

---

# Clase 097 — Agglomerative, BIRCH, Mean Shift, Affinity Propagation, Spectral

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9 § Other Clustering Algorithms.

Duración estimada: 70 min.

# Clase 097 — Agglomerative, BIRCH, Mean Shift, Affinity Propagation, Spectral

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9 § Other Clustering Algorithms. Duración estimada: 70 min.

## Objetivo

Que el alumno conozca el zoológico de algoritmos de clustering más allá de K-Means y DBSCAN — Agglomerative (jerárquico, lee dendrogramas), BIRCH (escalable a millones de filas), Mean Shift (denso, sin especificar  $k$ ), Affinity Propagation (elige exemplars por message-passing) y Spectral Clustering (clustering vía autovectores del grafo de similitud) — y sepa cuándo elegir cada uno según tamaño del dataset, forma de los clusters y necesidad de jerarquía.

## Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Construir un dendrograma con `scipy.cluster.hierarchy.linkage` y cortarlo a una altura dada para obtener clusters.
2. Elegir un linkage (ward, complete, average, single) según la forma esperada de los clusters y conocer el efecto del chaining en single.
3. Usar BIRCH para datasets que no entran en memoria, ajustando `threshold` y `branching_factor`.
4. Aplicar Mean Shift ajustando `bandwidth` (con `estimate_bandwidth`) y entender por qué descubre el número de clusters automáticamente.
5. Decidir entre Affinity Propagation y Spectral Clustering según escalabilidad (AP es  $O(n^2)$  en memoria) y geometría (Spectral funciona en clusters no convexos).

## Temas

#	Algoritmo	Hiperparámetro clave	k a priori	Complejidad	Cuándo usarlo
1	Agglomerative	linkage, n_clusters o d	Sí (o threshold)	$O(n^3) / O(n^2 \log n)$	Querés jerarquía + dendrograma; n s
2	BIRCH	threshold, branching_f	Opcional	$O(n)$	Datasets grandes (millones) que no e
3	Mean Shift	bandwidth	No	$O(n^2)$ por iter	Modos de densidad; clusters de tama
4	Affinity Propagation	damping, preference	No	$O(n^2 \cdot T)$	Pocos puntos, querés exemplars rea
5	Spectral	n_clusters, affinity (rbf)	Sí	$O(n^3)$ (eigen)	Clusters no convexos, manifolds, gra

## Definiciones y características

Clustering aglomerativo (jerárquico)

: Algoritmo bottom-up: arranca con cada punto como su propio cluster y en cada paso fusiona los dos clusters más cercanos hasta llegar a uno solo. Produce una jerarquía completa que se visualiza como dendrograma. En sklearn: `AgglomerativeClustering(n_clusters=k)` o `distance_threshold=d` (entonces `n_clusters=None`).

Linkage

: Criterio para medir la distancia entre dos clusters durante la fusión. Las cuatro opciones que importan:

- ward (default sklearn): minimiza la varianza intra-cluster al fusionar. Tiende a producir clusters de tamaño parecido. Solo con distancia euclidiana.
- complete (max-linkage): distancia entre los dos puntos más lejanos. Produce clusters compactos.
- average: promedio de todas las distancias entre puntos de ambos clusters. Compromiso razonable.
- single (min-linkage): distancia entre los dos puntos más cercanos. Sufre chaining — un puente fino de puntos une dos clusters que deberían quedar separados.

### Dendrograma

: Diagrama de árbol invertido que muestra el orden y la altura (= distancia entre clusters fusionados) de cada merge. Cortar con una línea horizontal a altura  $h$  da los clusters resultantes. Se grafica con `scipy.cluster.hierarchy.dendrogram(Z)` donde  $Z = \text{linkage}(X, \text{method}='ward')$ .

### BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

: Algoritmo de dos pasos para datasets enormes. (1) Recorre los datos una sola vez construyendo un CF-tree (Clustering Feature tree) que comprime puntos en sub-clusters compactos. (2) Aplica un clustering tradicional sobre los sub-clusters. Streaming-friendly (online). Hiperparámetros: `threshold` (radio máximo de un sub-cluster — más chico, más sub-clusters) y `branching_factor` (hijos máximos por nodo).

### Mean Shift

: Algoritmo basado en densidad. Cada punto "trepa" iterativamente hacia el modo (máximo local) de la densidad estimada con un kernel gaussiano de ancho `bandwidth`. Puntos que convergen al mismo modo forman un cluster. No requiere  $k$ . Sensible al `bandwidth`: chico → muchos clusters chiquitos; grande → todo un cluster. `sklearn.cluster.estimate_bandwidth(X)` da un valor de arranque razonable.

### Bandwidth

: Ancho del kernel en Mean Shift. Análogo conceptual al `eps` de DBSCAN — define qué tan "lejos" mira cada punto para estimar densidad.

### Affinity Propagation

: Algoritmo de message passing entre puntos. Cada punto manda dos tipos de mensajes (responsibility y availability) a los demás hasta que emerge un subconjunto de exemplars (puntos representativos reales del dataset) y cada otro punto se asigna al exemplar más afín. No requiere  $k$ . Caro:  $O(n^2)$  en memoria y  $O(n^2 \cdot T)$  en tiempo — no escala más allá de unos miles de puntos. Hiperparámetros: `damping` (entre 0.5 y 1, para evitar oscilaciones) y `preference` (controla cuántos exemplars emergen).

### Spectral Clustering

: Construye una matriz de similitud  $S$  (típicamente con kernel RBF o  $k$ -NN), calcula el grafo Laplaciano  $L = D - S$ , toma los  $k$  autovectores de menor autovalor, los apila como nuevas features y corre K-Means en ese embedding. Funciona donde K-Means falla: clusters no convexos, dos lunas entrelazadas, círculos concéntricos. Cuello de botella: eigendecomposición  $O(n^3)$ .

### Similarity matrix

: Matriz  $n \times n$  donde  $S[i,j]$  mide qué tan parecidos son los puntos  $i$  y  $j$  (alta similitud = cerca). Es la entrada de Spectral y Affinity Propagation. Para Spectral con kernel RBF:  $S[i,j] = \exp(-\gamma \cdot \|x_i - x_j\|^2)$ .

## Dataset / recursos

- `make_blobs` (sklearn) — para Agglomerative y BIRCH (clusters convexos).
- `make_moons` y `make_circles` — clusters no convexos, donde Spectral brilla y

K-Means/Agglomerative-ward fallan.

- Dataset opcional escalable: generará 1M de puntos sintéticos para probar BIRCH vs K-Means en tiempo y memoria.

## Ejercicios

1. Dendrograma sobre `make_blobs`. Generá 50 puntos en 4 blobs. Calculá  $Z = \text{linkage}(X, \text{method}='ward')$  y graficá el dendrograma. Cortá a una altura que dé 4 clusters. Comparalo con `AgglomerativeClustering(n_clusters=4)`.
2. Linkage comparison. Sobre `make_moons(n_samples=300, noise=0.05)`, ajustá `AgglomerativeClustering(n_clusters=2)` con `linkage='ward', 'complete', 'average', 'single'`. Ploteá los 4 resultados lado a lado. ¿Cuál recupera las dos lunas? ¿Por qué?
3. BIRCH escalable. Generá 100k puntos con `make_blobs`. Medí tiempo y memoria de `BIRCH(n_clusters=5)` vs `KMeans(n_clusters=5)`. Variá `threshold`  $\{0.1, 0.5, 1.0\}$  y mostrá cómo cambia el número de sub-clusters internos del CF-tree.
4. Mean Shift sin saber k. Sobre 3 blobs claros, corré `MeanShift(bandwidth=estimate_bandwidth(X, quantile=0.2))`. Verificá que recupera 3 clusters automáticamente. Después poné `quantile=0.5` y observá cómo colapsa a menos clusters.
5. Spectral vs K-Means en `make_circles`. Generá dos círculos concéntricos con `make_circles(n_samples=500, factor=0.5, noise=0.05)`. Ajustá `KMeans(n_clusters=2)` y `SpectralClustering(n_clusters=2, affinity='nearest_neighbors')`. Graficá ambos. Documentá por qué Spectral funciona y K-Means no.

## Homework verificable

Notebook con: (a) dendrograma de `load_iris().data` usando `ward` linkage, cortado para obtener 3 clusters; (b) tabla comparativa de Adjusted Rand Index entre las labels verdaderas y las predichas por Agglomerative, BIRCH, Mean Shift, Affinity Propagation y Spectral — corriendo los 5 sobre el mismo Iris; (c) gráfico 2D (con las 2 primeras PCs) coloreado por las labels de cada método; (d) párrafo justificando qué método ganó y por qué.

Criterio de aceptación: el dendrograma se visualiza correctamente con eje  $y = \text{distancia}$ . La tabla muestra ARI  $[-1, 1]$  para los 5 métodos. Spectral y Agglomerative-ward deberían superar 0.7 en Iris.

## Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
AgglomerativeClustering con <code>linkage='ward'</code>	<code>ward</code> solo soporta distancia euclidiana. Fix
Dendrograma con miles de hojas, ilegible	Demasiados puntos. Fix: pasale <code>truncate_m</code>
MeanShift corre eterno o devuelve 1 solo c	<code>bandwidth</code> mal calibrado (default = estimat
AffinityPropagation no converge — warning	<code>damping</code> muy bajo (oscilaciones). Fix: subí
SpectralClustering con <code>affinity='rbf'</code> da c	<code>gamma</code> del RBF mal escalado. Fix: usá <code>affin</code>

## Preguntas frecuentes

¿Cuándo Agglomerative en lugar de K-Means?

Cuando querés (a) explorar la estructura jerárquica del dataset con un dendrograma antes de decidir  $k$ , (b) no asumir clusters esféricos (con average o complete linkage), o (c) reproducibilidad determinística — Agglomerative no depende de inicialización aleatoria. Costo:  $O(n^3)$ , inviable para  $n > \sim 10k$ .

¿BIRCH reemplaza a MiniBatch K-Means?

Casi. Ambos son escalables. BIRCH gana cuando los datos llegan en streaming o no entran ni siquiera en lotes (es one-pass). MiniBatch K-Means es más simple y suele ser suficiente para datasets que entran en disco. Si dudás, empezá por MiniBatch K-Means.

¿Cómo elijo el bandwidth de Mean Shift sin probar a mano?

Usá `sklearn.cluster.estimate_bandwidth(X, quantile=0.2, n_samples=500)`. El `quantile` controla el ancho del kernel basado en distancias entre pares de puntos muestreados. Empezá con 0.2 y bajá si querés clusters más finos.

¿Affinity Propagation sirve para algo en la práctica?

Para datasets chicos ( $< 1000$  puntos) donde necesitás exemplars reales — por ejemplo, elegir 50 reviews representativas de 800. Para clustering general, los demás algoritmos lo superan en velocidad y robustez.

¿Spectral Clustering es lo mismo que clustering sobre PCA?

No. PCA proyecta sobre direcciones de máxima varianza (lineal). Spectral usa autovectores del grafo Laplaciano de la matriz de similitud — captura estructura no lineal (manifolds). Por eso recupera dos lunas entrelazadas, donde PCA + K-Means falla.

## Referencias

- Géron, Hands-On ML, cap. 9 § Other Clustering Algorithms.
- scikit-learn — Clustering overview (tabla comparativa)
- scipy linkage + dendrogram
- Frey & Dueck (2007) — Affinity Propagation paper
- von Luxburg (2007) — A Tutorial on Spectral Clustering

## Siguiente clase

Clase 098 — Gaussian Mixture Models

## Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

## Archivos complementarios

- notebook.ipynb