron a la luz las desventajas de los demás algoritmos.
- A medida que la dimensión aumentó, hubo menor precisión por parte de los algoritmos. Se comprobó que clústers más separados benefician al algoritmo de clustering espectral

Conclusiones

- La variación de esferas de influcencia tuvo resultados similares hasta dimensión 7, a partir de dimensión 15 ésta variación tuvo un peor desempeño que el de KNN.

Conclusiones

- La variación de esferas de influcencia tuvo resultados similares hasta dimensión 7, a partir de dimensión 15 ésta variación tuvo un peor desempeño que el de KNN.
  - Esto *puede* ser explicado por el hecho de que la variación construye gráfos con restricciones más débiles entre datos que KNN.

Conclusiones

- La variación de esferas de influcencia tuvo resultados similares hasta dimensión 7, a partir de dimensión 15 ésta variación tuvo un peor desempeño que el de KNN.
  - Esto *puede* ser explicado por el hecho de que la variación construye gráfos con restricciones más débiles entre datos que KNN.
  - Nos lleva a conjeturar sí puede ser el caso que a dimensiones más altas, construcciones más fuertes mejoren el desempeño a mayor distancia. Un ejemplo podría ser el 1-MST [Prim, 1957].

Recomendaciones

- Considerar más variaciones en la construcción de la matriz de pesos a la hora de implementar el algoritmo de clustering espectral (Como el MST, o el K-MST)
- Implementar otros mecanismos de calibración de clustering.
- Realizar un análisis de complejidad computacional.
- Evaluar diferentes construcciones en dimensiones altas, como híper casquetes de esferas concéntricos, híper elipsoides o híper esferas.

References I


Alashwal, H., El Halaby, M., Crouse, J. J., Abdalla, A., and Moustafa, A. A. (2019).
The application of unsupervised clustering methods to alzheimer's disease.
Frontiers in Computational Neuroscience, 13:31.


Brito, M., Chávez, E., Quiroz, A., and Yukich, J. (1997).
Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection.
Statistics Probability Letters, 35(1):33–42.


Caliński, T. and Harabasz, J. (1974).
A dendrite method for cluster analysis.
Communications in Statistics-Simulation and Computation, 3(1):1–27.

References II


Corredor, J. S. and Quiroz, A. J. (2020).
Shannon's entropy of partitions determined by hierarchical clustering trees in asymmetry and dimension identification.
Communications in Statistics - Simulation and Computation, 0(0):1–13.


Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996).
A density-based algorithm for discovering clusters in large spatial databases with noise.
pages 226–231. AAAI Press.


Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011).
Cluster Analysis, 5th Edition.
John Wiley & Sons.

References III


Fowlkes, E. B. and Mallows, C. L. (1983).
A method for comparing two hierarchical clusterings.
Journal of the American Statistical Association, 78(383):553–569.


Guattery, S. and Miller, G. L. (1998).
On the quality of spectral separators.
SIAM Journal on Matrix Analysis and Applications, 19:701–719.


Hartigan, J. A. and Wong, M. A. (1979).
Algorithm AS 136: A K-Means clustering algorithm.
Applied Statistics, 28(1):100–108.


Lloyd, S. P. (1982).
Least squares quantization in pcm.
IEEE Transactions on Information Theory, 28:129–137.

References IV


Milligan, G. W. and Cooper, M. C. (1985).
An examination of procedures for determining the number of clusters in a data set.
Psychometrika, 50(2):159–179.


Mohd ariff, N., Abu Bakar, M. A., and Rahmad, M. (2018).
Comparative study of document clustering algorithms.
International Journal of Engineering and Technology(UAE), 7:246–251.


Mojena, R. (1977).
Hierarchical grouping methods and stopping rules: An evaluation.
The Comp. J., 20:359–363.

References V


Nielsen, F. (2016).
Hierarchical clustering.
In *Introduction to HPC with MPI for Data Science*, pages 195–211. Springer.


Papalexakis, E. E. (2018).
Unsupervised content-based identification of fake news articles with tensor decomposition ensembles.


Pavithra, A. and Dhanaraj, M. S. (2019).
Prediction accuracy on academic performance of students using different data mining algorithms with influencing factors.

References VI


Prim, R. C. (1957).
Shortest connection networks and some generalizations.
The Bell System Technical Journal, 36(6):1389–1401.


Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017).
Dbscan revisited, revisited: Why and how you should (still) use dbscan.
ACM Trans. Database Syst., 42(3).


Selim, S. Z. and Ismail, M. A. (1984).
K-means-type algorithms: A generalized convergence theorem and characterization of local optimality.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(1):81–87.

References VII

Sharma, A. and Rastogi, V. (2014).
Spam filtering using k mean clustering with local feature selection classifier.
International Journal of Computer Applications, 108:35–39.

Von Luxburg, U. (2007).
A tutorial on spectral clustering.
Statistics and computing, 17(4):395–416.