

Algoritma DBSCAN

DBSCAN adalah algoritma klasterisasi berbasis kepadatan yang secara efektif mengidentifikasi klaster dalam data dengan bentuk arbitrer serta mendeteksi outlier. Algoritma ini mengelompokkan titik-titik data berdasarkan kepadatan lokalnya dengan dua parameter utama: radius lingkungan ε dan jumlah minimum titik minPts yang diperlukan untuk membentuk sebuah klaster.

1. Definisi Jarak

DBSCAN menggunakan metrik jarak $d(\cdot, \cdot)$ yang bisa dikonfigurasi melalui argumen **metric**. Secara default, digunakan jarak Euklidean:

$$d(x_i, x_j) = \left(\sum_{k=1}^d (x_i^{(k)} - x_j^{(k)})^2 \right)^{1/2}$$

Untuk metrik Minkowski umum dengan parameter p :

$$d(x_i, x_j) = \left(\sum_{k=1}^d |x_i^{(k)} - x_j^{(k)}|^p \right)^{1/p}$$

2. Titik Inti dan Lingkungan ε

Diberikan himpunan data $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, untuk setiap titik x_i , ditentukan himpunan tetangga ε -nya:

$$N_\varepsilon(x_i) = \{x_j \in X \mid d(x_i, x_j) \leq \varepsilon\}$$

Titik x_i disebut *core point* apabila:

$$|N_\varepsilon(x_i)| \geq \text{minPts}$$

3. Ekspansi Klaster

Proses klasterisasi dimulai dari titik inti x_i , dan dilakukan perluasan klaster dengan menelusuri seluruh tetangga langsungnya serta tetangga tidak langsung yang juga merupakan core point. Prosedur ini dilakukan secara rekursif melalui *cluster expansion*:

- Tandai titik x_i sebagai bagian dari klaster ke- k
- Untuk setiap tetangga $x_j \in N_\varepsilon(x_i)$ yang belum dikunjungi:
 - Jika x_j adalah core point, tambahkan $N_\varepsilon(x_j)$ ke dalam himpunan tetangga
 - Tandai x_j sebagai bagian dari klaster ke- k jika belum memiliki label

4. Titik Noise

Titik $x \in X$ yang bukan merupakan core point dan tidak dapat dijangkau dari core point manapun akan diberi label sebagai *noise*, dilambangkan dengan -1 .

$$\text{noise} = \{x_i \in X \mid |N_\epsilon(x_i)| < \text{minPts} \text{ dan } x_i \text{ tidak tergabung dalam kluster manapun}\}$$

5. Kompleksitas Waktu

Kompleksitas waktu algoritma DBSCAN secara umum adalah:

$$O(n^2)$$