- Density-based clustering (DBScan)
  - Reference: Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD 2006

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise

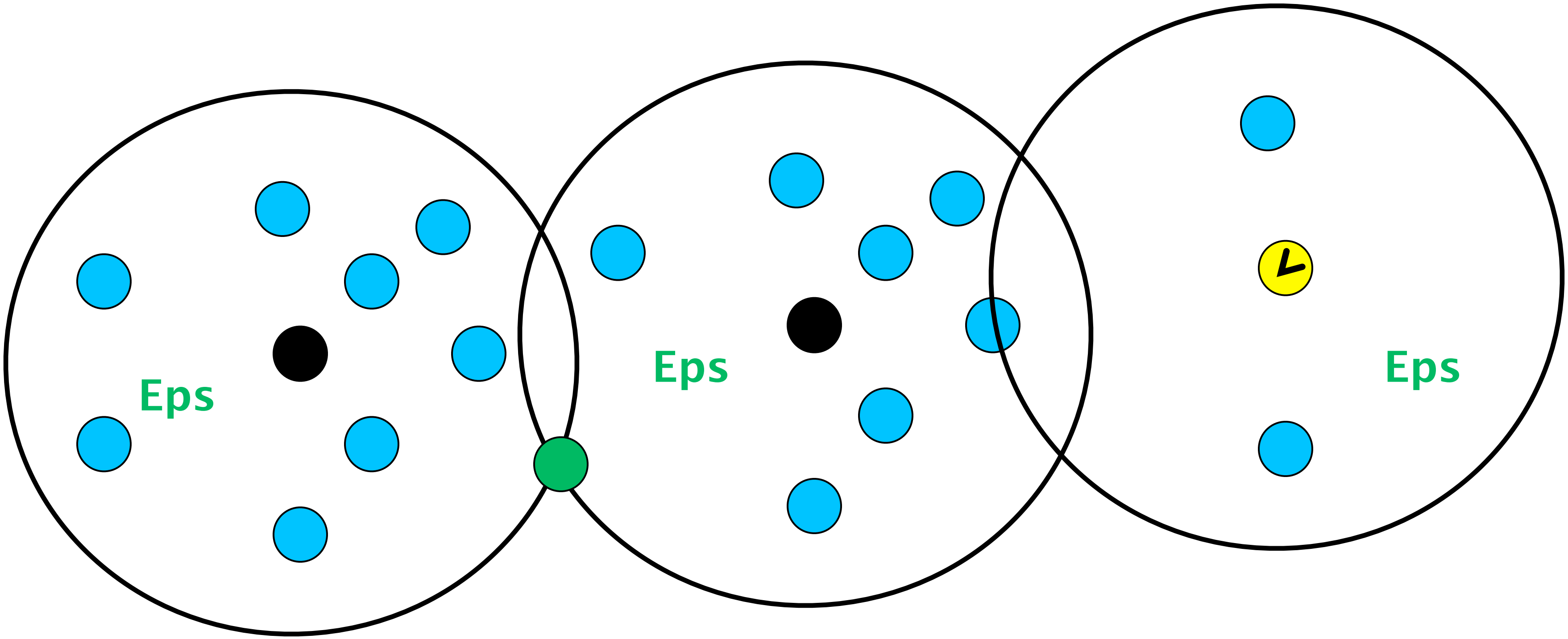
Types of points in density-based clustering

- **Core points**: Interior points of a density-based cluster. A point **p** is a core point if for distance **Eps** :
  - $|N_{Eps}(p) = \{q \mid dist(p,q) <= \varepsilon\}| \geq MinPts$

- **Border points:** Not a core point but within the neighborhood of a core point (it can be in the neighborhoods of many core points)

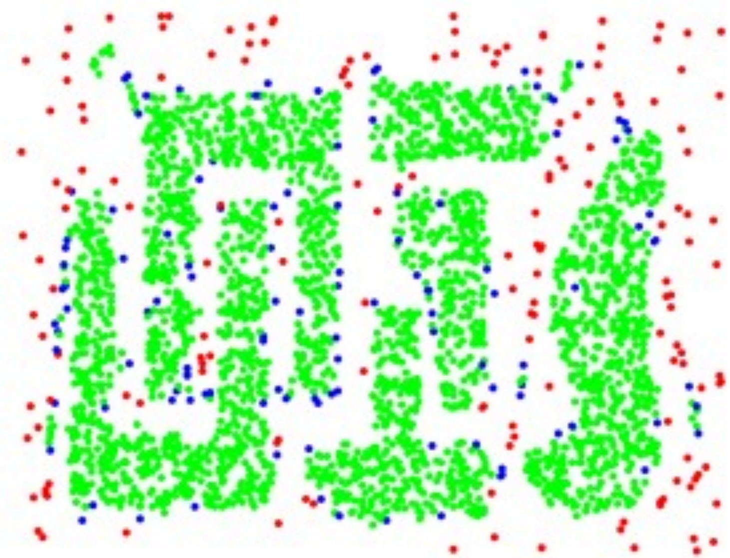
- **Noise points:** Not a core or a border point

Core, border and noise points



Core, Border and Noise points

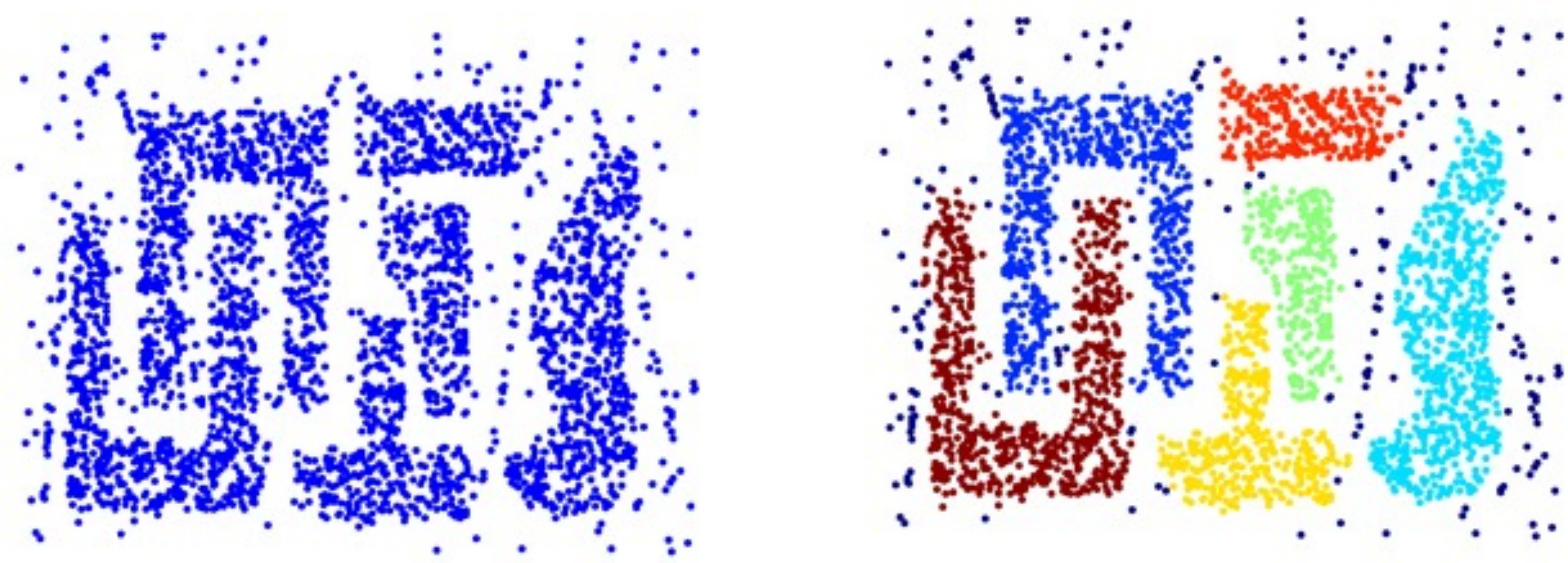
Original Points



Point types: core border and noise

MinPts = 4

Clusters output by DBScan



- Resistant to Noise
- Can handle clusters of different shapes and sizes

Classification of points in density-based clustering

- **Core points**: Interior points of a density-based cluster. A point **p** is a core point if for distance **Eps** :
  - $|N_{Eps}(p)=\{q \mid dist(p,q) <= \varepsilon\}| \geq MinPts$

- **Border points:** Not a core point but within the neighborhood of a core point (it can be in the neighborhoods of many core points)

- **Noise points:** Not a core or a border point

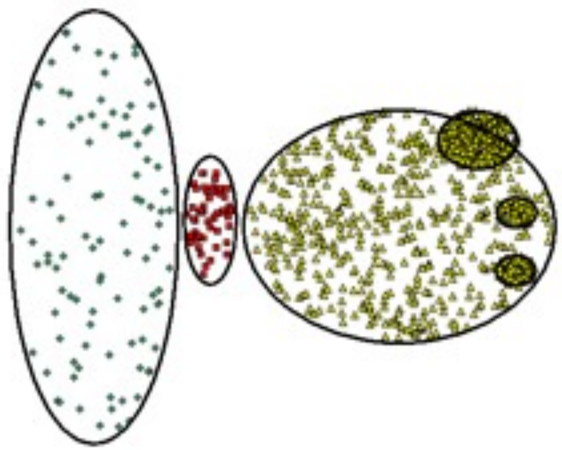
DBSCAN: The Algorithm

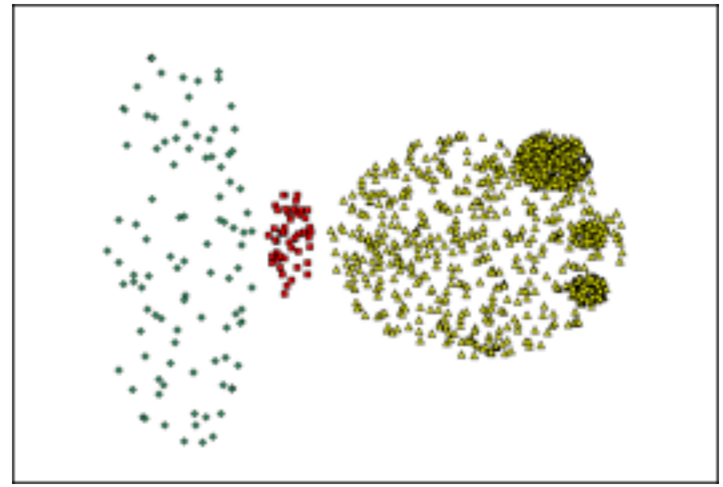
- Label all points as **core**, **border**, or **noise** points
- Eliminate noise points
- Put an edge between all core points that are within **Eps** of each other
- Make each group of connected core points into a separate cluster
- Assign each border point to one of the cluster of its associated core points

Time and space complexity of DBSCAN

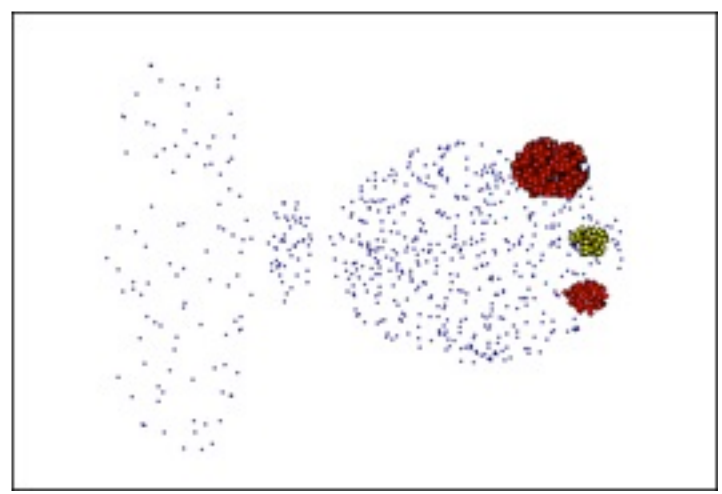
- For a dataset **X** consisting of **n** points, the time complexity of DBSCAN is **O(n x time to find points in the Eps-neighborhood)**
- Worst case **$O(n^2)$**
- In low-dimensional spaces **O(nlogn)**; efficient data structures (e.g., **kd-trees**) allow for efficient retrieval of all points within a given distance of a specified point

When DBSCAN Does NOT Work Well





(MinPts=4, Eps=9.75).

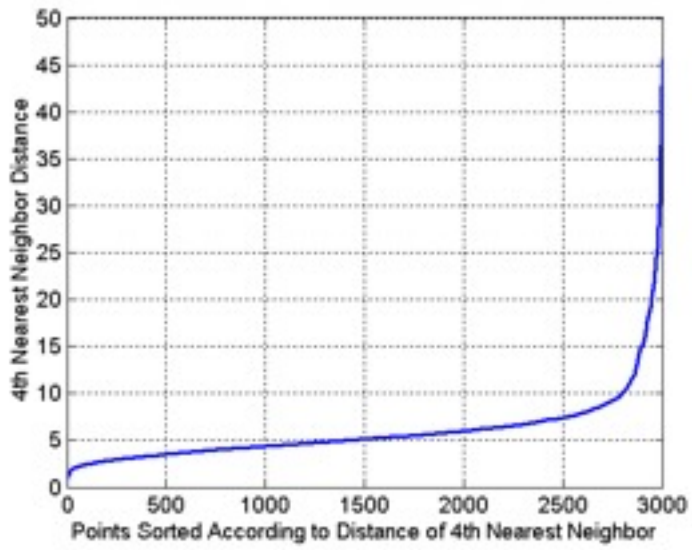


(MinPts=4, Eps=9.92)

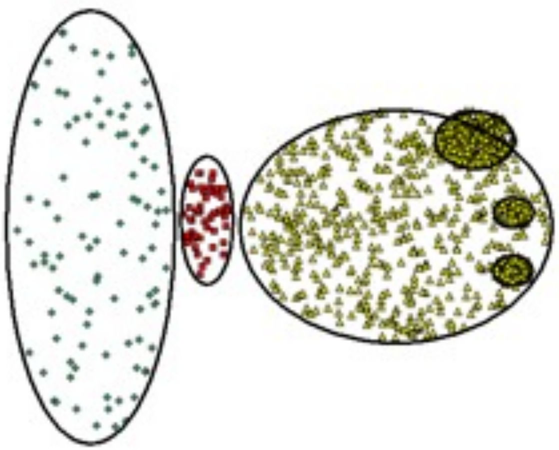
DBScan can fail to identify clusters of varying densities

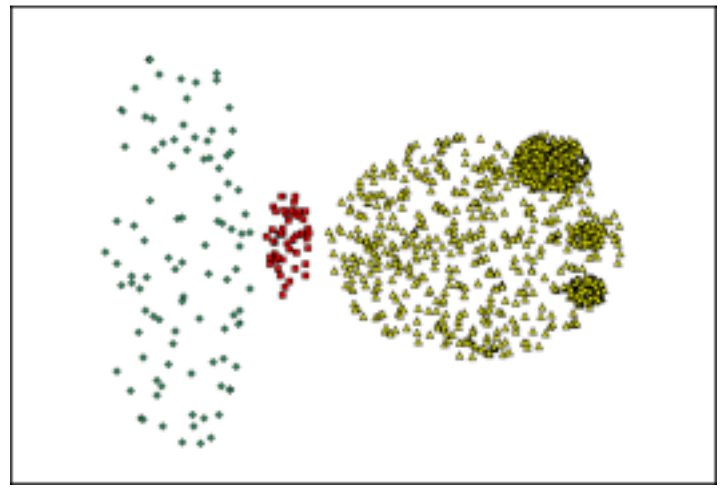
Determining EPS and MinPts

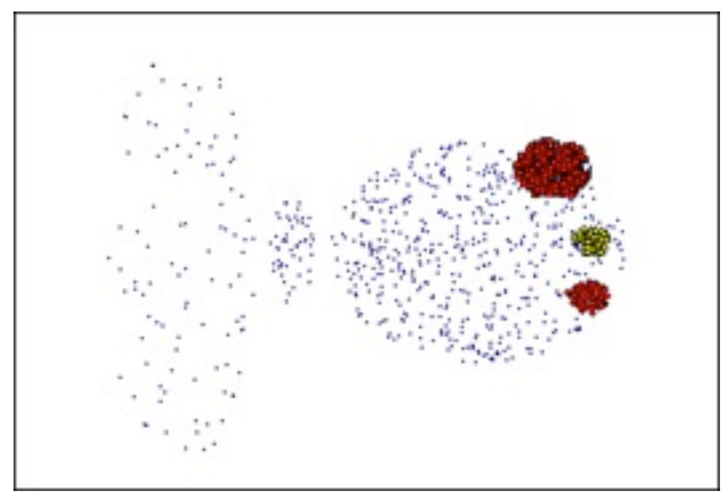
- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



When DBSCAN Does NOT Work Well

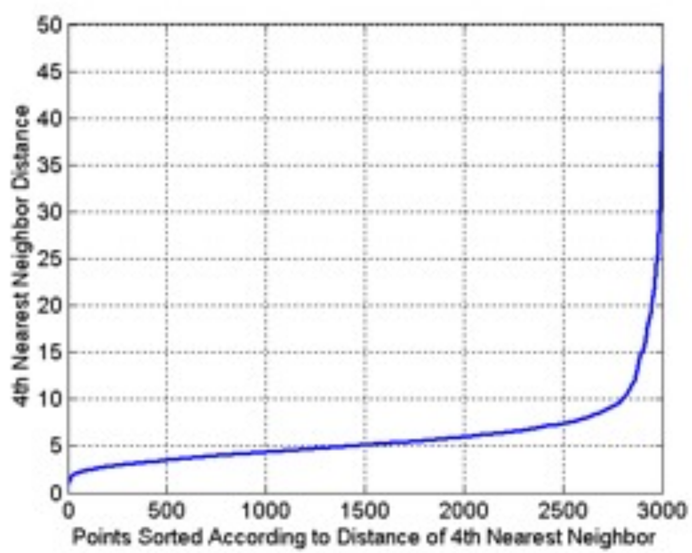






Determining EPS and MinPts

- Idea is that for points in a cluster, their **k**th nearest neighbors are at roughly the same distance
- Noise points have the **k**th nearest neighbor at farther distance
- So, plot sorted distance of every point to its **k**th nearest neighbor



Strengths and weaknesses of DBSCAN

- Resistant to noise
- Finds clusters of arbitrary shapes and sizes
- Difficulty in identifying clusters with varying densities
- Problems in high-dimensional spaces; notion of density unclear
- Can be computationally expensive when the computation of nearest neighbors is expensive