Fundamentals of Big Data Analytics

Lecture 9-10 - Density Based Clustering

Dr. Iqra Safder
Assistant Professor
FAST NUCES, Lahore

Density-based Clustering Approaches

- Why Density-Based Clustering methods?
    - Discover clusters of arbitrary shape.
    - Clusters – Dense regions of objects separated by regions of low density
- DBSCAN – Density Based Spatial Clustering of the Application with Noise
    - the first density based clustering
- OPTICS – density based cluster-ordering

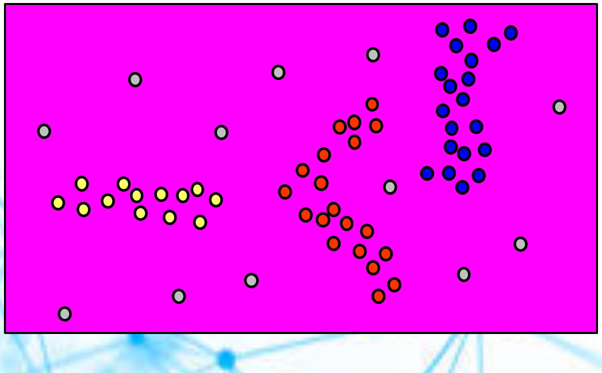
DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Proposed by Ester, Kriegel, Sander, and Xu (KDD96)
- Relies on a density-based notion of cluster: A cluster is defined as a maximal set of density-connected points.
- Discovers clusters of arbitrary shape in spatial databases with noise

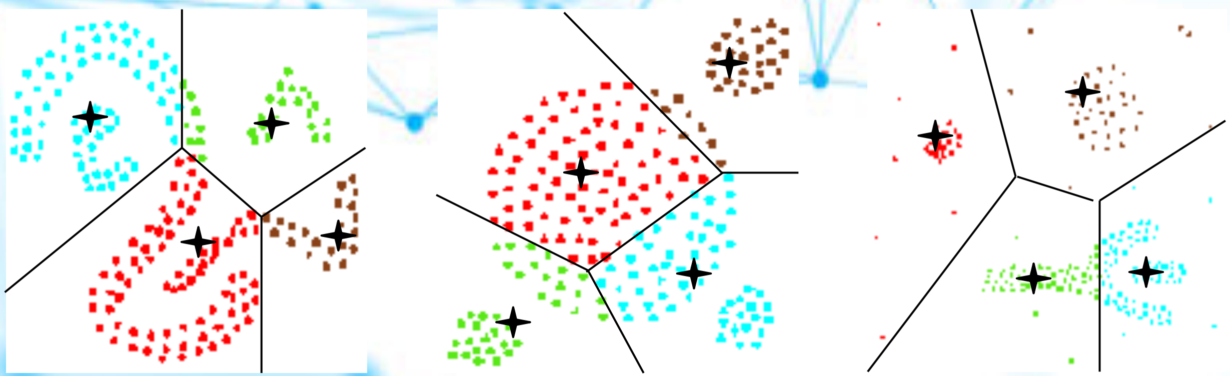
Density-Based Clustering

***Basic Idea*:**

Clusters are dense regions in the data space, separated by regions of lower object density



Why Density-Based Clustering?



Results of a *k*-medoid algorithm for *k*=4

The ***DBSCAN algorithm*** is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

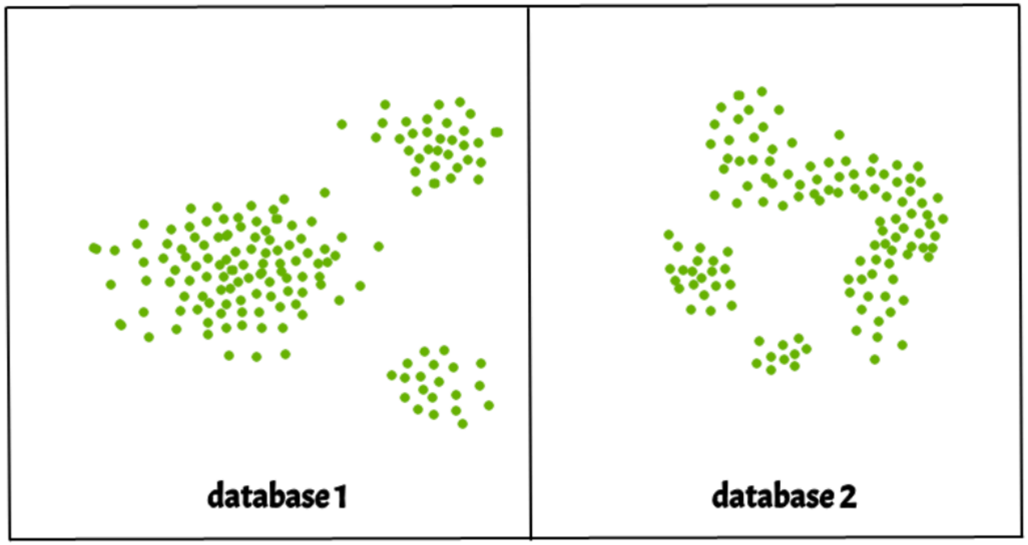
Density-Based Clustering

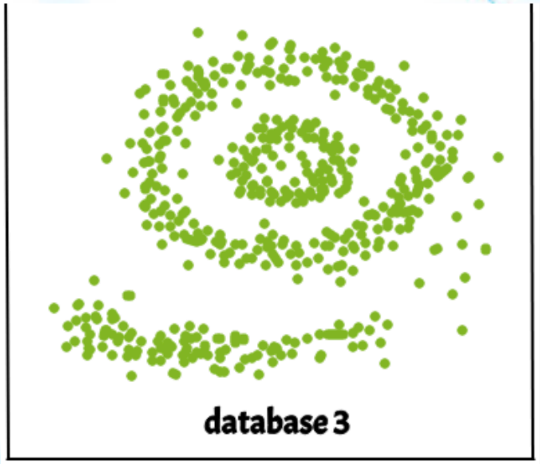
Why Density-Based Clustering?

Partitioning methods (K-means) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.

Real life data may contain irregularities, like:

1. Clusters can be of arbitrary shape such as those shown in the figure below.
2. Data may contain noise.





Density Based Clustering: Basic Concept

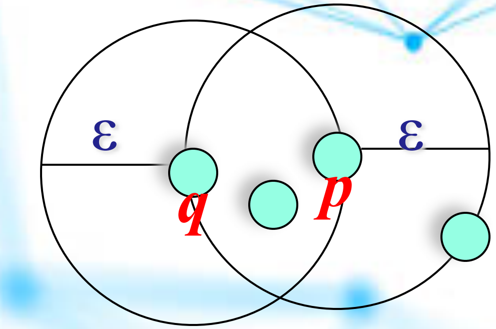
- Intuition for the formalization of the basic idea
  - For any point in a cluster, the local point density around that point has to exceed some threshold
  - The set of points from one cluster is spatially connected
- Local point density at a point *p* defined by two parameters
  - ε – radius for the neighborhood of point p:
    $N_{\varepsilon}(p) := \{q \text{ in data set } D \mid dist(p, q) \leq \varepsilon\}$
  - *MinPts* – minimum number of points in the given neighbourhood *N*(*p*)

ε-Neighborhood

- ε-Neighborhood – Objects within a radius of ε from an object.

$$N_{\varepsilon}(p):\{q \mid d(p,q) \leq \varepsilon\}$$

- “High density” - ε-Neighborhood of an object contains at least *MinPts* of objects.



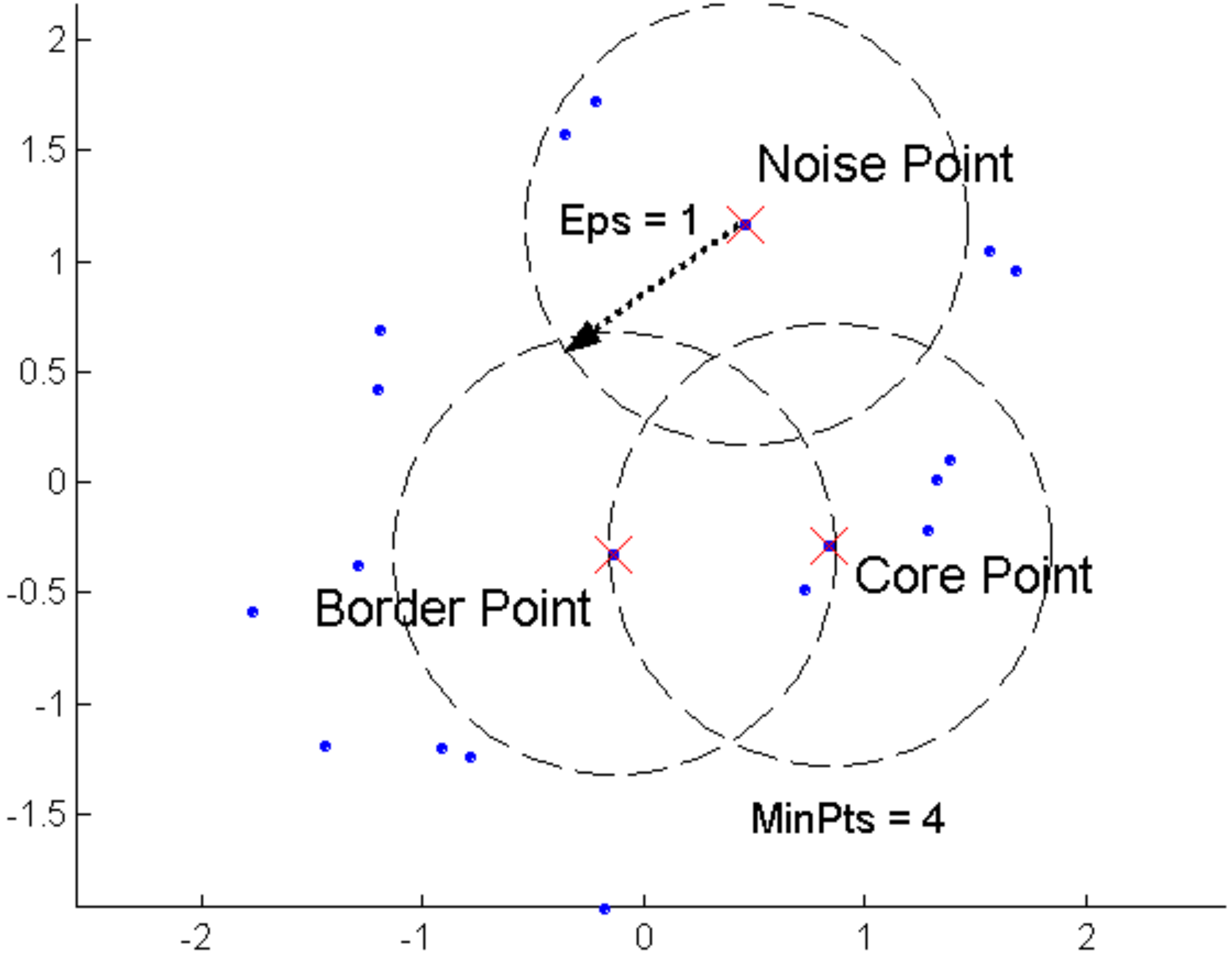
ε-Neighborhood of *p*

ε-Neighborhood of *q*

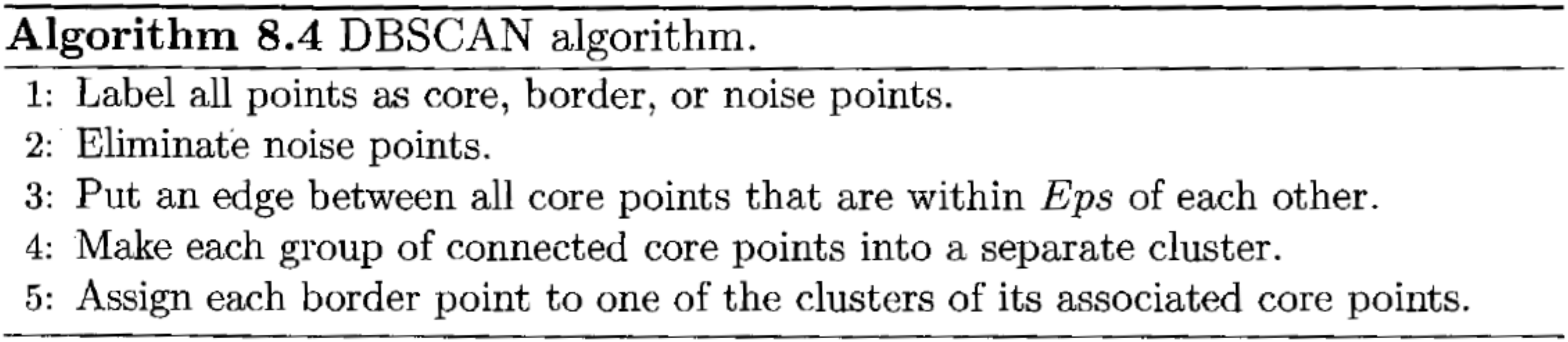
DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points



DBSCAN Algorithm



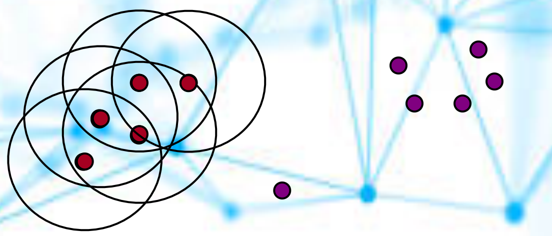
Algorithm 8.4 DBSCAN algorithm.

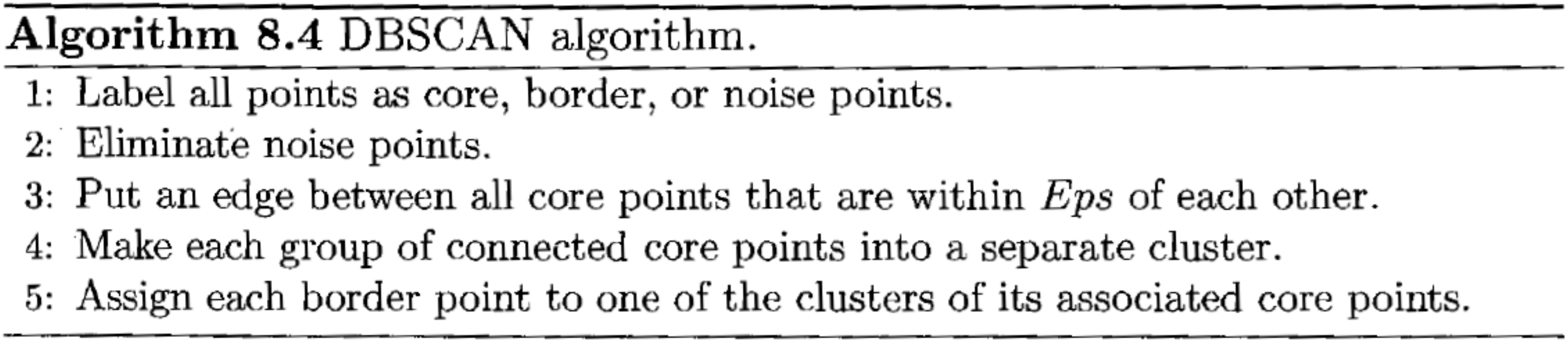
```
Label all points as core, border, or noise points.
Eliminate noise points.
Put an edge between all core points that are within $Eps$ of each other.
Make each group of connected core points into a separate cluster.
Assign each border point to one of the clusters of its associated core points.
```

DBSCAN Algorithm: Example

Parameter

- ε = 2 cm
- *MinPts* = 3



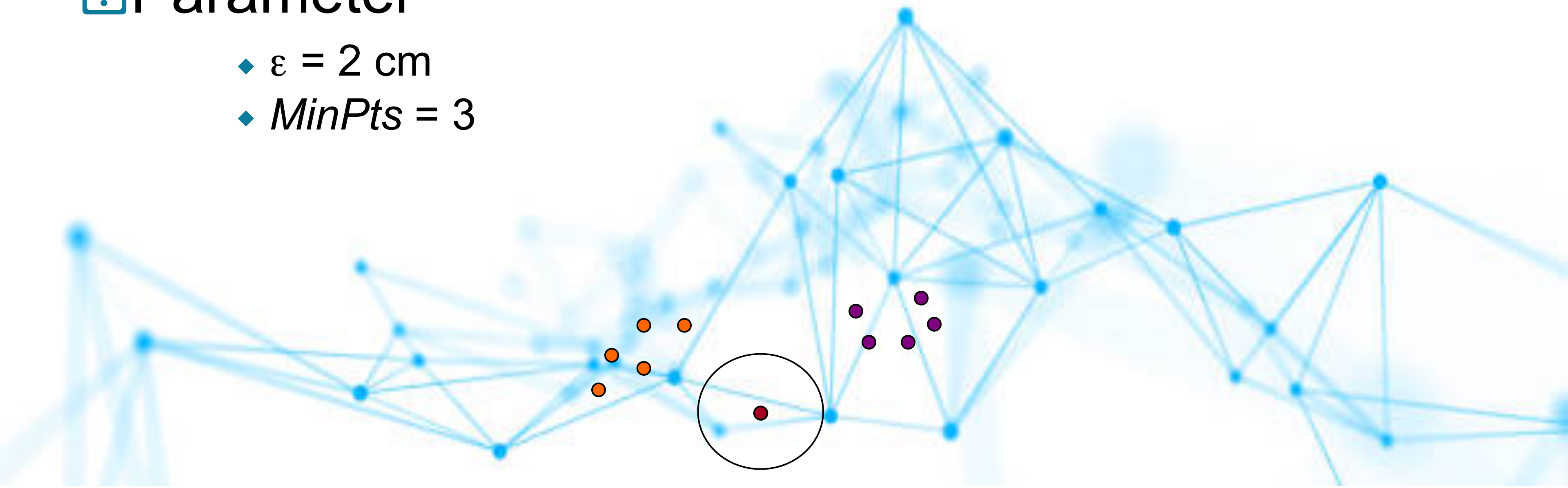


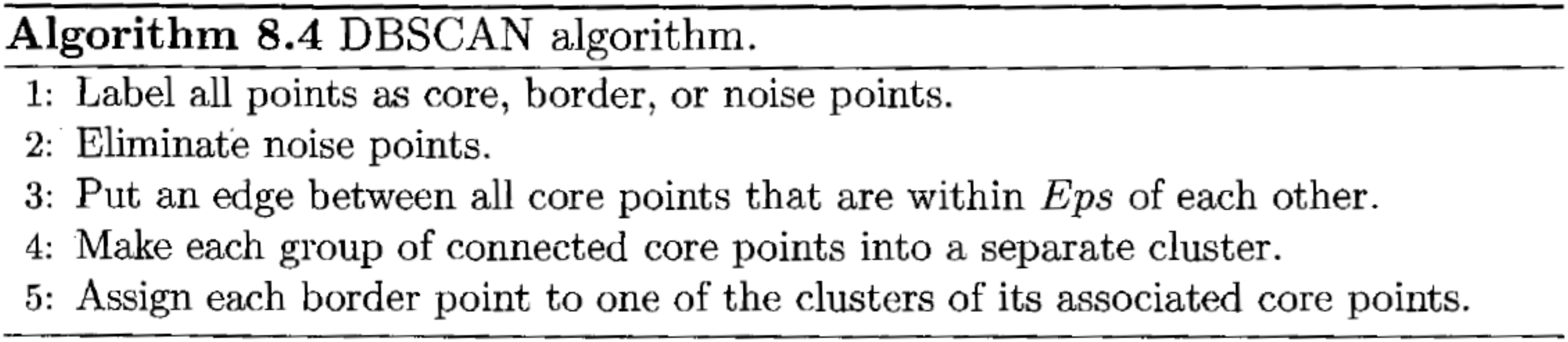
Algorithm 8.4 DBSCAN algorithm.

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

DBSCAN Algorithm: Example

- Parameter
  - ε = 2 cm
  - *MinPts* = 3





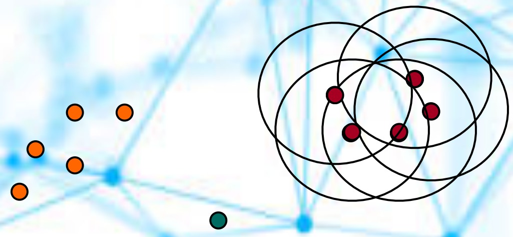
Algorithm 8.4 DBSCAN algorithm.

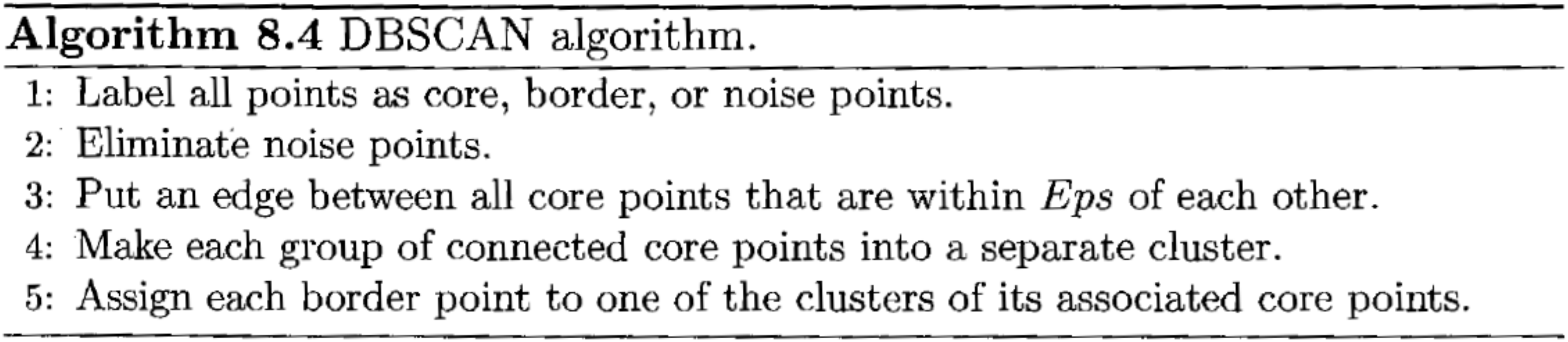
```
1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within $Eps$ of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.
```

DBSCAN Algorithm: Example

Parameter

- ε = 2 cm
- *MinPts* = 3



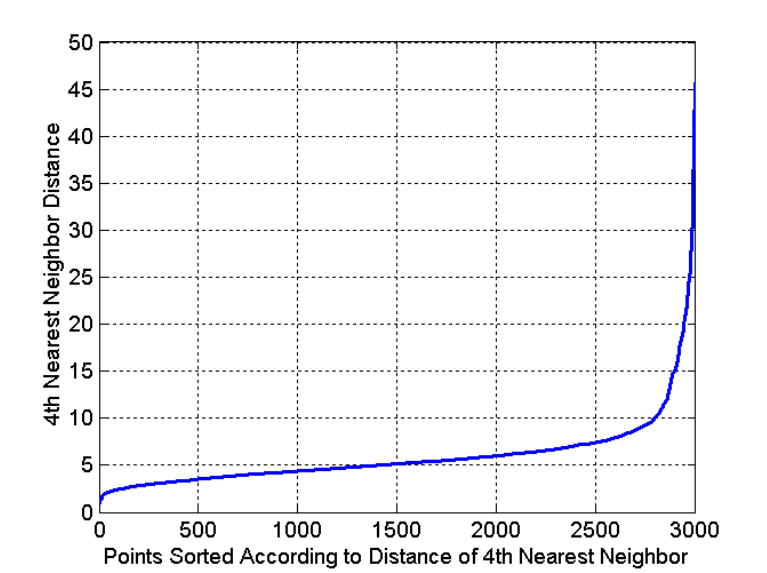


Algorithm 8.4 DBSCAN algorithm.

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

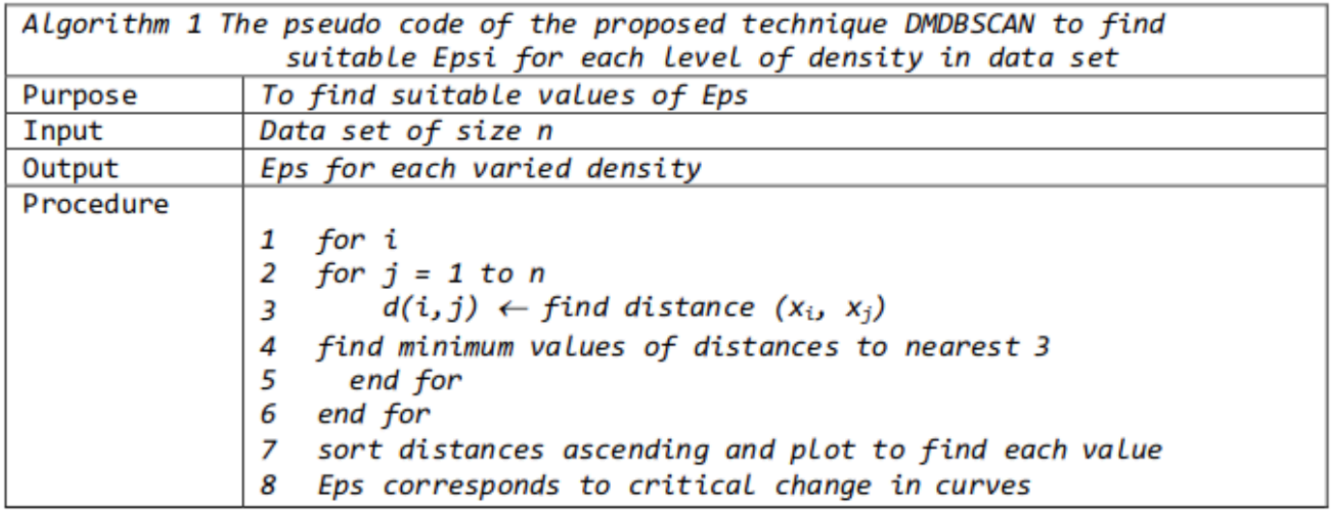
DBSCAN: Determining EPS and MinPts

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



- http://www.sefidian.com/2020/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/#:~:text=In%20layman's%20terms%2C%20we%20find,and%20select%20that%20as%20epsilon.

- https://iopscience.iop.org/article/10.1088/1755-1315/31/1/012012/pdf
- https://scikit-learn.org/stable/auto_examples/cluster/plot_dbscan.html



Algorithm 1 The pseudo code of the proposed technique DMDBSCAN to find suitable Epsi for each level of density in data set

Purpose	*To find suitable values of Eps*
Input	*Data set of size n*
Output	*Eps for each varied density*

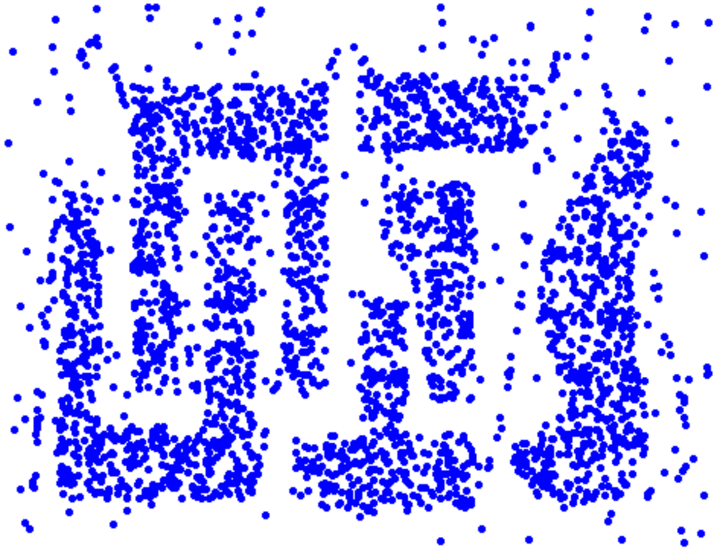
Procedure

```
for i
for j = 1 to n
    d(i,j) ← find distance ($x_i$, $x_j$)
find minimum values of distances to nearest 3
  end for
end for
sort distances ascending and plot to find each value
Eps corresponds to critical change in curves
```

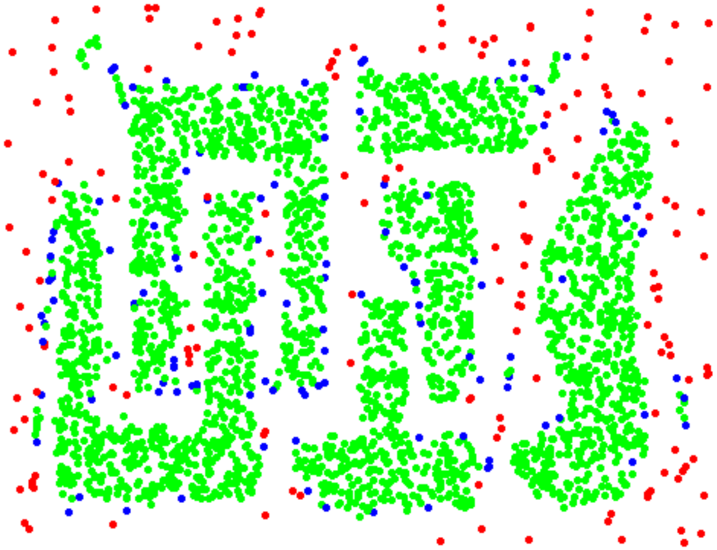
Figure 1 Pseudocode DMDBSCAN Algorithm (Elbatta 2012)

Algorithm to find Eps value in DBscan

DBSCAN: Core, Border and Noise Points



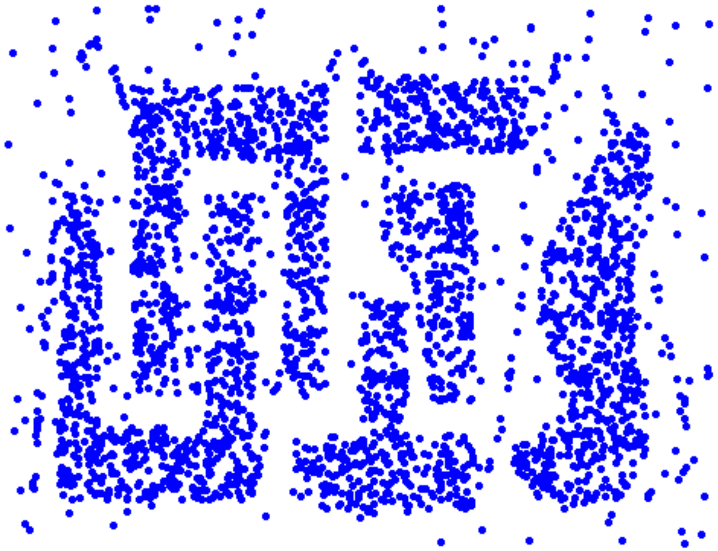
Original Points



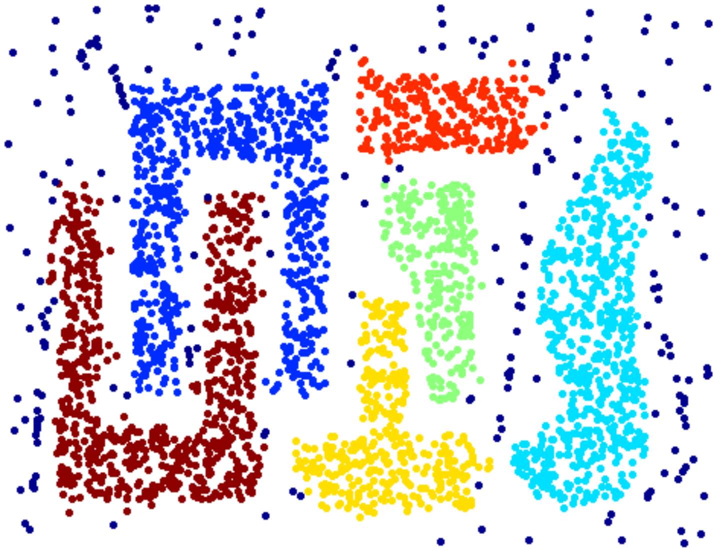
Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well



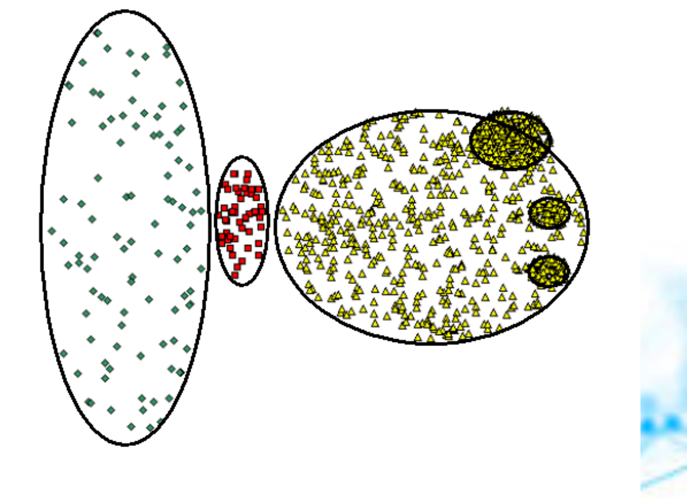
Original Points



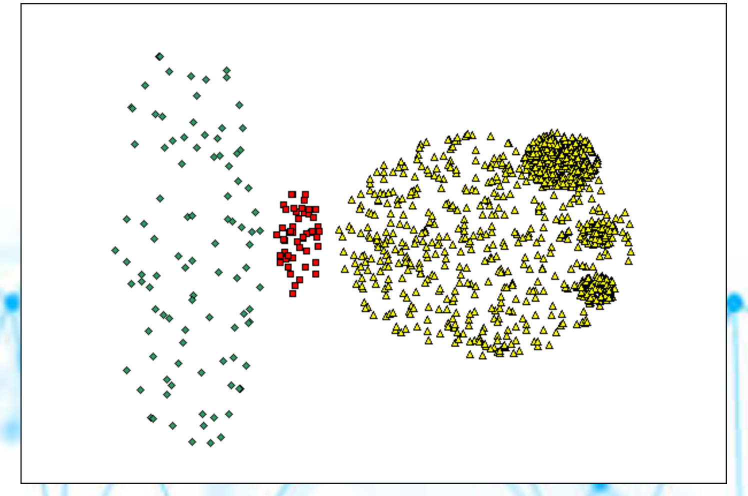
Clusters

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

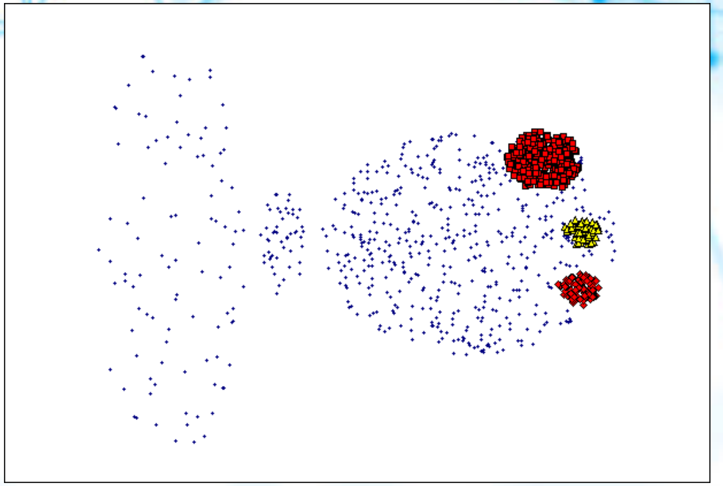
When DBSCAN Does NOT Work Well



Original Points



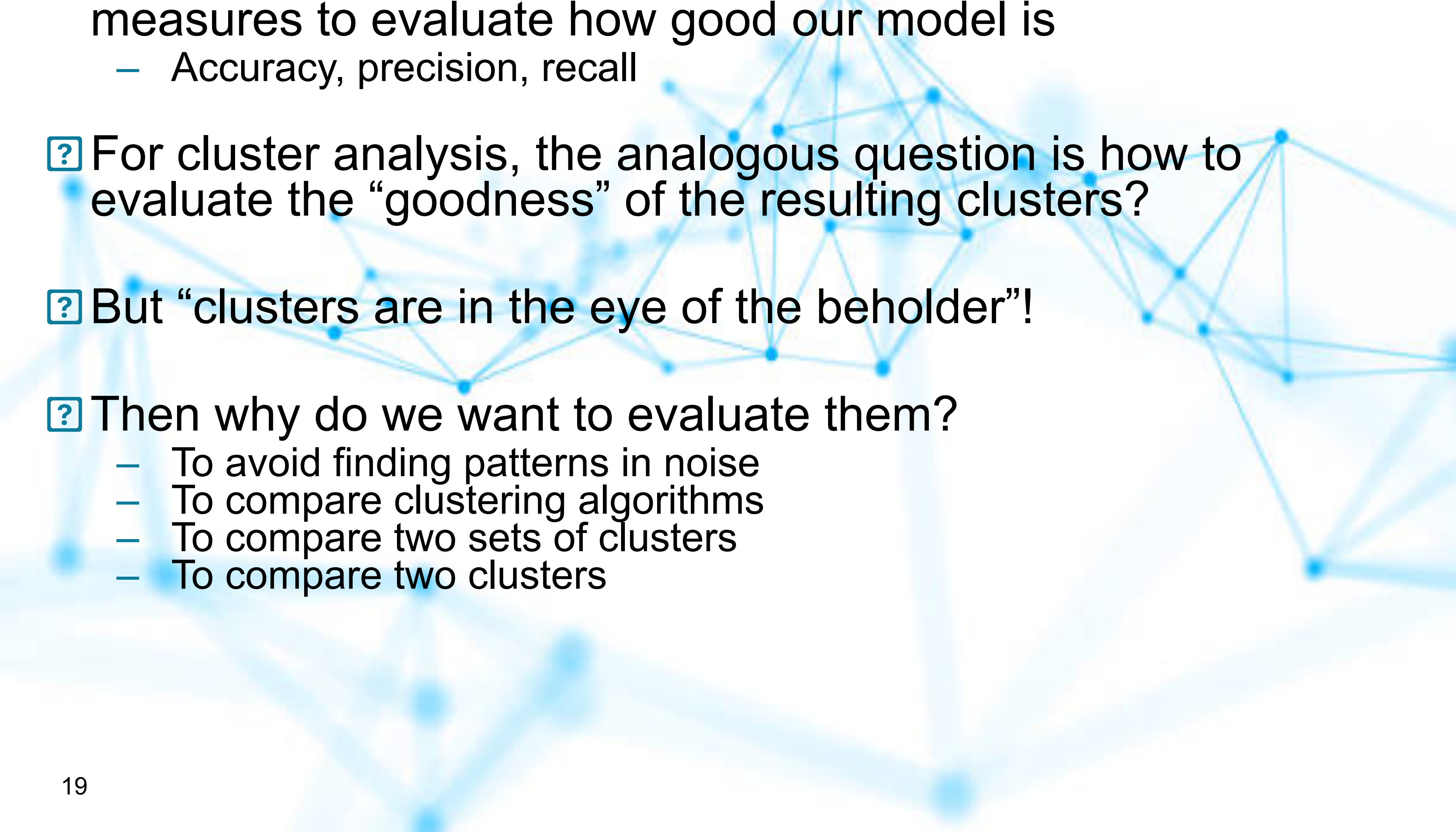
(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data

Cluster Validity



- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall

- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- But “clusters are in the eye of the beholder”!

- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as criteria instead of indices

What is a Good Clustering?

Internal criterion: A good clustering will produce high quality clusters in which:

the intra-class (that is, intra-cluster) similarity is high

the inter-class similarity is low

The measured quality of a clustering depends on both the document representation and the similarity measure used

External criteria for clustering quality

Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data

Assesses a clustering with respect to ground truth … requires *labeled data*

Assume documents with C gold standard classes, while our clustering algorithms produce K clusters, ω_1, ω_2, …, ω_K with n_i members.

External Evaluation of Cluster Quality

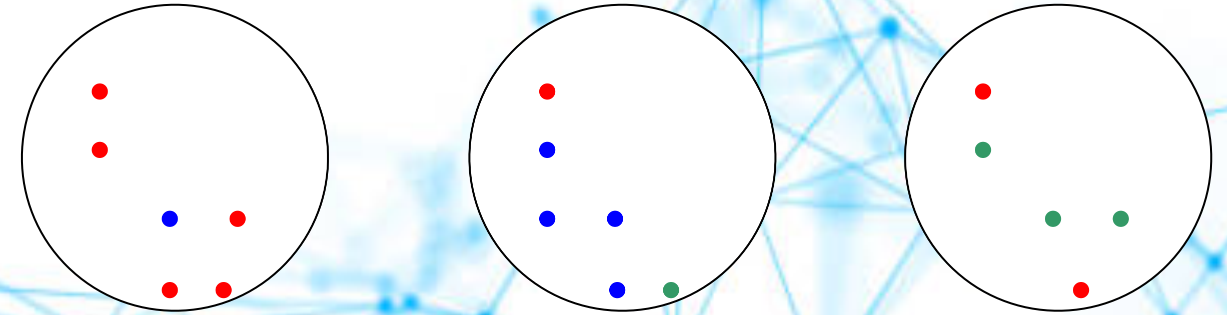
Simple measure: purity, the ratio between the dominant class in the cluster ω_i and the size of cluster ω_i

$$Purity(\omega_i) = \frac{1}{n_i}\max_j (n_{ij}) \quad j \in C$$

Biased because having *n* clusters maximizes purity

Others are entropy of classes in clusters (or mutual information between classes and clusters)

Purity example



Cluster I

Cluster II

Cluster III

Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6

Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5