Clustering Sets in High Dimensions

Alexander Miller

Database Research Group
Department of Computer Sciences
University of Salzburg

DB Retreat, 2020

January 20, 2020

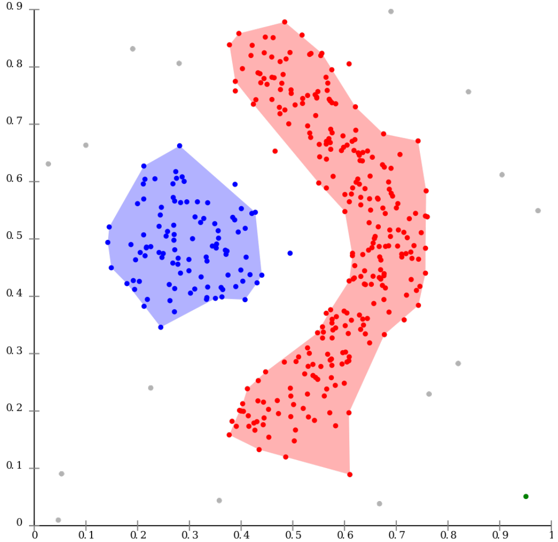


Figure: Visualization of a density-based clustering of two-dimensional data[1]

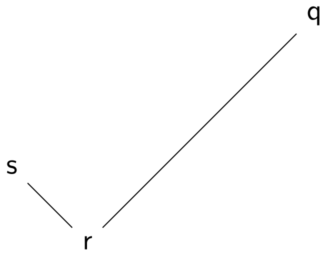
[1]By Chire - Own work, CC BY-SA 3.0,
`https://commons.wikimedia.org/w/index.php?curid=17085332`

Data

- A collection of sets containing integer tokens
- Dimensionality d is the number of different tokens in all sets

$$\begin{array}{cl} r_1 & \{1,3,5\} \\ r_2 & \{1,2,3,4\} \\ \hline \multicolumn{2}{c}{d = 5} \end{array}$$

Distance Metric



Hamming Distance

Definition (Hamming Distance)

The Hamming distance H of two sets r and s is defined as
$H(r, s) = |(r \cup s)| - |(r \cap s)|$.

- $H(r, s) = 0 \Leftrightarrow r = s$
- $H(r, s) = H(s, r)$

Definition

Definition (Clustering)

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).

from Wikipedia[2]

[2] Wikipedia contributors. *Cluster analysis — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=931629639`. [Online; accessed 17-January-2020]. 2019.

DBSCAN

- Density-based spatial clustering of applications with noise[3]
- Computes a clustering for a collection of points D
- Every point $\in D$ is identified as member of one cluster or noise
- ε: Distance threshold
- *MinPts*: Minimum number of points in ε-neighborhood
- Core points, border points, noise points

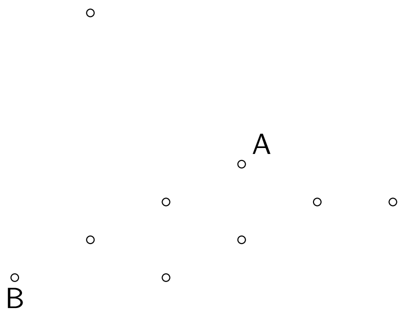
[3] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

DBSCAN Definitions

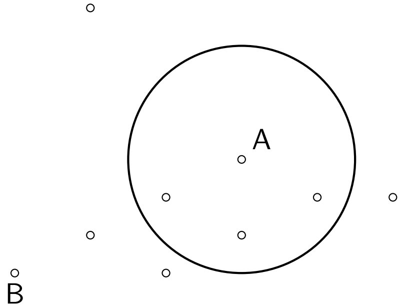
Definition (ε-neighborhood)

The ε-neighborhood of a point q is the set $N_\varepsilon(q)$ of all points $p \in D$ with $H(p, q) \leq \varepsilon$.

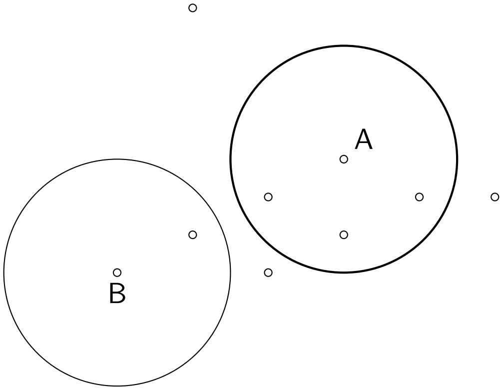
DBSCAN Example



DBSCAN Example



DBSCAN Example



DBSCAN Definitions

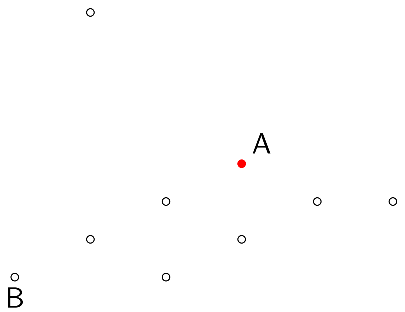
Definition (directly density-reachable)

A point p is *directly density-reachable* from a point q wrt. ε, $MinPts$ if

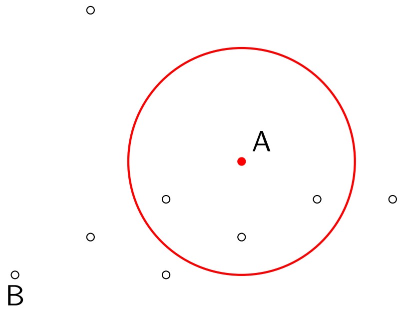
1. $p \in N_\varepsilon(q)$ and
2. $|N_\varepsilon(q)| \geq MinPts$ (core point condition)

Let $MinPts = 3$ for the example.

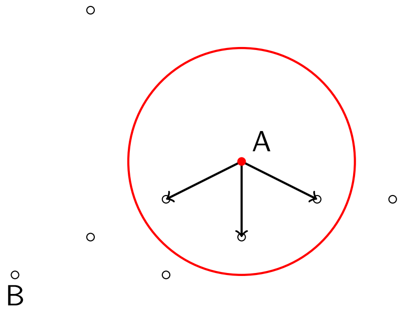
DBSCAN Example



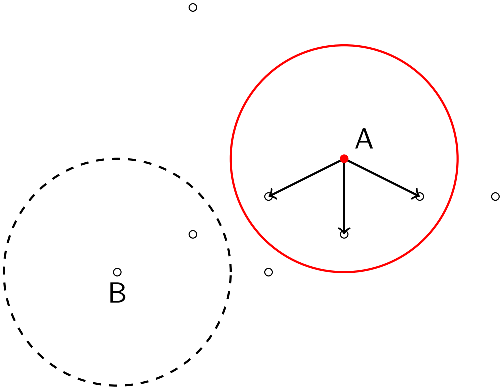
DBSCAN Example



DBSCAN Example



DBSCAN Example

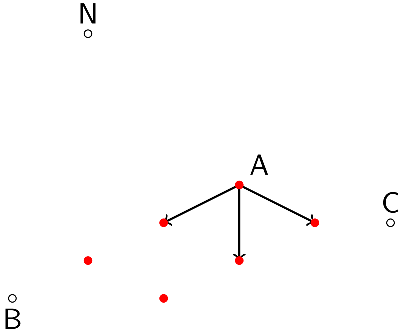


DBSCAN Definitions

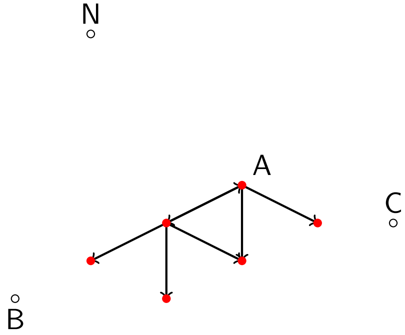
Definition (density-reachable)

A point p is *density-reachable* from a point q wrt. ε, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = q$ such that p_{i+1} is directly density-reachable from p_i.

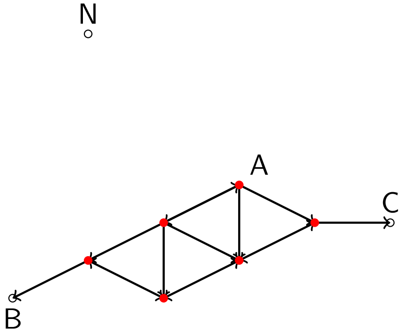
DBSCAN Example



DBSCAN Example



DBSCAN Example

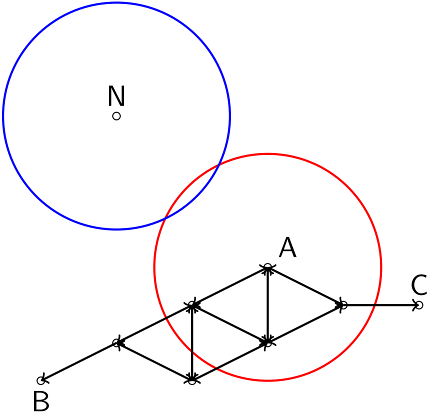


DBSCAN Definitions

Definition (density-connected)

A point p is *density-connected* to a point q wrt. ε, *MinPts* if there is a point o such that both p and q are density-reachable from o.

DBSCAN Example



Very High-Level DBSCAN Algorithm

1. Pick any unvisited point $p \in D$
2. If p is a core point then all density-reachable, unvisited points belong to the same cluster
    1. Find all neighbors of p and set their cluster id
    2. Repeat for all neighboring core points
3. Repeat steps 1 and 2 until all points have been visited
4. Points not belonging to any cluster are noise points

Problems

- Designed for spatial data
- DBSCAN assumes a time complexity of $O(\log n)$ for a region query
  This does not hold for high-dimensional data!

Baseline Approach

1. Compute the neighborhoods of all sets as an AllPairs set similarity join
   Finds all pairs (r, s) with $H(r, s) \leq \varepsilon$
2. Store the result in a data structure with constant time lookup
3. Use the result for the DBSCAN algorithm

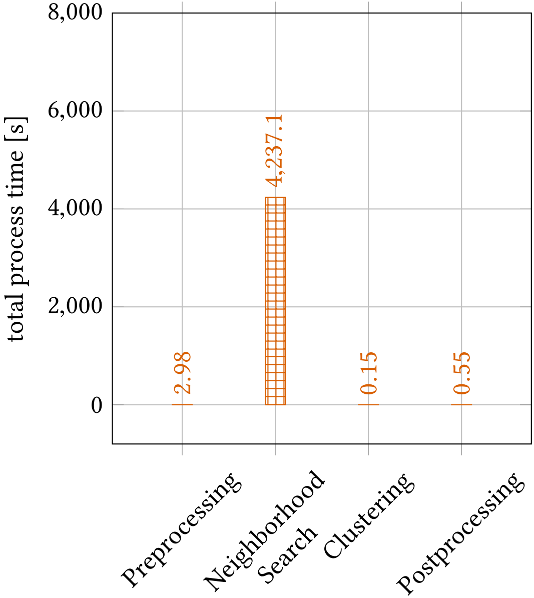
Statistics

	#sets	max set size	avg set size	Dimensionality
AOL	$1.0 \cdot 10^7$	245.0	3.0	$3.9 \cdot 10^6$
NETFLIX	$4.8 \cdot 10^5$	$1.8 \cdot 10^4$	209.5	$1.8 \cdot 10^4$
ORKUT	$2.7 \cdot 10^6$	$4 \cdot 10^4$	119.7	$8.7 \cdot 10^6$
OURS	$9.2 \cdot 10^6$	$6.8 \cdot 10^4$	28.0	$1.2 \cdot 10^4$

Table: Characteristics of datasets from literature[4] and our dataset.

[4] Willi Mann, Nikolaus Augsten, and Panagiotis Bouros. "An Empirical Evaluation of Set Similarity Join Techniques". In: *Proc. VLDB Endow.* 9.9 (May 2016), pp. 636–647. ISSN: 2150-8097. DOI: 10.14778/2947618.2947620. URL: http://dx.doi.org/10.14778/2947618.2947620.

Results

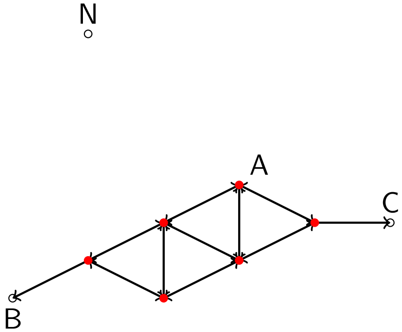


8,000
6,000
4,000
2,000
0
total process time [s]
2.98
4,237.1
0.15
0.55
Preprocessing
Neighborhood Search
Clustering
Postprocessing

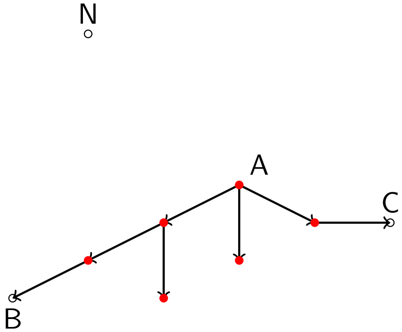
Problems

- Computing neighbors of a given point is expensive
- We compute too many neighborhoods

What is actually computed



What should be computed



Open Questions

- How to compute a set clustering efficiently?
- How can we efficiently identify core points?
  Circular problem: We only need neighbors of core points but to find out if a point is a core point we need its neighbors
- How can we avoid redundant neighborhood computations?

Questions

1. Is the Hamming distance symmetric?

Questions

1. Is the Hamming distance symmetric?
2. How does the high dimensionality of our dataset affect the neighborhood query?

Questions

1. Is the Hamming distance symmetric?
2. How does the high dimensionality of our dataset affect the neighborhood query?
3. Is there a case where DBSCAN is non-deterministic?