Introducción a la Ciencia de Datos

Maestría en Probabilidad y Estadística

Dr. Marco Antonio Aquino López

Centro de Investigación en Matemáticas

Agosto-Diciembre 2025

Motivación

- En muchos contextos no existen etiquetas ni categorías predefinidas.
- Queremos descubrir estructuras, patrones o representaciones ocultas en los datos.
- Este es el dominio del **Aprendizaje No Supervisado (Unsupervised Learning)**.

Ejemplos:

- Agrupar especies biológicas por similitud genética.
- Detectar temas latentes en un corpus de textos.
- Reducir la dimensionalidad de imágenes o señales para visualización.

¿Qué es el Aprendizaje No Supervisado?

Definición

Conjunto de técnicas donde buscamos **estructuras latentes** en datos **sin etiquetas** ni respuestas conocidas.

¿Qué es el Aprendizaje No Supervisado?

Definición

Conjunto de técnicas donde buscamos **estructuras latentes** en datos **sin etiquetas** ni respuestas conocidas.

- **Problema**: Dado un conjunto de datos $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ con $\mathbf{x}_i \in \mathbb{R}^p$

¿Qué es el Aprendizaje No Supervisado?

Definición

Conjunto de técnicas donde buscamos **estructuras latentes** en datos **sin etiquetas** ni respuestas conocidas.

- **Problema**: Dado un conjunto de datos $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ con $\mathbf{x}_i \in \mathbb{R}^p$
- **Objetivo**: Encontrar patrones, agrupaciones o representaciones subyacentes

¿Qué es el Aprendizaje No Supervisado?

Definición

Conjunto de técnicas donde buscamos **estructuras latentes** en datos **sin etiquetas** ni respuestas conocidas.

- **Problema**: Dado un conjunto de datos $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ con $\mathbf{x}_i \in \mathbb{R}^p$
- **Objetivo**: Encontrar patrones, agrupaciones o representaciones subyacentes
- **Contraste**: En aprendizaje supervisado tenemos $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$

Supervisado vs. No Supervisado

Aprendizaje Supervisado

- Datos con etiquetas (x_i, y_i).
- Objetivo: predecir y a partir de x.
- Ejemplos: regresión, clasificación.

Aprendizaje No Supervisado

- Solo observamos $\{x_i\}$.
- Objetivo: encontrar estructura, agrupamientos, proyecciones.
- Ejemplos: clustering, PCA, ICA, autoencoders.

Meta

Extraer regularidades y dependencias intrínsecas en los datos.

Perspectiva Estadística

- Modelamos la **distribución** de los datos.

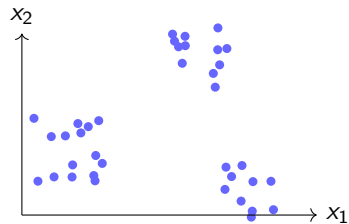


Figura: Agrupamiento natural en los datos

Perspectiva Estadística

- Modelamos la **distribución** de los datos.
- Buscamos **estructura de dependencia**.

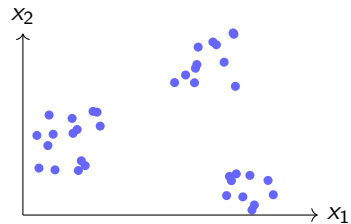


Figura: Agrupamiento natural en los datos

Perspectiva Estadística

- Modelamos la **distribución** de los datos.
- Buscamos **estructura de dependencia**.
- **Reducción de dimensionalidad**.

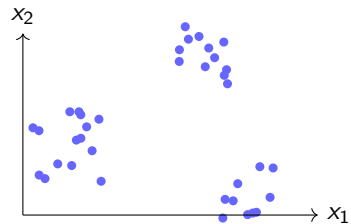


Figura: Agrupamiento natural en los datos

Perspectiva Estadística

- Modelamos la **distribución** de los datos.
- Buscamos **estructura de dependencia**.
- **Reducción de dimensionalidad**.
- **Detección de anomalías**.

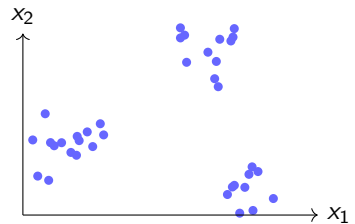


Figura: Agrupamiento natural en los datos

Paradigmas Principales

1. **Agrupamiento (Clustering)**: descubrir grupos naturales o similitudes.
2. **Reducción de Dimensionalidad**: encontrar representaciones compactas.
3. **Proyecciones y Visualización**: preservar distancias o estructuras.

Estos enfoques buscan una *estructura latente* que explique los datos observados.

Mapa de Temas

- **1. Agrupamiento:**
  - k-means y variantes (k-medians, k-medoids)
  - Agrupamiento jerárquico (agglomerativo, divisivo)
  - Clustering espectral y de densidad (DBSCAN)
  - Modelos de mezcla y EM (Gaussian Mixture Models)
  - Clustering difuso (Fuzzy c-means)
- **2. Reducción de Dimensionalidad:**
  - PCA, SVD, NMF, ICA
  - Métodos no lineales (Isomap, LLE, t-SNE, UMAP)

Aplicaciones en Estadística

- **Análisis exploratorio de datos**

Aplicaciones en Estadística

- **Análisis exploratorio de datos**
- **Preprocesamiento** para modelos supervisados

Aplicaciones en Estadística

- **Análisis exploratorio de datos**
- **Preprocesamiento** para modelos supervisados
- **Detección de valores atípicos**

Aplicaciones en Estadística

- **Análisis exploratorio de datos**
- **Preprocesamiento** para modelos supervisados
- **Detección de valores atípicos**
- **Segmentación de mercados**

Desafíos Estadísticos

Consideraciones Importantes

- Selección del número de clusters
- Validación de resultados
- Sensibilidad a inicialización
- Interpretabilidad
- Curse of dimensionality

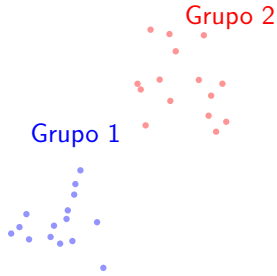
Agrupamiento

Agrupación

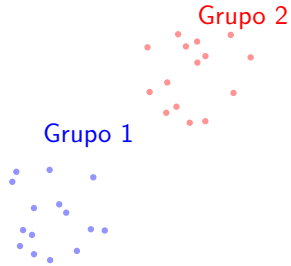
Definición según la Real Academia Española:

- f. Acción y efecto de agrupar.
  Sin.: asociación, reunión, agrupamiento, concentración, junta.
- f. Mil. Unidad homogénea, de importancia semejante a la del regimiento.

¿Qué significa agrupar?



¿Qué significa agrupar?



Objetivo general

Encontrar subconjuntos $\{C_1, \ldots, C_K\}$ tales que:

$$C_i \cap C_j = \emptyset, \quad \bigcup_i C_i = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\},$$

maximizando la similitud dentro del grupo y minimizando la similitud entre grupos.

Agrupamiento

- Agrupar consumidores con hábitos similares.
- Identificar regiones climáticas homogéneas.

Idea central

Objetos “cercanos” según alguna métrica deben pertenecer al mismo grupo.

Tipos de agrupamiento

Particional

- Asigna cada punto a un grupo.
- Ejemplo: k-means.

Jerárquico

- Construye una jerarquía de grupos.
- Dendrogramas, clustering aglomerativo o divisivo.

Tipos de agrupamiento

Particional

- Asigna cada punto a un grupo.
- Ejemplo: k-means.

Jerárquico

- Construye una jerarquía de grupos.
- Dendrogramas, clustering aglomerativo o divisivo.

Otras variantes: difuso (fuzzy), basado en densidad (DBSCAN), en grafos, espectral, etc.

Criterios de calidad

- **Cohesión**: qué tan similares son los elementos dentro del grupo.
- **Separación**: qué tan distintos son los grupos entre sí.

Ejemplo de función objetivo

Minimizar la suma de distancias cuadradas dentro de los grupos:

$$\text{SSE} = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

Hacia el método k-means

- Buscamos una partición de los datos en K grupos.
- Cada grupo se representa por un centroide $\boldsymbol{\mu}_k$.
- Queremos minimizar la variabilidad dentro de cada grupo.

Problema de optimización

$$\min_{\{C_k, \boldsymbol{\mu}_k\}} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

Hacia el método k-means

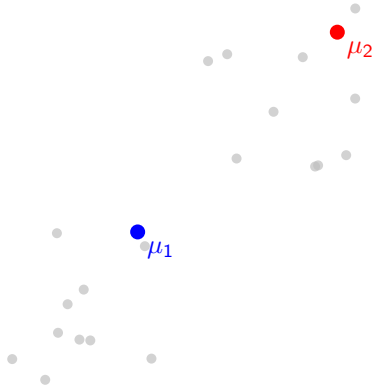
- Buscamos una partición de los datos en K grupos.
- Cada grupo se representa por un centroide $\boldsymbol{\mu}_k$.
- Queremos minimizar la variabilidad dentro de cada grupo.

Problema de optimización

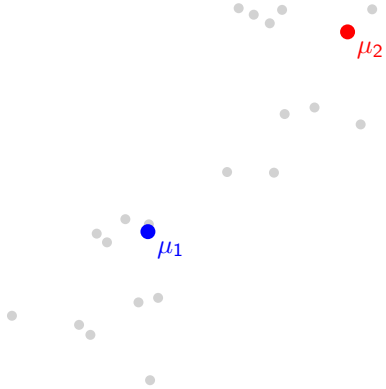
$$\min_{\{C_k, \boldsymbol{\mu}_k\}} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

Idea: alternar entre asignar puntos a su centroide más cercano y recalcular los centroides.

Intuición geométrica



Intuición geométrica



Idea

El algoritmo k-means busca las posiciones de los centroides que minimizan la distancia total a los puntos del grupo.

Formalización matemática del problema

Sea $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ y $K \in \mathbb{N}$ fijo.
Definimos el problema de optimización:

$$\min_{\{C_k\}_{k=1}^K, \{\boldsymbol{\mu}_k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2$$

sujeto a:

$$C_i \cap C_j = \emptyset \quad \forall i \neq j$$

$$\bigcup_{k=1}^{K} C_k = \mathcal{X}$$

$$\boldsymbol{\mu}_k \in \mathbb{R}^d \quad \forall k = 1, \ldots, K$$

Complejidad computacional

K-means resuelve una relajación continua mediante "coordinate descent".

Función de distorsión

El problema general de agrupamiento puede expresarse como la minimización de una función de costo:

$$J(\mathcal{C}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \ell(\mathbf{x}_i, \boldsymbol{\mu}_k),$$

donde ℓ es una **función de pérdida** que mide la disimilitud entre $\mathbf{x}_i$ y su prototipo.

Función de distorsión

El problema general de agrupamiento puede expresarse como la minimización de una función de costo:

$$J(\mathcal{C}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \ell(\mathbf{x}_i, \boldsymbol{\mu}_k),$$

donde ℓ es una **función de pérdida** que mide la disimilitud entre $\mathbf{x}_i$ y su prototipo.

Ejemplo común

Si $\ell(\mathbf{x}_i, \boldsymbol{\mu}_k) = \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$ (distancia euclidiana cuadrática), entonces se busca minimizar la **suma de errores cuadráticos (SSE)**.

Reformulación como problema de asignación

Sea $z_i \in \{1, \ldots, K\}$ la etiqueta de cluster para $\mathbf{x}_i$.

Reformulación como problema de asignación

Sea $z_i \in \{1, \ldots, K\}$ la etiqueta de cluster para $\mathbf{x}_i$.

$$\min_{\{z_i\},\{\boldsymbol{\mu}_k\}} \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu}_{z_i}\|^2$$

Reformulación como problema de asignación

Sea $z_i \in \{1, \ldots, K\}$ la etiqueta de cluster para $\mathbf{x}_i$.

$$\min_{\{z_i\},\{\boldsymbol{\mu}_k\}} \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu}_{z_i}\|^2$$

Podemos escribir una función indicadora:

$$r_{ik} = \begin{cases} 1, & \text{si } z_i = k, \\ 0, & \text{en otro caso,} \end{cases}$$

y reescribir el costo como:

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2,$$

sujeto a $\sum_{k=1}^{K} r_{ik} = 1$ para todo i.

Solución alternante: intuición

- Fijando $\{\boldsymbol{\mu}_k\}$, la asignación óptima r_{ik} es:

$$r_{ik} = \begin{cases} 1, & \text{si } k = \arg\min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2, \\ 0, & \text{en otro caso.} \end{cases}$$

Solución alternante: intuición

- Fijando $\{\boldsymbol{\mu}_k\}$, la asignación óptima r_{ik} es:

$$r_{ik} = \begin{cases} 1, & \text{si } k = \arg\min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2, \\ 0, & \text{en otro caso.} \end{cases}$$

- Fijando $\{r_{ik}\}$, la actualización óptima de los centroides es:

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}}.$$

Solución alternante: intuición

- Fijando $\{\boldsymbol{\mu}_k\}$, la asignación óptima r_{ik} es:

$$r_{ik} = \begin{cases} 1, & \text{si } k = \arg\min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2, \\ 0, & \text{en otro caso.} \end{cases}$$

- Fijando $\{r_{ik}\}$, la actualización óptima de los centroides es:

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}}.$$

Comentario

Este principio de optimización alternada lleva naturalmente al algoritmo k-means, que garantiza una disminución monótona del costo J en cada iteración.

Algoritmo k-means: idea general

Objetivo recordatorio

Minimizar

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2, \quad \text{sujeto a } \sum_{k=1}^{K} r_{ik} = 1.$$

Algoritmo k-means: idea general

Objetivo recordatorio

Minimizar

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2, \quad \text{sujeto a} \quad \sum_{k=1}^{K} r_{ik} = 1.$$

- Este problema no es convexo en $(r_{ik}, \boldsymbol{\mu}_k)$ simultáneamente.
- Pero sí lo es en cada conjunto de variables por separado.
- Por tanto, se usa una **optimización alternada**:
  1. Asignar cada punto al centroide más cercano.
  2. Recalcular los centroides como promedio de los puntos asignados.

Algoritmo k-means

Algorithm 1: k-means por optimización alternada de asignaciones y centroides

Entrada: Datos $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, número de clusters K.

Salida: Centroides $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ y asignaciones $z_1, \ldots, z_n$.

Inicializar aleatoriamente $\boldsymbol{\mu}_1^{(0)}, \ldots, \boldsymbol{\mu}_K^{(0)}$

$t \leftarrow 0$

repeat

  `// Asignación`

  for $i \leftarrow 1$ **to** n **do**

    $z_i^{(t)} \leftarrow \arg\min_{k \in \{1,\ldots,K\}} \|\mathbf{x}_i - \boldsymbol{\mu}_k^{(t)}\|^2$

  `// Actualización de centroides`

  for $k \in \{1, \ldots, K\}$ **do**

    $n_k \leftarrow \left|\{i : z_i^{(t)} = k\}\right|$

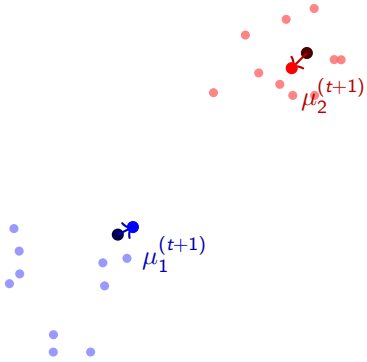
    if $n_k > 0$ **then**

      $\boldsymbol{\mu}_k^{(t+1)} \leftarrow \frac{1}{n_k} \sum_{i : z_i^{(t)} = k} \mathbf{x}_i$

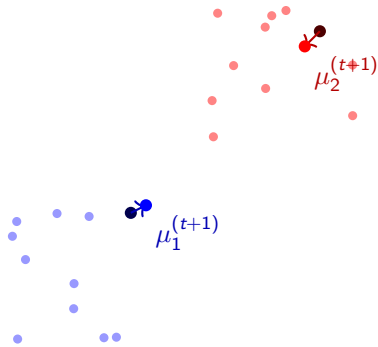
  $t \leftarrow t + 1$

until *las asignaciones no cambian* ***o*** J *converge*

Intuición geométrica de las iteraciones



Intuición geométrica de las iteraciones



Visualmente

Los centroides se desplazan hacia el promedio de sus puntos asignados. El proceso converge cuando los centroides dejan de moverse.

Propiedades y convergencia

- Cada iteración reduce (o mantiene) el valor de J.

$$J^{(t+1)} \leq J^{(t)}.$$

- El algoritmo converge en un número finito de pasos porque existen sólo un número finito de particiones posibles.

Propiedades y convergencia

- Cada iteración reduce (o mantiene) el valor de J.

$$J^{(t+1)} \leq J^{(t)}.$$

- El algoritmo converge en un número finito de pasos porque existen sólo un número finito de particiones posibles.
- Sin embargo, la convergencia es hacia un **mínimo local**.
  - Depende fuertemente de la inicialización.
  - Suele ejecutarse varias veces con distintas semillas.

Supuestos (implícitos) de k-means

- **Clusters aproximadamente esféricos** y de tamaño/densidad similares.
- **Métrica euclidiana** relevante para la noción de "cercanía".
- **Escalas comparables** entre variables (recomendado estandarizar).
- Los centroides μ_k son **prototipos promedio** (minimizan SSE).

Regla práctica

Si las variables están en unidades muy distintas o hay colas pesadas/outliers, estandariza, transforma (log/Box–Cox) o considera métodos robustos/alternativos.

Cuándo *no* usar k-means

- Clusters no convexos (formas lunares, anillos).
- Diferencias fuertes de tamaño/densidad entre grupos.
- Variables categóricas puras (ver k-modes/k-prototypes).
- Presencia de **muchos** outliers (k-means es sensible).

Señal de alerta

SSE baja pero **etiquetas no interpretables** $\Rightarrow$ revisa métrica, escala y K.

Inicialización importa

- K-means puede caer en **mínimos locales**; repetir con varias semillas.
- **k-means++**: elige centros iniciales espaciados (mejora estabilidad y SSE esperado).
- Reinicios múltiples + elegir la solución con menor SSE validada.

Tip de práctica

Usa $10-50$ reinicios con k-means++ y criterio de parada por ΔJ pequeño.

Inicialización k-means++

1. Elegir el primer centroide $\boldsymbol{\mu}_1$ al azar entre los datos.
2. Para cada punto $\mathbf{x}_i$, calcular su distancia al centroide más cercano:

$$D_i = \min_{j<t} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|$$

3. Seleccionar el siguiente centroide $\boldsymbol{\mu}_t$ con probabilidad proporcional a la distancia cuadrada:

$$P(\mathbf{x}_i) = \frac{D_i^2}{\sum_{j=1}^{n} D_j^2}$$

4. Repetir hasta obtener K centroides iniciales.
5. Ejecutar el algoritmo k-means estándar con esos centroides.

Inicialización k-means++

1. Elegir el primer centroide $\boldsymbol{\mu}_1$ al azar entre los datos.
2. Para cada punto $\mathbf{x}_i$, calcular su distancia al centroide más cercano:

$$D_i = \min_{j<t} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|$$

3. Seleccionar el siguiente centroide $\boldsymbol{\mu}_t$ con probabilidad proporcional a la distancia cuadrada:

$$P(\mathbf{x}_i) = \frac{D_i^2}{\sum_{j=1}^{n} D_j^2}$$

4. Repetir hasta obtener K centroides iniciales.
5. Ejecutar el algoritmo k-means estándar con esos centroides.

Intuición y ventajas

- Los puntos más alejados de los centroides actuales tienen mayor probabilidad de ser elegidos.
- Así, los centroides iniciales quedan bien espaciados, mejorando estabilidad y reduciendo el SSE esperado.

Mini-batch k-means (gran escala)

- Procesa **lotes pequeños** de datos: actualiza centroides en línea.
- Acelera en n grande con costo de mayor varianza de la solución.

Cuándo usarlo

Datasets enormes o en flujo continuo; cuando un pase completo es costoso.

¿Cómo elegir K?

- **Codo (Elbow)**: gráfica de SSE vs. K; busca punto de inflexión.
- **Silhouette** $\in [-1, 1]$: cohesión vs. separación promedio.
- **Gap Statistic**: compara SSE observado vs. referencia nula.

Heurística rápida

Empieza con varios K, compara Silhouette/Gap y **prefiere el más simple** si el beneficio marginal es pequeño.

Índice de Silhouette

Idea principal

Evalúa qué tan bien se adapta cada punto a su propio cluster comparado con los demás. Combina cohesión (proximidad intra–cluster) y separación (distancia inter–cluster).

- Para cada punto $\mathbf{x}_i$:

$$a(i) = \text{promedio de distancias a puntos del mismo cluster}$$

$$b(i) = \text{mínimo promedio de distancias a puntos de otro cluster}$$

- El coeficiente de Silhouette se define como:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\ b(i)\}} \quad \in [-1, 1]$$

Índice de Silhouette

Idea principal

Evalúa qué tan bien se adapta cada punto a su propio cluster comparado con los demás. Combina cohesión (proximidad intra–cluster) y separación (distancia inter–cluster).

- Para cada punto $\mathbf{x}_i$:

$$a(i) = \text{promedio de distancias a puntos del mismo cluster}$$

$$b(i) = \text{mínimo promedio de distancias a puntos de otro cluster}$$

- El coeficiente de Silhouette se define como:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\ b(i)\}} \quad \in [-1, 1]$$

Interpretación

- $s(i) \approx 1$: bien agrupado.
- $s(i) \approx 0$: cerca de la frontera entre clusters.

Gap Statistic

Motivación

Compara la calidad del agrupamiento observado con lo que se esperaría si los datos no tuvieran estructura (distribución aleatoria de referencia).

1. Ejecutar k-means para $K = 1, \ldots, K_{\text{máx}}$ y calcular:

$$W_K = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

2. Generar muestras de referencia sin estructura (uniformes en el mismo rango) y obtener su W_K^* esperado.

3. Definir la **Gap Statistic**:

$$\text{Gap}(K) = \mathbb{E}[\log W_K^*] - \log W_K$$

Partición de Voronoi inducida por los centroides

- K-means induce regiones de **Voronoi** en $\mathbb{R}^p$: cada punto se asigna al centroide más cercano.
- La frontera entre clusters son **hiperplanos** (con métrica euclidiana).

Consecuencia

Favorece clusters convexos y **aprox. esféricos**; formas no convexas quedan mal partidas.

Por qué el centroide es la media

- Para un cluster C, el óptimo de $\sum_{x \in C} \|x - \mu\|^2$ respecto a μ es $\bar{x}_C$ (la media del cluster).
- Esto justifica la actualización $\mu_k \leftarrow \frac{1}{|C_k|} \sum_{x \in C_k} x$.

Coste computacional y parada

- Número de iteraciones suele ser moderado, pero **depende de datos/inicialización**.
- **Parada**: (i) asignaciones no cambian; (ii) $\Delta J/J < \varepsilon$; (iii) tope de iteraciones.

Práctico

Estandarizar datos reduce problemas numéricos y acelera la convergencia.

Preprocesamiento: decisiones que importan

- **Escalado**: estándar o robusto (mediana/IQR) si hay outliers.
- **Selección/ponderación de variables**: evita que una variable domine.
- **Tratamiento de NA**: imputación coherente con la métrica.
- **Proyección previa** (PCA): reduce ruido y colinealidad.

Checklist de buenas prácticas

1. Escala y limpia (outliers, NA).
2. Prueba varios K y *seeds* (k-means++).
3. Compara métricas internas (Silhouette/DB/CH) y simplicidad.
4. Inspección visual (PCA/UMAP).
5. Reporta μ_k, tamaños de cluster y "semilla".

Pausa

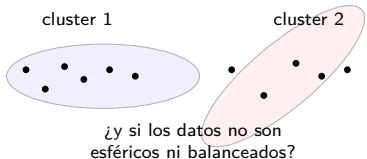
Limitaciones de k-means: Motivación Formal

- k-means minimiza la **suma de cuadrados intra-cluster**:

$$J(C, \mu) = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

donde $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$.

- **Problemas fundamentales**:
  - **Convexidad**: Asume clusters esféricos (elipsoidales con misma covarianza)
  - **No identificabilidad**: Múltiples mínimos locales, sensibilidad a inicialización
- **Alternativa**: Agrupamiento jerárquico proporciona:
  - Múltiples resoluciones simultáneas
  - Estructura de proximidad global
  - No requiere convexidad de clusters

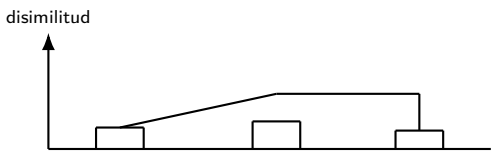


Motivación: hacia el agrupamiento jerárquico

- En lugar de fijar un número k, podemos **construir toda una jerarquía** de agrupaciones.
- Comenzamos con cada punto como su propio grupo y fusionamos los más similares.
- El proceso genera una secuencia anidada de particiones:

$$\{1\}, \ldots, \{n\} \;\Rightarrow\; \cdots \;\Rightarrow\; \{1, \ldots, n\}.$$

- El resultado es un **dendrograma**, que muestra cómo los clusters se forman a diferentes niveles de similitud.



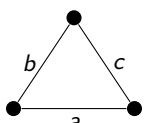
Así pasamos de una sola partición fija (k-means) a una jerarquía completa de agrupaciones.

Fundamentos Matemáticos

Una **ultramétrica** $h : X \times X \to \mathbb{R}_+$ satisface:

1. $h(x, y) = 0 \Leftrightarrow x = y$
2. $h(x, y) = h(y, x)$
3. **Desigualdad ultramétrica**: $h(x, y) \leq \max\{h(x, z), h(z, y)\}$

Propiedad clave: En una ultramétrica, todos los triángulos son isósceles con base menor o igual.



$\max(a, b, c)$ aparece al menos dos veces

Implicación: La estructura jerárquica impone restricciones fuertes en las distancias.

Del Espacio Ultramétrico al Dendrograma

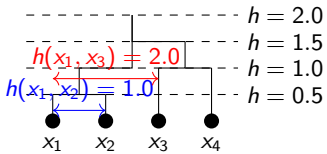
Existe una correspondencia biunívoca entre ultramétricas y dendrogramas

Espacio Ultramétrico

- $h(x, y)$ = distancia entre puntos
- Desigualdad ultramétrica: $h(x, y) \leq \max h(x, z), h(z, y)$
- $h(x, y)$ = altura del ancestro común más bajo

Dendrograma

- Altura = nivel de fusión
- Estructura arbórea
- Altura(x,y) = distancia ultramétrica



Interpretación: La distancia ultramétrica $h(x_i, x_j)$ es exactamente la altura en el dendrograma donde los clusters que contienen x_i y x_j se fusionan. Notar que la desigualdad es diferente a la desigualdad del triangulo ya que $h(x, y) \leq h(x, z) + h(z, y)$

Cómo se Construye la Ultramétrica

Theorem

La función h definida por el proceso de clustering jerárquico es una ultramétrica.

Demostración intuitiva.

Para cualquier x, y, z:

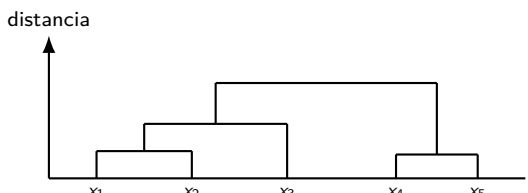
- Sean t_{xy}, t_{xz}, t_{yz} los tiempos de fusión
- Por construcción del algoritmo, $t_{xy} \leq \max(t_{xz}, t_{yz})$
- Por tanto, $h(x, y) \leq \max(h(x, z), h(z, y))$

□

Conclusión: El dendrograma es la representación visual del espacio ultramétrico inducido por el clustering.

Agrupamiento jerárquico: idea general

- Construye una *jerarquía* de particiones: de la más fina $\{1\}, \ldots, \{n\}$ a la más gruesa $\{1, \ldots, n\}$.
- En cada paso se fusionan los dos clusters más cercanos según una **regla de enlace** (*linkage*).
- Resultado visual: un **dendrograma** que muestra el orden y la altura de las fusiones.



**

Reglas de Enlace: Formalización Matemática

Sea $d : X \times X \to \mathbb{R}_+$ una métrica, $C, D \subseteq X$ clusters.

Definiciones Formales

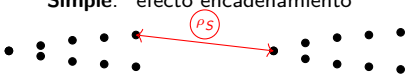
- **Single Linkage**:
$$\rho_S(C, D) = \text{mín}\{d(x, y) : x \in C, y \in D\}$$
- **Complete Linkage**:
$$\rho_C(C, D) = \text{máx}\{d(x, y) : x \in C, y \in D\}$$
- **Average Linkage (UPGMA)**:
$$\rho_A(C, D) = \frac{1}{|C||D|} \sum_{x \in C} \sum_{y \in D} d(x, y)$$
- **Ward Linkage**:
$$\rho_W(C, D) = \frac{|C||D|}{|C| + |D|} \|\mu_C - \mu_D\|^2$$

  donde μ_C, μ_D son centroides.

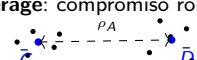
Propiedad de Consistencia: Solo Ward garantiza optimalidad local para SSE.

Distancias entre clusters (*linkage*): Interpretación Geométrica

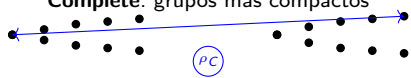
Simple: "efecto encadenamiento"



Average: compromiso robusto



Complete: grupos más compactos



Ward: minimiza varianza

Algoritmo aglomerativo (bottom-up)

Algorithm 2: Clustering jerárquico aglomerativo

Entrada: Matriz de distancias $[d(x_i, x_j)]$, regla de enlace ρ
Salida: Dendrograma y jerarquía de particiones

1 Inicializar $\mathcal{C} \leftarrow \{\{1\}, \ldots, \{n\}\}$
2 **while** $|\mathcal{C}| > 1$ **do**
3 | Elegir $C, D \in \mathcal{C}$ que minimicen $\rho(C, D)$
4 | Fusionar: $E \leftarrow C \cup D$ y actualizar $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{C, D\}) \cup \{E\}$
5 | Actualizar distancias $\rho(E, \cdot)$ según la regla de enlace
6 **end**

- Cada fusión añade una "rama" al dendrograma a una *altura* igual a la distancia de la fusión.
- Para elegir k clusters, *corta* el dendrograma a una altura y toma las componentes resultantes.

Propiedades Estadísticas y Consistencia

Consistencia de Enlaces:

- **Single**: Sensible a ruido, produce encadenamiento
- **Complete**: Robustez a outliers, clusters compactos
- **Ward**: Minimiza incremento de varianza intra-cluster

Teorema (Fisher, 1958): Para datos Gaussianos esféricos, Ward produce la partición de máxima verosimilitud.

Propiedad de Monotonicidad:

- Alturas de fusión no decrecientes
- Garantizado para single, complete, average, Ward
- Violación indica inconsistencia en datos

Validación y Selección del Número de Clusters

Métodos de Validación Interna:

- **Índice de Silueta**:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

donde $a(i)$: distancia intra-cluster, $b(i)$: distancia al cluster más cercano

Métodos de Corte:

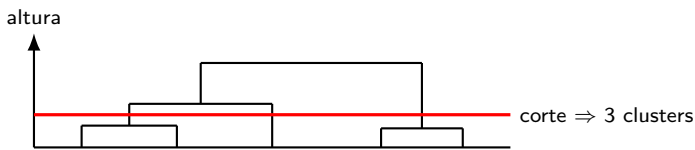
- **Método del Codo**: Maximizar $\Delta h_k = h_k - h_{k-1}$
- **Método de Gap**:

$$\text{Gap}(k) = E[\log W_k] - \log W_k$$

donde W_k: suma de cuadrados intra-cluster

Interpretación y selección de k

- **Dendrograma**: la *altura* de corte controla la granularidad de la partición.
- **Heurísticas**: "codo" en alturas de fusión; métricas de calidad (p. ej., silueta) calculadas tras cortar en k.
- **Elección del enlace**:
  - Simple: detecta estructuras alargadas, pero encadena.
  - Complete: favorece grupos compactos, sensible a outliers.
  - Average: compromiso entre ambos.
  - Ward: minimiza varianza intra-grupo (cercano a k-means en espíritu).



Extensiones y Casos Especiales

Datos Categóricos:

- **Métrica de Gower**:

$$d_G(x, y) = \frac{1}{p} \sum_{j=1}^{p} \delta_j(x_j, y_j)$$

  donde δ_j depende del tipo de variable

- Enlace preferido: Average o Ward (con modificaciones)

Datos de Alta Dimensión:

- **Maldición de dimensionalidad**: $d(x, y) \to$ constante cuando $p \to \infty$
- Solución: Reducción de dimensionalidad previa (PCA, t-SNE)

Comparación Teórica: k-means vs. Jerárquico

Propiedad	***k*-means**	**Jerárquico**
Convexidad clusters	Requerida	No requerida
Número clusters k	Pre-fijado	Determinado post-hoc
Optimalidad	Local	Subóptima (aglomerativo)
Robustez a outliers	Media	Depende del enlace
Interpretabilidad	Media	Alta (dendrograma)
Consistencia estadística	Bajo Gaussianos	Bajo Gaussianos esféricos

Recomendaciones:

- **Jerárquico**: n pequeño, estructura desconocida, interpretabilidad
- ***k*-means**: n grande, clusters convexos, eficiencia computacional
- **Híbrido**: Inicialización jerárquica + refinamiento con k-means

Limitaciones y Buenas Prácticas

- **Escala de d**: el dendrograma depende críticamente de la métrica y del escalado de variables.
- **Outliers**: pueden elevar alturas tempranas y alterar cortes; considerar limpieza/robustez.
- **Tamaños desbalanceados**: complete/average suelen ser más estables que single; Ward favorece tamaños similares.
- **Reproducibilidad**: fijar desempates, documentar preprocesamiento y la regla de enlace.

Buenas Prácticas:

- Estandarizar variables antes de calcular distancias
- Probar múltiples enlaces y comparar estabilidad
- Validar con métodos internos y externos cuando sea posible
- Considerar enfoques híbridos para grandes volúmenes de datos

Resumen y Conclusiones

- **Agrupamiento jerárquico** proporciona una alternativa flexible a métodos particionales como k-means
- **Ultramétricas** ofrecen el marco matemático para entender la estructura jerárquica
- La **elección del enlace** determina las propiedades estadísticas del agrupamiento
- **Validación rigurosa** es esencial para la selección del número de clusters

Próximos temas:

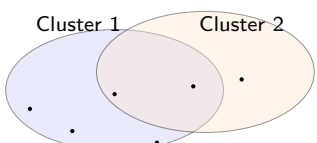
- Agrupamiento espectral
- Modelos de mezclas Gaussianas (EM algorithm)
- Métodos de agrupamiento para datos de alta dimensión

Referencias clave: Ward (1963), Lance & Williams (1967), Johnson (1967), Murtagh & Contreras (2012)

Pausa

Motivación

- K-means realiza asignaciones *duras*: cada punto pertenece a un único cluster.
- Supone grupos esféricos y de igual tamaño.
- En muchos casos, los datos presentan formas elípticas o solapadas.
- **Idea**: permitir que cada punto pertenezca parcialmente a varios clusters.



Modelos de mezcla

Los datos provienen de una mezcla de K distribuciones:

$$p(x) = \sum_{k=1}^{K} \pi_k f(x \mid \theta_k),$$

donde:

- π_k: proporción del cluster k, $\pi_k \geq 0$, $\sum_k \pi_k = 1$.
- $f(x \mid \theta_k)$: densidad del componente k.

Modelos de mezcla

Los datos provienen de una mezcla de K distribuciones:

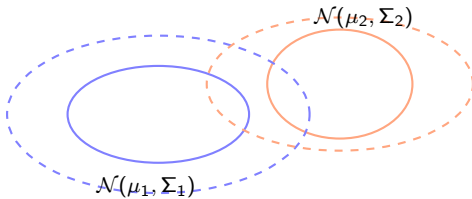
$$p(x) = \sum_{k=1}^{K} \pi_k f(x \mid \theta_k),$$

donde:

- π_k: proporción del cluster k, $\pi_k \geq 0$, $\sum_k \pi_k = 1$.
- $f(x \mid \theta_k)$: densidad del componente k.

Caso gaussiano:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$



Log-verosimilitud del modelo

Dado $X = \{x_1, \ldots, x_n\}$:

$$\ell(\Theta) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$$

- Difícil de maximizar directamente.
- Introducimos variables latentes $z_i \in \{1, \ldots, K\}$ indicando el cluster de origen.

Log-verosimilitud del modelo

Dado $X = \{x_1, \ldots, x_n\}$:

$$\ell(\Theta) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$$

- Difícil de maximizar directamente.
- Introducimos variables latentes $z_i \in \{1, \ldots, K\}$ indicando el cluster de origen.

Objetivo: Estimar $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$ que maximiza la verosimilitud.

Idea del algoritmo EM

Alternamos dos pasos:
E-step (Expectation):

$$\gamma_{ik} = P(z_i = k \mid x_i, \Theta^{(t)}) = \frac{\pi_k^{(t)} \mathcal{N}(x_i \mid \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_j \pi_j^{(t)} \mathcal{N}(x_i \mid \mu_j^{(t)}, \Sigma_j^{(t)})}$$

M-step (Maximization):

$$\pi_k^{(t+1)} = \frac{1}{n} \sum_i \gamma_{ik}, \quad \mu_k^{(t+1)} = \frac{\sum_i \gamma_{ik} x_i}{\sum_i \gamma_{ik}}, \quad \Sigma_k^{(t+1)} = \frac{\sum_i \gamma_{ik} (x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^T}{\sum_i \gamma_{ik}}$$

Principio de aumento de la verosimilitud

El EM maximiza el funcional auxiliar:

$$Q(\Theta, \Theta^{(t)}) = \mathbb{E}_{Z|X,\Theta^{(t)}}[\log p(X, Z \mid \Theta)]$$

Principio de aumento de la verosimilitud

El EM maximiza el funcional auxiliar:

$$Q(\Theta, \Theta^{(t)}) = \mathbb{E}_{Z|X,\Theta^{(t)}}[\log p(X, Z \mid \Theta)]$$

Propiedad:

$$\ell(\Theta^{(t+1)}) \geq \ell(\Theta^{(t)}),$$

por lo que la verosimilitud no disminuye en cada iteración.

Principio de aumento de la verosimilitud

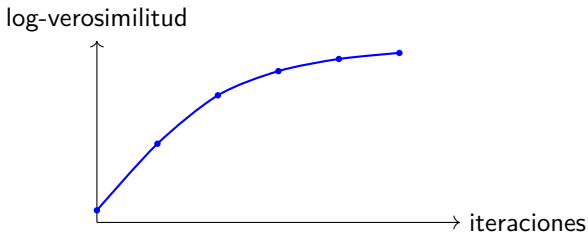
El EM maximiza el funcional auxiliar:

$$Q(\Theta, \Theta^{(t)}) = \mathbb{E}_{Z|X,\Theta^{(t)}}[\log p(X, Z \mid \Theta)]$$

Propiedad:

$$\ell(\Theta^{(t+1)}) \geq \ell(\Theta^{(t)}),$$

por lo que la verosimilitud no disminuye en cada iteración.



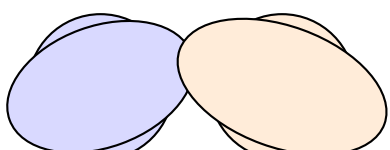
Comparación con K-means

- Si $\Sigma_k = \sigma^2 I$ y se usa asignación dura:

$$\gamma_{ik} = \begin{cases} 1 & \text{si } k = \arg\min_j \|x_i - \mu_j\|^2, \\ 0 & \text{en otro caso.} \end{cases}$$

- Entonces EM se reduce a K-means.
- EM permite formas elípticas y pertenencias suaves (*soft clustering*).

EM (elipses)



K-means (esferas)

Pseudocódigo del algoritmo EM

Algorithm 3: Algoritmo de Expectation–Maximization

Data: Datos $X = \{x_1, \ldots, x_n\}$, número de clusters K
Result: Parámetros π_k, μ_k, Σ_k
Inicializar π_k, μ_k, Σ_k aleatoriamente
while *no converja* **do**
    E-step: calcular γ_{ik} para todo i, k
    M-step: actualizar parámetros usando γ_{ik}
    Calcular log-verosimilitud $\ell(\Theta)$
end

Resumen y extensiones

- EM es un método general para modelos con variables latentes.
- En clustering, permite:
  - Pertenencias suaves.
  - Covarianzas generales (clusters elípticos).
- Limitaciones:
  - Puede converger a máximos locales.
  - Requiere inicialización adecuada y K fijo.
- Extensiones:
  - Modelos no gaussianos.
  - EM variacional, mixtures de Dirichlet, y Bayesian GMM.

Continuación: del Jerárquico al Espectral

Recapitulando

- **Clustering jerárquico:** Construye una jerarquía (aglomerativo/divisivo).
- **Ventajas:** No requiere especificar k y ofrece dendrograma.
- **Limitación:** Decisiones locales $\Rightarrow$ la estructura global puede no ser óptima.

Problema fundamental

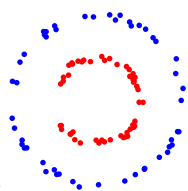
Tanto **k-means** como **jerárquico** asumen clusters compactos basados en distancias euclidianas.

Motivación: límites de métodos clásicos

Estructura “Dos Lunas”

k-means falla

Jerárquico falla



Círculos concéntricos

Necesidad

Capturar **conectividad** local/global, no solo distancia euclidiana.

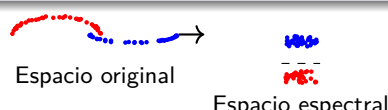
Idea central del clustering espectral

Intuición geométrica

Transformar los datos a un **nuevo espacio** donde los clusters sean linealmente separables.

Analogía

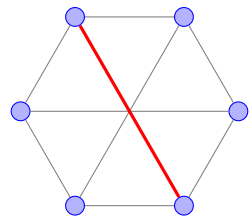
"Estirar" y "reformar" el espacio original para revelar conectividad.



Representación como grafo

Construcción del grafo

- Datos: $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$.
- Grafo ponderado $G = (V, E)$ con pesos w_{ij}.
- **Kernel gaussiano**: $w_{ij} = \exp\left(-\dfrac{\|x_i - x_j\|^2}{2\sigma^2}\right)$.



Grafo de similitud

Del grafo al problema espectral

Construcción

- A partir de los datos $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, construimos un grafo ponderado $G = (V, E)$ con pesos $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$.
- Definimos la matriz de grados $D_{ii} = \sum_j w_{ij}$.
- La matriz Laplaciana se define como $L = D - W$.

Propiedad clave

$$v^T L v = \frac{1}{2} \sum_{i,j} w_{ij}(v_i - v_j)^2$$

donde $v \in \mathbb{R}^n$ mide la **suavidad** de los valores sobre el grafo.

Problema espectral

Formulación

Resolver el sistema

$$Lv = \lambda v,$$

obteniendo pares (λ_i, v_i) donde v_i son los **eigenvectores** y λ_i los **eigenvalores**.

Propiedades

- $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.
- v_1 es constante y los siguientes $v_2, \ldots, v_k$ capturan las principales estructuras del grafo.

Idea central

Los primeros k eigenvectores $v_1, \ldots, v_k$ definen un nuevo **espacio espectral** donde los clusters son más evidentes.

Del problema espectral al nuevo espacio

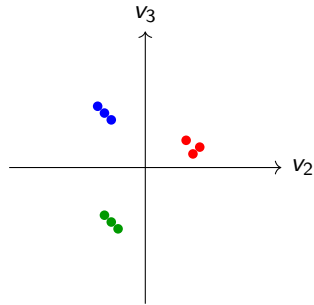
Matriz de eigenvectores

$$U = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_k \\ | & | & & | \end{bmatrix}$$

Cada fila de U corresponde a un punto de los datos representado como:

$$x_i \longmapsto [v_1(i), v_2(i), \ldots, v_k(i)].$$

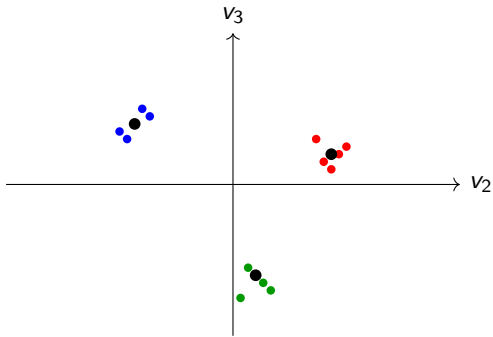
Visualización



Interpretación

U redefine los puntos en un espacio donde la conectividad del grafo se traduce en proximidad geométrica.

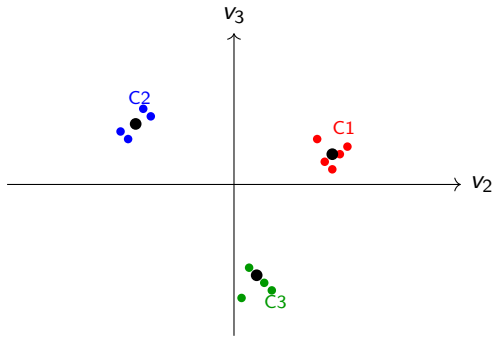
Clustering en el espacio espectral



Resultado

Al aplicar k-means sobre las filas de U, los clusters se separan fácilmente en el **espacio espectral**.

Paso 5: K-means en el espacio espectral



Resultado

En el **nuevo espacio**, los clusters son fácilmente separables con k-means.

Variantes del laplaciano: ¿cuál elegir?

No normalizado

$L = D - W$

Simple
Sensible
a grados

Simétrico

$L_{sym} = D^{-1/2} L D^{-1/2}$

Estable
eigenvectores
ortogonales

Random walk

$L_{rw} = D^{-1} L$

Interpretación
probabilística

Recomendación práctica

L_{sym} suele ofrecer mejores propiedades teóricas y estabilidad numérica.

Ventajas y consideraciones

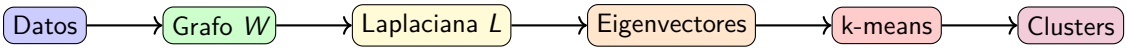
Ventajas

- Captura estructura no lineal
- No asume forma paramétrica de cluster
- Base teórica sólida
- Robusto ante outliers

Desventajas

- Coste computacional
- Elección de parámetros (σ, vecinos/k)
- Sensible a la construcción del grafo

Resumen: el flujo del clustering espectral



Idea esencial

El clustering espectral **cambia la representación**: lleva los puntos a un **nuevo espacio** donde la estructura de clusters se vuelve evidente.

Matemática elegante + representación adecuada = resultados poderosos.