NEURAL INFORMATION PROCESSING SYSTEMS

Scalable DBSCAN with Random Projections

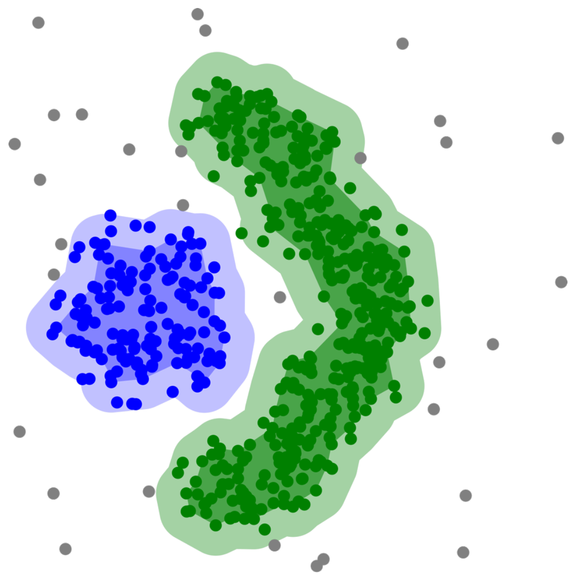
HaoChuan Xu and Ninh Pham

Project Pages: https://github.com/NinhPham/sDbscan

Paper: https://neurips.cc/virtual/2024/poster/94318

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

DBSCAN Algorithm: Overview



(Image sourced from Wikipedia)

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, pages 226–231, 1996.

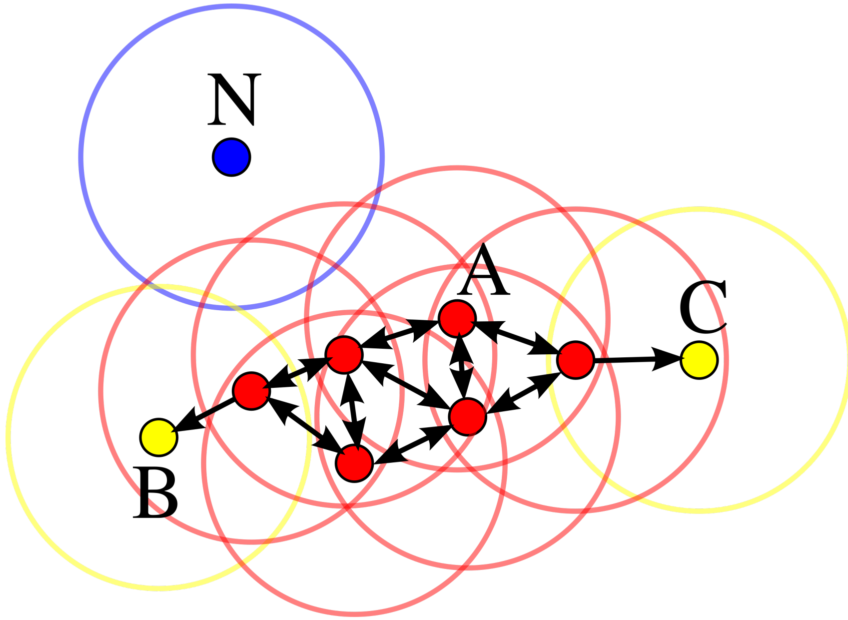
The Primary Steps of DBSCAN

DBSCAN takes parameters (ε, minpts) and performs the following two primary steps:

1. **Core points identification**: For each data point p, find all the *neighbours* x where dist(x, p) < ε. The point p is defined as *core* point if the number of its neighbours is greater than the specified minpts.

Computation Cost: $O(dn^2)$

2. **Cluster Formation**: Connect each point with its neighboring points to form the cluster

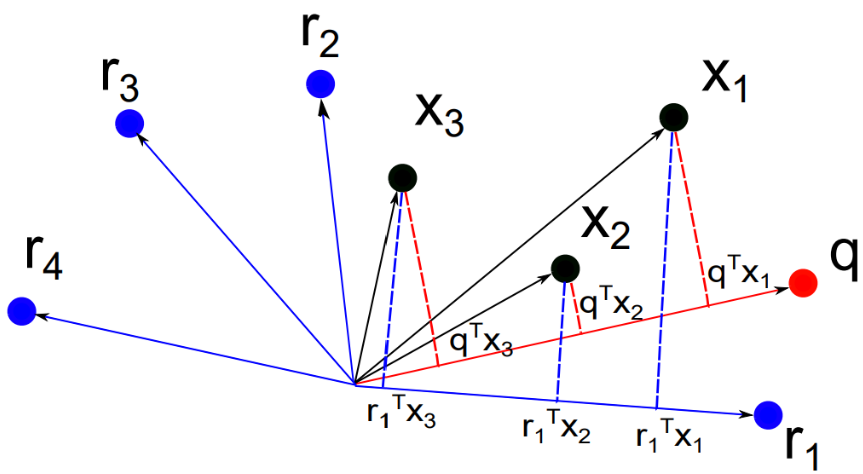


In this diagram, minPts = 4.

Random projection-based neighborhood preservation

Lemma 3.1 *For any two points $q, x \in X$ with D Gaussian random vectors generated. Suppose D is sufficient large and random $r_q = \text{argmax}_{r_i}\, q \cdot r_i$. The following statement can be deduced:*

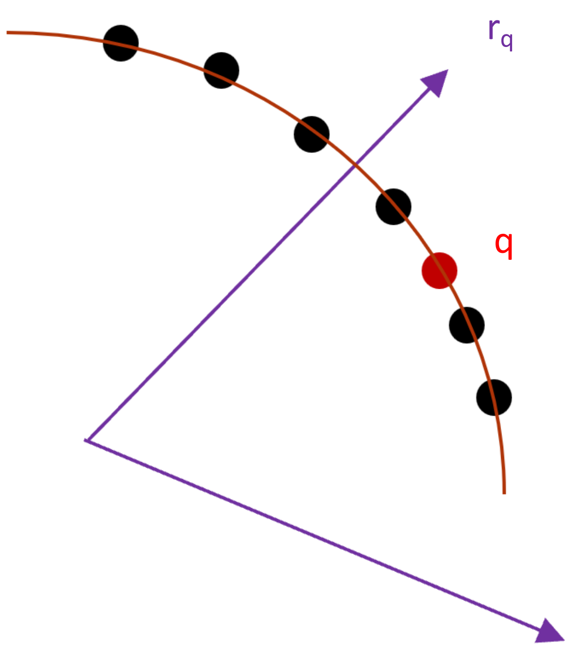
$$x^\top r_q \sim N(x^\top q\sqrt{2\ln D}, 1 - (x^\top q)^2) \tag{1}$$



Ninh Pham. Simple yet efficient algorithms for maximum inner product search via extreme order statistics. In KDD, pages 1339–1347, 2021.

sDbscan: Intuition

- Idea:
  - Use random vectors as pivots/references
  - "Neighbors of neighbors are neighbors"
    - If q is close to r_q, then q should be close to points around r_q
- sDbscan:
  - For each random vector r_i, keep top-minPts closest points in L_i
  - If q is closest to r_q, then compute dist(q, x) for all x ∈ L_q to find approximate ε-neighborhood → O(minPts) distances



sDBSCAN: Algorithm Procedures

Algorithm 2 Preprocessing

Inputs: $\mathbf{X} \subset \mathcal{S}^{d-1}$, D random vectors $\mathbf{r}_i$, $k, m = O(minPts)$
for each $\mathbf{q} \in \mathbf{X}$, compute and store top-$k$ closest and top-k furthest vectors $\mathbf{r}_i$ to $\mathbf{q}$.
for each random vector $\mathbf{r}_i$, compute and store top-m closest and top-m furthest points to $\mathbf{r}_i$.

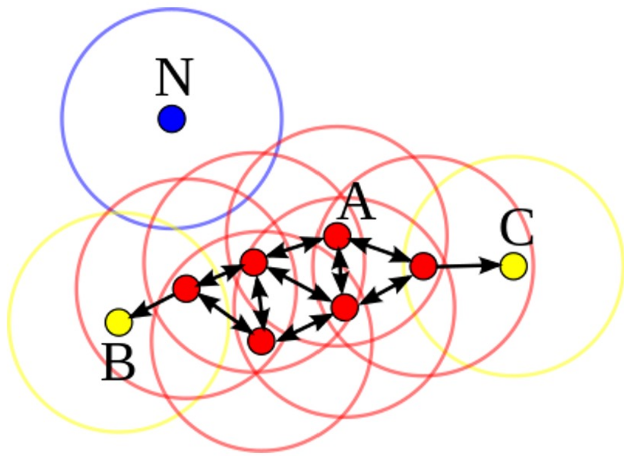
Algorithm 3 Finding core points and their approximate neighborhoods

Inputs: $\mathbf{X} \subset \mathcal{S}^{d-1}$, D random vectors $\mathbf{r}_i$, $k, \varepsilon, m = O(minPts)$
Initialize an empty set $\widetilde{B}_\varepsilon(\mathbf{q})$ for each $\mathbf{q} \in \mathbf{X}$
for each $\mathbf{q} \in \mathbf{X}$ **do**
  for each $\mathbf{r}_i$ from top-k closest (or furthest) random vectors of $\mathbf{q}$ **do**
    for each $\mathbf{x}$ from top-m closest (or furthest) points of $\mathbf{r}_i$ **do**
      if $dist(\mathbf{x}, \mathbf{q}) \leq \varepsilon$ **then**
        Insert $\mathbf{x}$ into $\widetilde{B}_\varepsilon(\mathbf{q})$ and insert $\mathbf{q}$ into $\widetilde{B}_\varepsilon(\mathbf{x})$
for each $\mathbf{q} \in \mathbf{X}$ **do**
  if $|\widetilde{B}_\varepsilon(\mathbf{q})| \geq minPts$ **then**
    Output $\mathbf{q}$ as a core point and $\widetilde{B}_\varepsilon(\mathbf{q})$ as an approximate $B_\varepsilon(\mathbf{q})$ for DBSCAN (Alg. 1)
    Output $dist(\mathbf{x}, \mathbf{q})$ for each $\mathbf{x} \in \widetilde{B}_\varepsilon(\mathbf{q})$ for OPTICS (Alg. 6)

Time Complexity: $O(n \cdot k \cdot minPts)$
Space Complexity: $O(n \cdot (d + k))$

sDbscan: Theory

- Guarantees:
  - Guarantee on recovering Dbscan's result if nearby core points share at least t = log(n) common core points (i.e. cluster is not thin anywhere)
- Extension:
  - Extend to L1, L2, Jensen-Shannon, χ^2 distances via random features
  - sOptics to guide the setting of (ε, minPts)



Two close core points share at least t common core points In their neighborhood

Challenge of DBSCAN and sDBSCAN

The clustering quality of the DBSCAN and sDBSCAN depends on the chosen ε parameter.

- It becomes more sensitive in high-dimensional space
  - Changing the ε value by 0.005 can decrease the clustering accuracy by 10% on pamap2 dataset

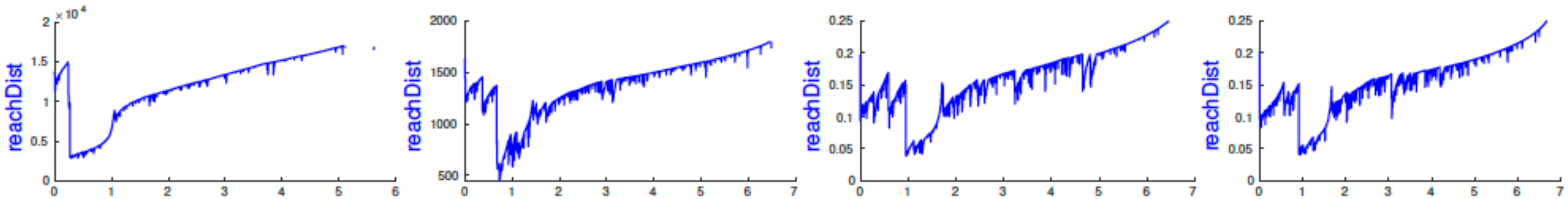
OPTICS plot is commonly used approach to select the ε parameter.

sOPTICS are designed to finding the optimal ε parameter for sDBSCAN

Experiment: Mnist (n = 70,000, d = 784)

sDbscan returns the same clustering accuracy (NMI 43%) as scikit-learn but runs **100x** faster with minPts = 50

sOptics runs in **3 seconds** while scikit-learn runs in **1.5 hours**



(a) sOptics (L1) (b) sOptics (L2) (c) sOptics (Cosine) (d) sOptics (JS)

Experiment:Mnist8m (n = 8.1M, d = 784)

sDbscan and sOptics run in 15’ in a single machine

- NMI = 38% with minPts = 50
- NMI = 40% with minPts = 100

Kernel k-mean runs in **15’** in a **supercomputer** with **32 nodes**

- NMI = 41% with k = 10 (prior knowledge of # clusters)

Scikit-learn cannot run any clustering algorithms due to memory limits