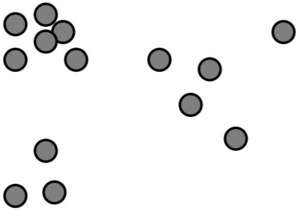
Overview

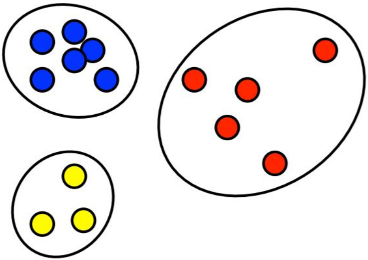
What is clustering?

- Given data objects



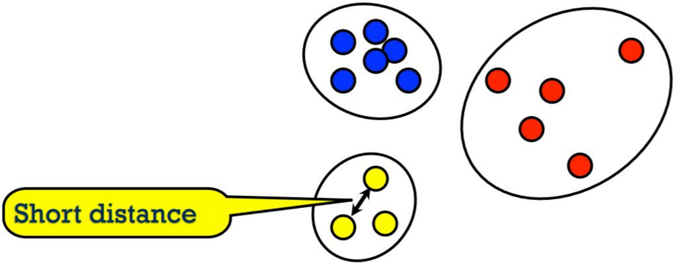
What is clustering?

- Given data objects
- Find a grouping (**clustering**) such that the objects are:



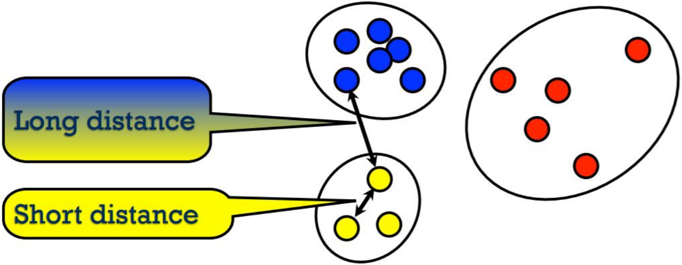
What is clustering?

- Given data objects
- Find a grouping (**clustering**) such that the objects are:
  - similar (related) to the objects in the same group



What is clustering?

- Given data objects
- Find a grouping (**clustering**) such that the objects are:
  - similar (related) to the objects in the same group
  - dissimilar (unrelated) from objects in other groups



Why to cluster data?

- Intuition building

Why to cluster data?

- Intuition building
- Hypothesis generation

Why to cluster data?

- Intuition building
- Hypothesis generation
- Discover structures and patterns in high-dimensional data

Why to cluster data?

- Intuition building
- Hypothesis generation
- Discover structures and patterns in high-dimensional data
- Summarizing / compressing large data

Clustering is subjective

- Suppose that we need to put you in some groups for projects

Clustering is subjective

- Suppose that we need to put you in some groups for projects
- How to define those groups?
  - Work experience
  - Age
  - Education
  - Preferences

Clustering is subjective

- Suppose that we need to put you in some groups for projects
- How to define those groups?
  - Work experience
  - Age
  - Education
  - Preferences
- Which way is correct? Depends on the goal!

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation
- Search result grouping

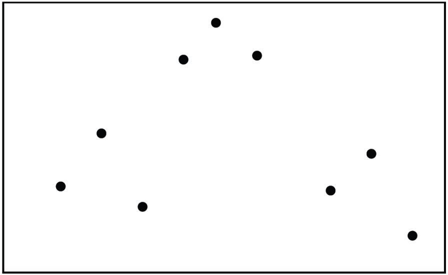
Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation
- Search result grouping
- Social network analysis

Examples

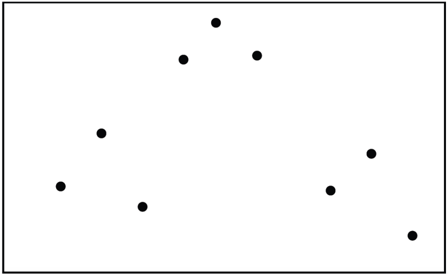
- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation
- Search result grouping
- Social network analysis
- Anomaly detection

Clustering



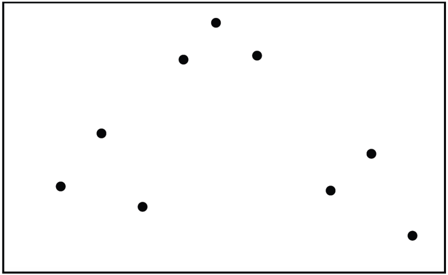
- Assume the data $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ lives in a Euclidean space, $\mathbf{x}^{(n)} \in \mathbb{R}^d$.

Clustering



- Assume the data $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ lives in a Euclidean space, $\mathbf{x}^{(n)} \in \mathbb{R}^d$.
- Assume the data belongs to K classes (patterns)

Clustering



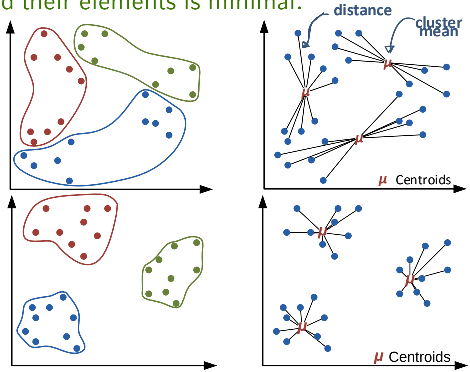
- Assume the data $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ lives in a Euclidean space, $\mathbf{x}^{(n)} \in \mathbb{R}^d$.
- Assume the data belongs to K classes (patterns)
- How can we identify those classes (data points that belong to each class)?

Idea of K-means: find a clustering such that the *within-cluster variation* of each cluster is small and use the *centroid* of a cluster as representative.

Objective: For a given k, form k groups so that the sum of the (squared) distances between the mean of the groups and their elements is minimal.

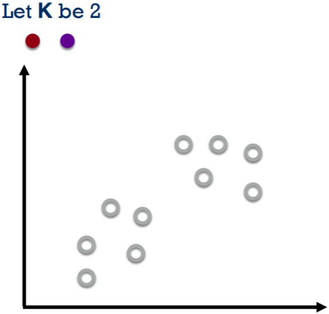
Poor Clustering
(large sum of distances)

Optimal Clustering
(minimal sum of distances)



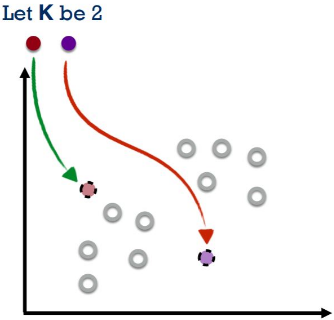
K-means Clustering

1. Choose K, the number of potential clusters



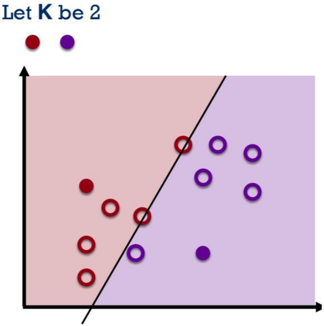
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data



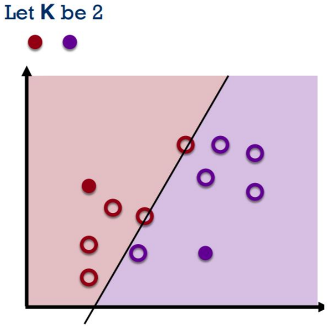
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre



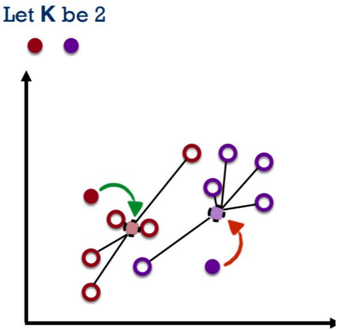
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre



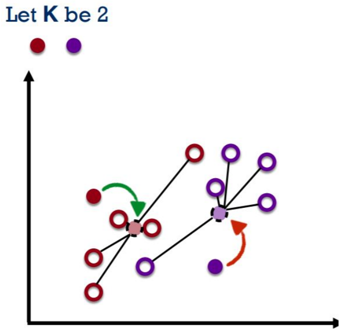
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers



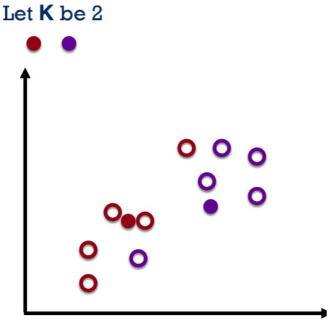
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers



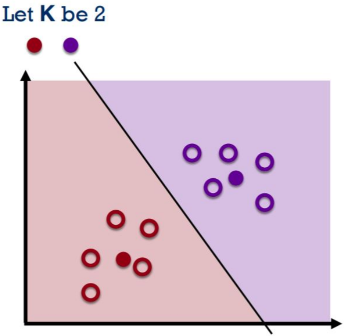
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers
5. Steps 3 and 4 are repeated until convergence



K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers
5. Steps 3 and 4 are repeated until convergence



K-means for Vector Quantization

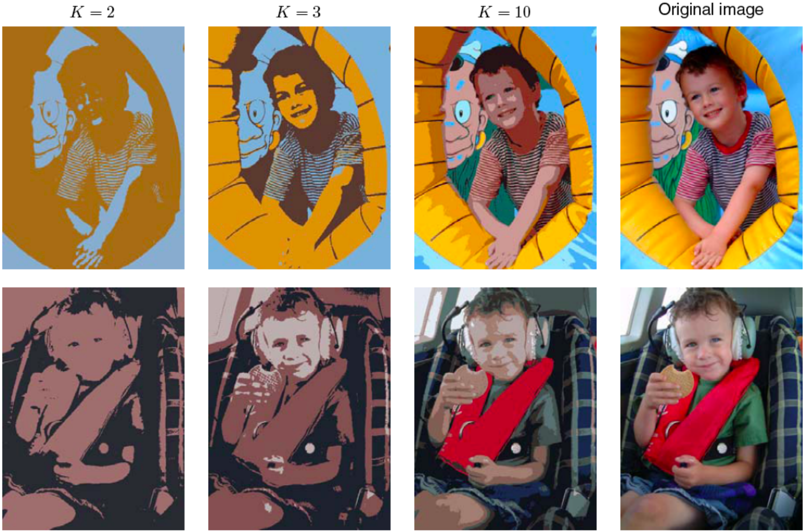


Figure from Bishop

How to choose K in K-means?

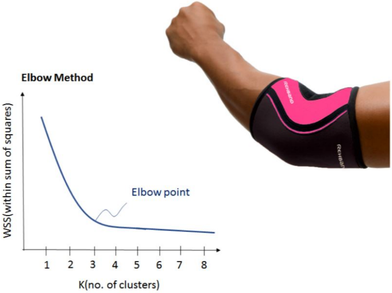
- **Elbow method:** increase K until it does not help to describe data better

How to choose K in K-means?

- **Elbow method:** increase K until it does not help to describe data better
- We are interested in finding K such that the sum of within-group Euclidean distances is smaller

$$J = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \|\mathbf{x}_i^{(j)} - \mathbf{c}_j\|^2,$$

where $\mathbf{c}_j$ is the centroid (mean) of the j^{th} cluster



Why K-means Converges

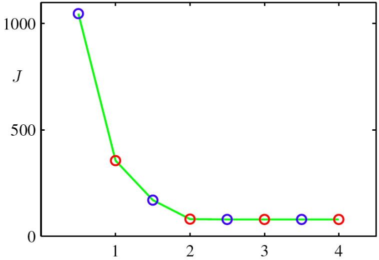
- Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.

Why K-means Converges

- Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.
- Whenever a cluster center is moved, J is reduced.

Why K-means Converges

- Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.
- Whenever a cluster center is moved, J is reduced.
- Test for convergence: If the assignments do not change in the assignment step, we have converged (to at least a local minimum).

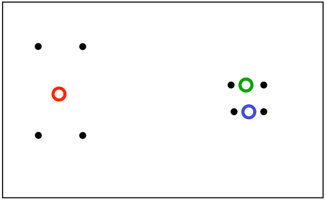


- K-means cost function after each E step (blue) and M step (red). The algorithm has converged after the third M step

Local Minima

- The objective J is non-convex (so coordinate descent on J is not guaranteed to converge to the global minimum)
- There is nothing to prevent k-means getting stuck at local minima.
- We could try many random starting points
- We could try non-local split-and-merge moves:
  - Simultaneously merge two nearby clusters
  - and split a big cluster into two

A bad local optimum



Summary of K-means

Pro: Simple, easy to implement

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Con: The dissimilarity measure is fixed to Euclidean distance and features should be quantitative (numeric)

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Con: The dissimilarity measure is fixed to Euclidean distance and features should be quantitative (numeric)

Con: Squared Euclidean distance places the highest influence on the largest distances, therefore this approach is sensitive to outliers in the data

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Con: The dissimilarity measure is fixed to Euclidean distance and features should be quantitative (numeric)

Con: Squared Euclidean distance places the highest influence on the largest distances, therefore this approach is sensitive to outliers in the data

Con: In K Means clustering, the results produced by running the algorithm multiple times might differ because of the random initialization of the centroids. While results are reproducible in Hierarchical clustering.

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure
4. Medoids of each of the K clusters are updated, taking the ones that are closer to all other points in the cluster

K-medoids (Partitioning Around Medoids) clustering

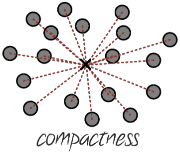
1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure
4. Medoids of each of the K clusters are updated, taking the ones that are closer to all other points in the cluster

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure
4. Medoids of each of the K clusters are updated, taking the ones that are closer to all other points in the cluster
5. Steps 3 and 4 are repeated until convergence

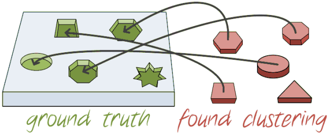
- Evaluation based on **expert's** opinion
  - \+ may reveal new insight into the data
  - – very expensive, results are not comparable

- Evaluation based on **internal** measures
  - \+ no additional information needed
  - – approaches optimizing the evaluation criteria will always be preferred

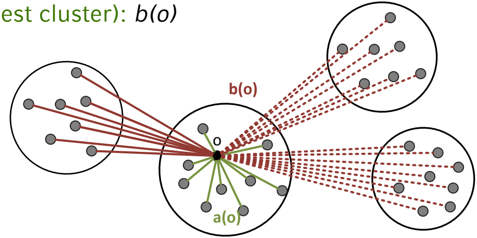


- Evaluation based on **external** measures
  - \+ objective evaluation
  - – needs „ground truth"

  e.g., comparison of two clusterings



- Basic idea:
  - How good is the clustering = how appropriate is the mapping of objects to clusters
  - Elements in cluster should be „similar" to their representative
    → measure the average distance of objects to their representative: *a(o)*
  - Elements in different clusters should be „dissimilar"
    → measure the average distance of objects to alternative clusters (i.e. second closest cluster): *b(o)*

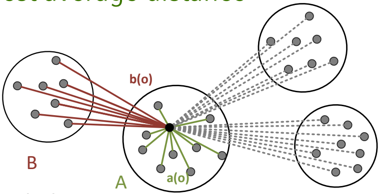


- $a(o)$: average distance between object o and the objects in its cluster A

$$a(o) = \frac{1}{|C(o)|} \sum_{p \in C(o)} dist(o,p)$$

- $b(o)$: for each other cluster C_i compute the average distance between o and the objects in C_i. Then take the smallest average distance

$$b(o) = \min_{C_i \neq C(o)} \left(\frac{1}{|C_i|} \sum_{p \in C_i} dist(o,p) \right)$$



- The silhouette of o is then defined as

$$s(o) = \begin{cases} 0 & if\ a(o) = 0, e.g. |C_i| = 1 \\ \dfrac{b(o) - a(o)}{\max\{a(o), b(o)\}} & else \end{cases}$$

- The values of the silhouette coefficient range from −1 to +1

- The silhouette of a cluster C_i is defined as:

$$silh(C_i) = \frac{1}{|C_i|} \sum_{o \in C_i} s(o)$$

- The silhouette of a clustering $\mathcal{C} = (C_1, \dots, C_k)$ is defined as:

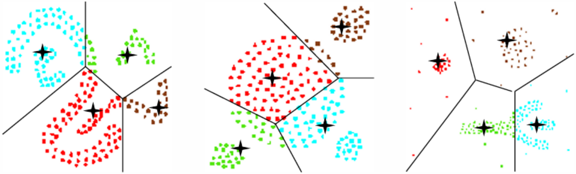
$$silh(\mathcal{C}) = \frac{1}{|D|} \sum_{o \in D} s(o),$$

where D denotes the whole dataset.

- „Reading" the silhouette coefficient:
  Let $a(o) \neq 0$.
  - $b(o) \gg a(o) \Rightarrow s(o) \approx 1$: good assignment of o to its cluster A
  - $b(o) \approx a(o) \Rightarrow s(o) \approx 0$: o is in-between A and B
  - $b(o) \ll a(o) \Rightarrow s(o) \approx -1$: bad, on average o is closer to members of B

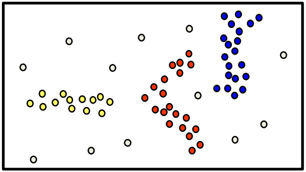
- Silhouette Coefficient $s_{\mathcal{C}}$ of a clustering: average silhouette of all objects
  - $0.7 < s_C \leq 1.0$ strong structure, $0.5 < s_C \leq 0.7$ medium structure
  - $0.25 < s_C \leq 0.5$ weak structure, $s_C \leq 0.25$ no structure

What to do if K-Means/K-Medoids fail?



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

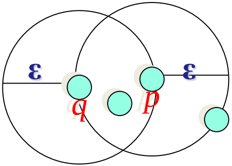
- **Basic idea**
  - Clusters are dense regions in the data space, separated by regions of lower object density
  - A cluster is defined as a maximal set of density-connected points
  - Discovers clusters of arbitrary shape
- **Method**
  - DBSCAN



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

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

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

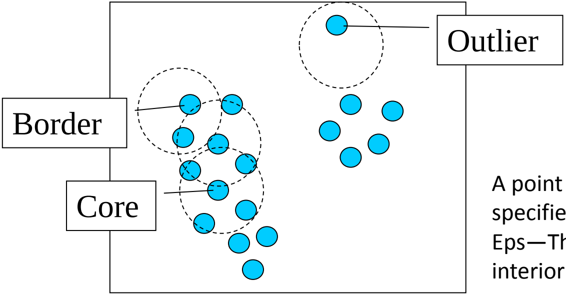


ε-Neighborhood of *p*

ε-Neighborhood of *q*

Density of p is "high" (MinPts = 4)

Density of q is "low" (MinPts = 4)



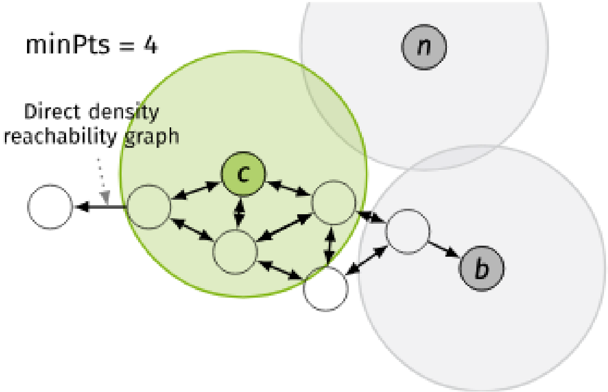
Given ε and *MinPts*, categorize the objects into three exclusive groups.

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.

ε = 1unit, MinPts = 5

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 nor a border point.

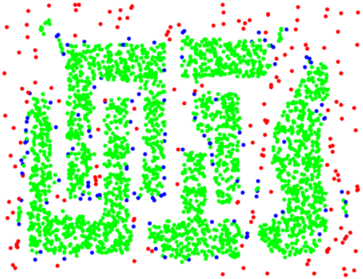


n is not core.

c is a core point.

b is not core.

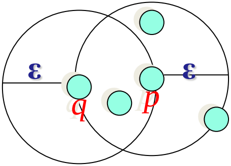
Original Points



Point types: core, border and outliers

ε = 10, MinPts = 4

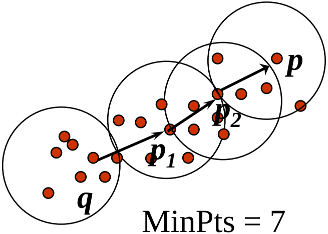
- Directly density-reachable
  - An object *q* is directly density-reachable from object *p* if *p* is a core object and *q* is in p's ε-neighborhood.



MinPts = 4

- *q* is directly density-reachable from *p*
- *p* is not directly density-reachable from *q*
- Density-reachability is asymmetric

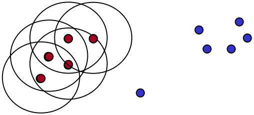
- Density-Reachable (directly and indirectly):
  - A point p is directly density-reachable from p_2
  - p_2 is directly density-reachable from p_1
  - p_1 is directly density-reachable from q
  - $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$ form a chain



MinPts = 7

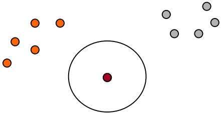
- p is (indirectly) density-reachable from q
- q is not density-reachable from p

- Parameter
  - $\varepsilon = 2.0$
  - $MinPts = 3$



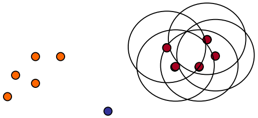
for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

- Parameter
  - $\varepsilon = 2.0$
  - $MinPts = 3$



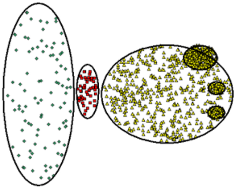
for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

- Parameter
  - $\varepsilon = 2.0$
  - $MinPts = 3$



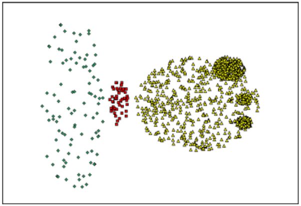
for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

When DBSCAN does not work well:

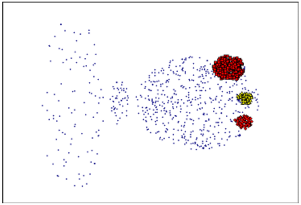


Original Points

DBScan can fail to identify clusters of varying densities



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

Further reading

- K-Means interactive playgrounds:
  https://hckr.pl/k-means-visualization
  https://www.naftaliharris.com/blog/visualizing-k-means-clustering
- Visual explanation of DBSCAN:
  https://www.youtube.com/watch?v=RDZUdRSDOok
- DBSCAN interactive playground:
  https://www.naftaliharris.com/blog/visualizing-dbscan-clustering