Clase 096 — DBSCAN

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9. Duración estimada: 55 min.

Clase 096 — DBSCAN

> *Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9. Duración estimada: 55 min.*

Objetivo

Que el alumno aplique DBSCAN (Density-Based Spatial Clustering of Applications with Noise) para descubrir clusters de forma arbitraria e identificar outliers nativamente, sin tener que predefinir k como en K-Means. Que sepa elegir eps con un k-distance plot y entienda cuándo conviene escalar a HDBSCAN.

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Ejecutar DBSCAN con sklearn.cluster.DBSCAN y tunear eps y min_samples para un dataset 2D.
2. Elegir eps mirando el codo del k-distance plot (no a ojo).
3. Identificar outliers vía la etiqueta -1 que DBSCAN asigna a los puntos ruido.
4. Distinguir core / border / noise points y entender la noción de density-reachable.
5. Comparar DBSCAN vs K-Means y saber cuándo usar HDBSCAN (eps variable, clusters de densidad mixta).

Temas

#	Tema	Por qué importa
1	Intuición de densidad vs centroides	DBSCAN encuentra "regiones densas" — no ne
2	Hiperparámetros eps y min_samples	Son los dos botones del modelo. Mal puesto
3	Core / border / noise points	Vocabulario base para leer la salida y dep
4	k-distance plot para elegir eps	Método estándar para no ir a ciegas.
5	Etiqueta -1 y detección de outliers	DBSCAN es de los pocos clusterers que dete
6	Limitaciones: densidad uniforme, curse of	Por qué falla en datasets reales con clust
7	HDBSCAN como evolución	eps jerárquico: maneja densidad variable y

Definiciones y características

DBSCAN

: Algoritmo de clustering basado en densidad. Agrupa puntos que están densamente conectados y marca como ruido los que quedan en regiones de baja densidad. No requiere k. Complejidad ~O(n log n) con índice espacial.

eps (epsilon)

: Radio de la vecindad alrededor de cada punto. Define qué se considera "cerca". Es el hiperparámetro más sensible — un cambio chico colapsa o explota los clusters.

min_samples

: Cantidad mínima de puntos (incluyendo el propio) dentro de eps para que un punto se considere core. Regla

práctica: min_samples ≥ dim + 1, típicamente 4–10 para 2D.

Core point

: Punto con al menos min_samples vecinos dentro de eps. Es el "núcleo" desde el cual crece el cluster.

Border point

: Punto que está dentro del eps de un core, pero él mismo no tiene min_samples vecinos. Pertenece al cluster pero no propaga.

Noise point (outlier)

: Punto que no es core ni border. DBSCAN lo etiqueta como -1. Sale gratis del modelo, sin entrenar un detector aparte.

Density-reachable

: Relación que conecta dos puntos vía una cadena de core points. Es la definición formal de "pertenecer al mismo cluster" en DBSCAN.

HDBSCAN (Hierarchical DBSCAN)

: Evolución de DBSCAN que reemplaza eps por una jerarquía. Maneja clusters de densidades distintas, expone un parámetro más intuitivo (min_cluster_size) y devuelve probabilidades de pertenencia. Es el default moderno para clustering no supervisado.

Dataset / recursos

- make_moons(n_samples=1000, noise=0.05) de scikit-learn — dos lunas entrelazadas, el caso canónico donde K-Means falla y DBSCAN brilla.
- make_blobs con densidades distintas para mostrar la limitación de DBSCAN y motivar HDBSCAN.

Ejercicios

1. DBSCAN sobre moons. Entrená DBSCAN(eps=0.2, min_samples=5) sobre make_moons. Graficá los clusters y contá cuántos puntos quedaron como -1.

2. K-distance plot. Calculá la distancia al k-ésimo vecino más cercano (k=min_samples) para todos los puntos, ordenala y graficala. Identificá el codo y usalo como eps.

3. Sensibilidad a eps. Probá eps {0.05, 0.1, 0.2, 0.5} y reportá número de clusters y % de ruido en cada caso. Mostrá cómo eps chico = todo ruido y eps grande = un cluster.

4. DBSCAN vs K-Means. Sobre el mismo make_moons, corré ambos con k=2. Mostrá visualmente que K-Means parte las lunas por la mitad y DBSCAN las separa bien.

5. HDBSCAN sobre densidades mixtas. Generá blobs con cluster_std distinto por blob. Mostrá que un eps único en DBSCAN no puede capturar ambos, y que HDBSCAN(min_cluster_size=20) sí.

Homework verificable

Notebook que sobre un dataset 2D sintético (moons + outliers inyectados a mano) haga: (a) k-distance plot y elección razonada de eps; (b) DBSCAN con esos hiperparámetros; (c) reporte de % de outliers detectados vs inyectados; (d) comparación visual con K-Means; (e) corrida con HDBSCAN sobre blobs de densidad mixta.

Criterio de aceptación: eps justificado con el codo del k-distance plot (no a ojo). DBSCAN recupera al menos

el 80% de los outliers inyectados. La comparación con K-Means muestra explícitamente la falla de los centroides en datos no convexos.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Elegir eps a ojo y obtener un solo cluster	El parámetro es altamente sensible y depen
DBSCAN sobre features sin escalar	eps es una distancia euclidiana; si una fe
min_samples=1 o min_samples=2	Todo termina siendo core point, no hay rui
Aplicar DBSCAN en alta dimensión (>10)	Curse of dimensionality: las distancias se
Pedirle predict() a DBSCAN	DBSCAN no tiene predict — no hay forma nat

Preguntas frecuentes

¿K-Means o DBSCAN?

K-Means si los clusters son aproximadamente esféricos, de tamaño parecido, y sabés (o podés estimar) k. DBSCAN si los clusters tienen forma arbitraria, no sabés cuántos hay, o necesitás detectar outliers en la misma pasada. Para producción moderna: HDBSCAN suele ganar a ambos cuando no tenés intuición previa.

¿Por qué DBSCAN devuelve -1?

Es la etiqueta convencional para noise points — puntos que no pertenecen a ningún cluster. No es un cluster más, es la salida natural del algoritmo para outliers.

¿Cómo elijo min_samples?

Regla práctica: min_samples = 2 * dim. Para 2D, usá 4. Para 3D, 6. Más grande = clusters más robustos pero más ruido. En la práctica, min_samples mueve menos la aguja que eps.

¿DBSCAN escala a millones de puntos?

Con algorithm='ball_tree' o 'kd_tree' queda en ~O(n log n). Para >1M puntos en alta dimensión, considerá HDBSCAN o aproximaciones tipo sklearn.cluster.OPTICS.

¿Cuándo HDBSCAN vence a DBSCAN?

Siempre que los clusters tengan densidades distintas entre sí. DBSCAN usa un eps global; HDBSCAN construye una jerarquía y elige el corte óptimo por cluster. Además expone min_cluster_size que es más intuitivo de tunear que eps.

Referencias

- Géron, cap. 9 § "DBSCAN".
- scikit-learn DBSCAN user guide
- HDBSCAN docs
- Ester et al. (1996), A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise — paper original.

Siguiente clase

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

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb