Lecture 13. Clustering

Information Systems
(Machine Learning)

Sergey Muravyov

23.05.2018

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

- The presentation is prepared with materials of the K.V. Vorontsov's course "Machine Leaning"

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

Problem statement

Problem: split set of objects of the same type to archive groups, such that object in these group have similar properties.

"Similarity" is formalized with an abstract measure.

X^m is training set consisting of objects from X

$\rho: X \times X \to [0; +\infty)$ is metric measure on X.

Find algorithm $a: X \to Y$, where Y is cluster set.

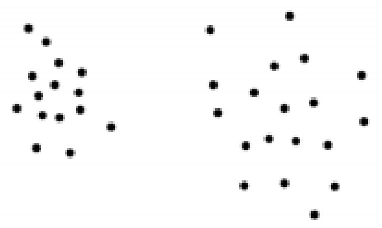
Problem formulation incorrectness

- No correct problem statement
- No universal quality criterion
- No universal metric measure (consequence of the Kleinberg theorem)
- Number of clusters is usually unknown

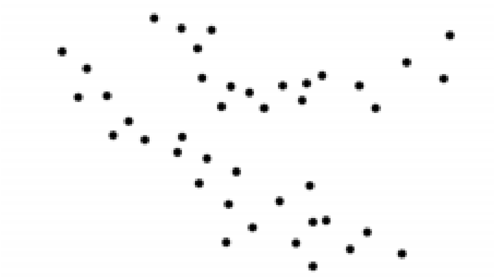
Goals of clustering

- Decrease data volume
- Find groups of similar objects
- Find unusual object
- Find hierarchy of objects (groups)

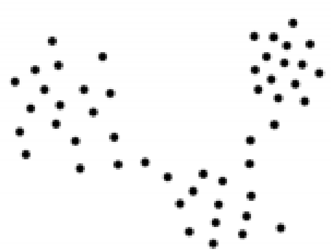
Clusters examples(1/2)



Explicitly separable

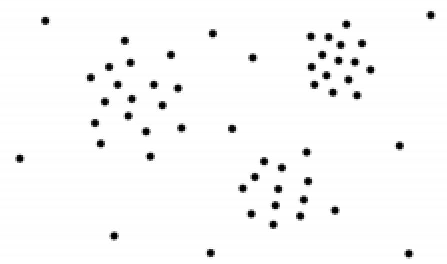


Stripes



With bridges

Clusters examples(2/2)



With regular noise

Distribution mixture

No clusters

Clustering applications

- Biology and medicine
  - Sequence analysis
  - Medical imaging (PET scans)
- Social science
  - Crime analysis
- Computer science
  - Image segmentations
- Marketing
  - Target groups
- Text analysis
- Social networks

Evaluation

- **External** – based on data that was not used for clustering, such as known class labels and external benchmarks.
- **Internal** – forbid using any external information, based on the structure of partition.

Metric space quality functional

Mean inner cluster distances:

$$F_0 = \frac{\sum_{i<j}[y_i = y_j]\rho(x_i, x_j)}{\sum_{i<j}[y_i = y_j]} \to \text{min}.$$

Mean outer cluster distance:

$$F_1 = \frac{\sum_{i<j}[y_i \neq y_j]\rho(x_i, x_j)}{\sum_{i<j}[y_i \neq y_j]} \to \text{max}.$$

Relation:

$$F_0/F_1 \to \text{min}.$$

Vector space quality functional

Mean inner cluster distances:

$$\Phi_0 = \sum_{y \in Y} \frac{1}{|K_y|} \sum_{i: y_i = y} \rho^2(x_i, c_y) \to \min.$$

Sum of outer cluster distances:

$$\Phi_1 = \sum_{y \in Y} \rho^2(c_y, c) \to \max.$$

Relation:

$$\Phi_0 / \Phi_1 \to \min.$$

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

Graph-based approach

Main idea: we will work with graphs, its vertices are objects and its edge lengths are equal to distances between the corresponding objects.

Clusters can be well-represented in graph description.

Connected component selection

Fix a radius R.

Delete edges $\{x, y\}$: $\rho(x, y) > R$.
Clusters are equal to connected components.

Fix K_1, K_2.
Change R until number of clusters is in interval $[K_1, K_2]$.

Shortest path

Fix number of clusters K.

Find minimum spanning tree (Kruskal, Boruvka, MST).

Delete $K-1$ edges with maximal lengths.

We can change for each K.

FOREL

Input: $U = X^m -$ set of unclusterized points.

1. Repeat
2. Choose a random point x from U
3. Repeat
4. $B \leftarrow$ sphere with radius R and center x
5. $c \leftarrow$ mass center of B
6. Until the sphere does not change
7. $U \leftarrow U \backslash B$
8. Until $U \neq \emptyset$

Return set of clusters

FOREL properties

Depends on R

How to choose mass center?

- Mass center in (vector space)
- Object, such that sum of distances from it to all the other objects in minimal
- Object, which in sphere of radius R contains maximum number of objects from the sample
- Object, which in sphere of radius r contains maximum number of object from sphere of radius R

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

Hierarchical approach

Main idea: build cluster hierarchy.

You can build beautiful pictures (**dendrograms**). And then you can think about number of clusters as about height of this tree.

Two approaches:

Division (split clusters)

Agglomeration (join clusters)

Lance-Williams algorithms

1. 1-element clusters:

   $t = 1, C_t = \{x_1, \ldots, x_l\};$

   $R(\{x_i\}, \{x_j\}) = \rho(x_i, x_j);$

2. For all $t = 2 \ldots l$ (t − iteration number):
3. In C_{t-1} find 2 *closest* clusters:

   $(U, V) = argmin_{U \neq V} R(U, V)$;R(U,V);

   $R_t = R(U, V);$

4. Merge them to one cluster:

   $W = U \cup V;$

   $C_t = C_{t-1} \cup \{W\} \setminus \{U, V\};$

5. For all $S \in C_t$ count $R(W, S)$.

Lance-Williams distance

Distance $R(W, S)$ between clusters

$W = U \cup V$ and S

Lance-Williams distance:

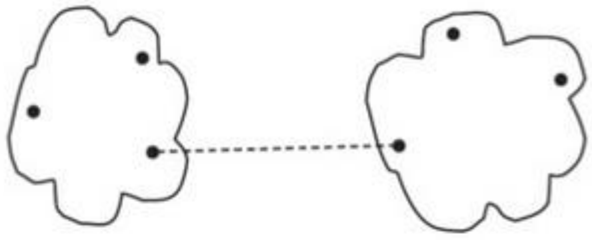
$$R(U \cup V, S) = \alpha_U R(U, S) + \\ + \alpha_V R(V, S) + \\ + \beta R(U, V) + \\ + \gamma |R(U, S) - R(V, S)|$$

Options of $R(W, S)$ (1/2)

1. Nearest neighbor distance

$$R^N(W, S) = \min_{w \in W, s \in S} \rho(w, s);$$

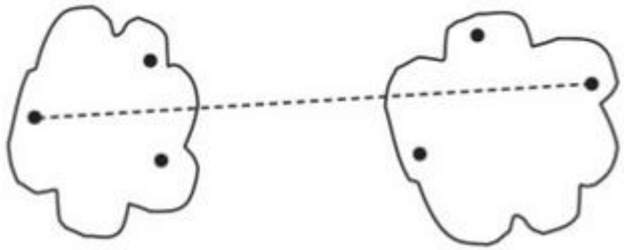
$$\alpha_U = \alpha_V = \tfrac{1}{2},\ \ \beta = 0,\ \ \gamma = -\tfrac{1}{2}.$$



2. Most distant neighbor distance

$$R^D(W, S) = \max_{w \in W, s \in S} \rho(w, s);$$

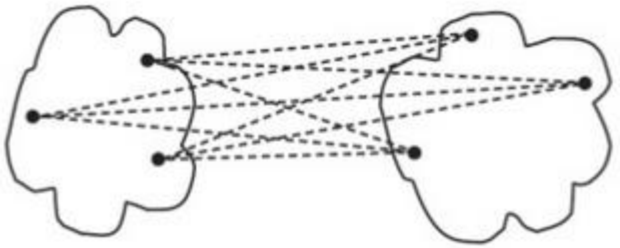
$$\alpha_U = \alpha_V = \tfrac{1}{2},\ \ \beta = 0,\ \ \gamma = \tfrac{1}{2}.$$



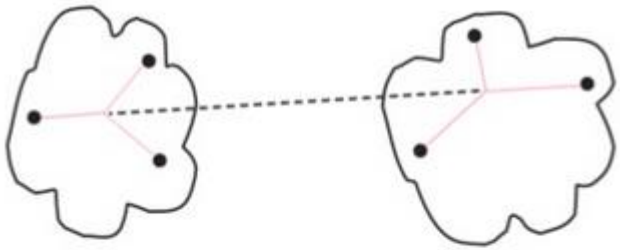
3. Mean group distance

$$R^G(W, S) = \frac{1}{|W||S|} \sum_{w \in W} \sum_{s \in S} \rho(w, s);$$

$$\alpha_U = \frac{|U|}{|W|},\ \ \alpha_V = \frac{|V|}{|W|},\ \ \beta = \gamma = 0.$$



Options of $R(W, S)$ (2/2)



4. Distance between centres

$$R^c(W,S) = \rho^2\Big(\sum_{w\in W}\frac{w}{|W|}, \sum_{s\in S}\frac{s}{|S|}\Big);$$

$$\alpha_U = \frac{|U|}{|W|},\quad \alpha_V = \frac{|V|}{|W|},$$

$$\beta = -\alpha_U\alpha_V,\quad \gamma = 0.$$

5. Ward's distance

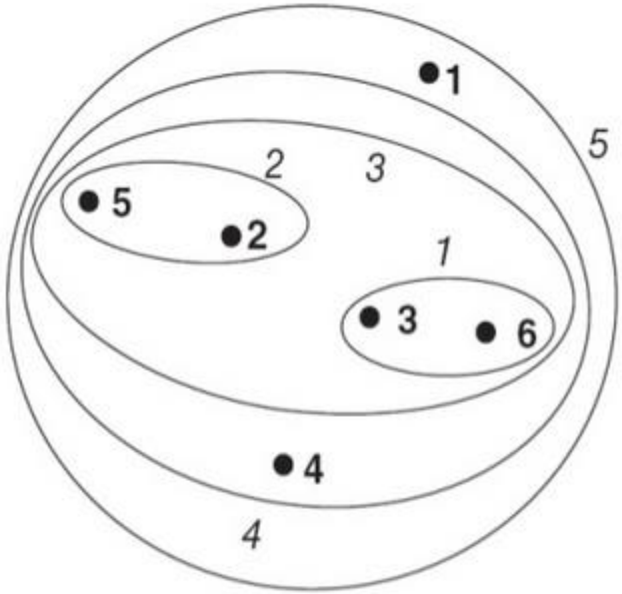
$$R^w(W,S) = \frac{|S||W|}{|S|+|W|}\,\rho^2\Big(\sum_{w\in W}\frac{w}{|W|}, \sum_{s\in S}\frac{s}{|S|}\Big);$$

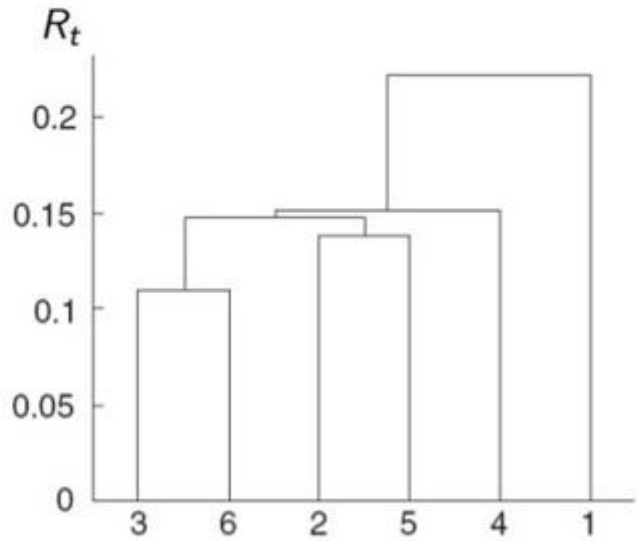
$$\alpha_U = \frac{|S|+|U|}{|S|+|W|},\quad \alpha_V = \frac{|S|+|V|}{|S|+|W|},\quad \beta = \frac{-|S|}{|S|+|W|},\quad \gamma = 0.$$

Nearest neighbor visualization

Inclusion plot

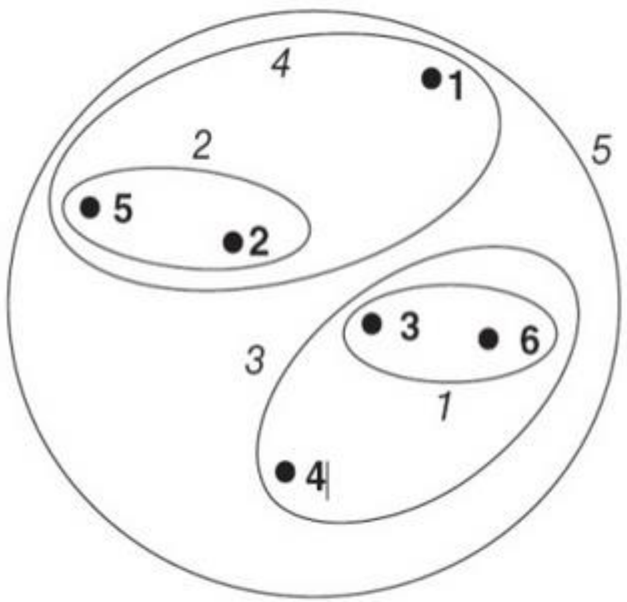
Dendrogram



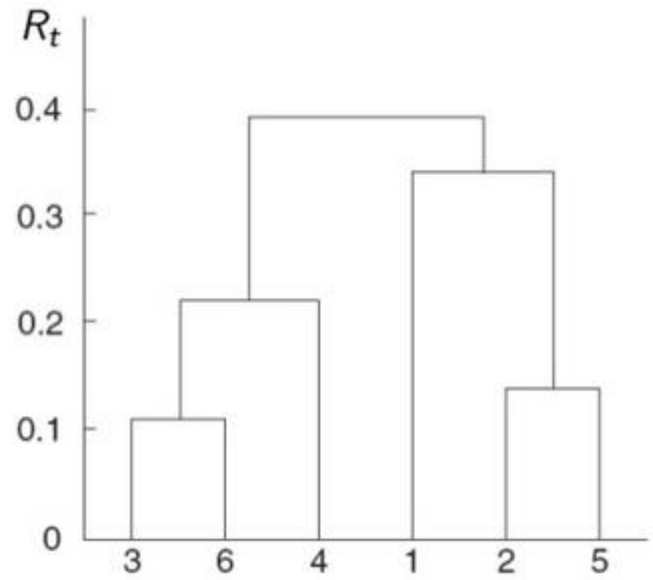


Most distant neighbor visualization

Inclusion plot

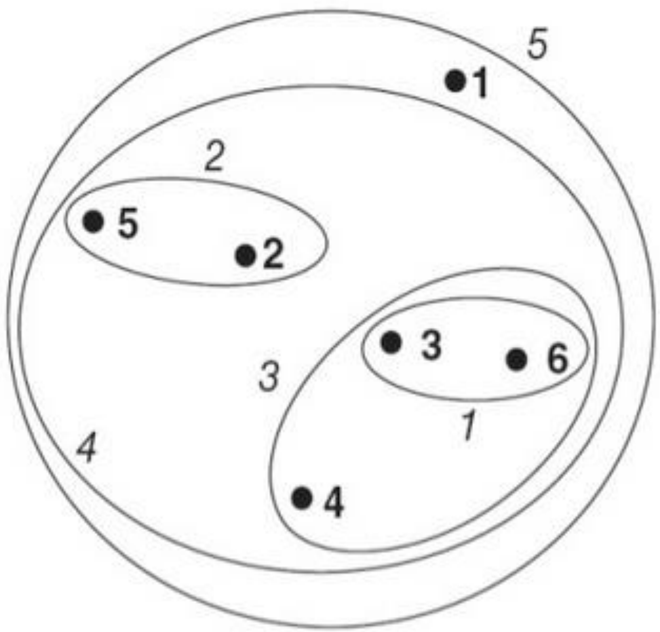


Dendrogram

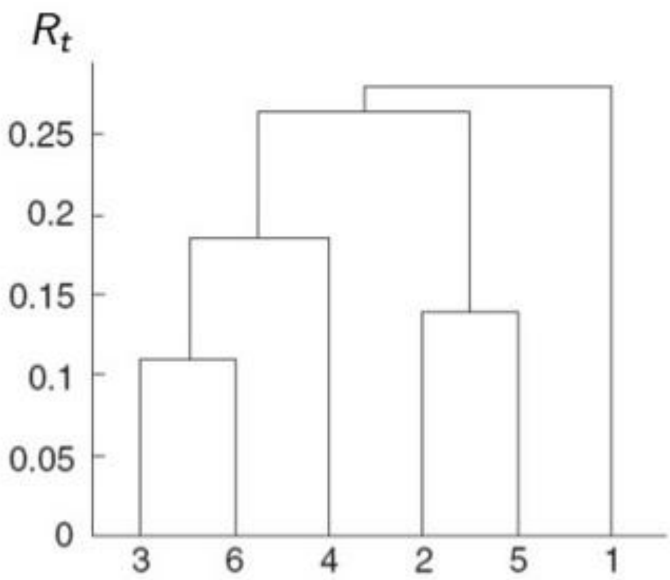


Group mean visualization

Inclusion plot

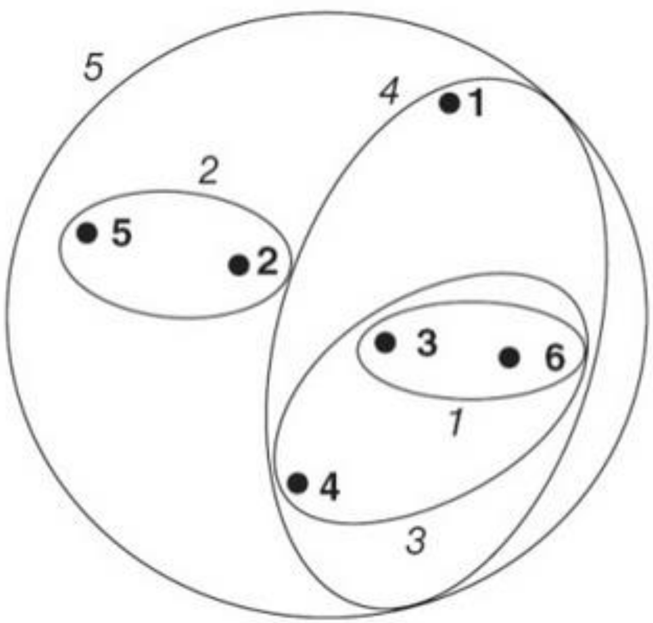


Dendrogram

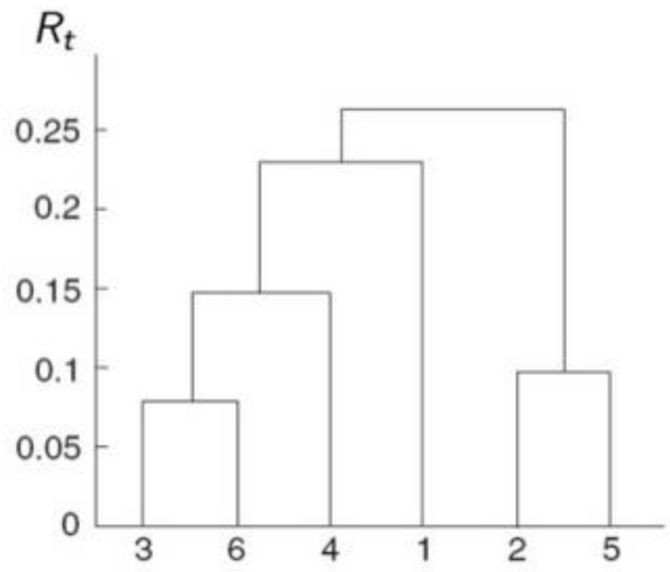


Ward's distance visualization

Inclusion plot



Dendrogram



Monotonic clustering

Clustering is **monotonic** if cluster distance do not decrease with after joining.

Theorem (Milligan, 1979)

Clustering is monotonic, if

$\alpha_U \geq 0, \alpha_V \geq 0, \alpha_U + \alpha_V + \beta \geq 1, \min\{\alpha_U, \alpha_V\} + \gamma \geq 0.$

If clustering is monotonic, dendrogram has no intersections.

Monotonic clustering

Clustering is **monotonic** if cluster distance do not decrease with after joining.

Theorem (Milligan, 1979)

Clustering is monotonic, if

$\alpha_U \geq 0, \alpha_V \geq 0, \alpha_U + \alpha_V + \beta \geq 1, \min\{\alpha_U, \alpha_V\} + \gamma \geq 0.$

If clustering is monotonic, dendrogram has no intersections.

R^C is not monotonic.

General recommendation

- Ward's distance is more preferable;
- Accelerate algorithms: join locally close clusters.
- Choose number of clusters by minimizing $|R_{t+1} - R_t|$.

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

EM

Works in the same way as original EM

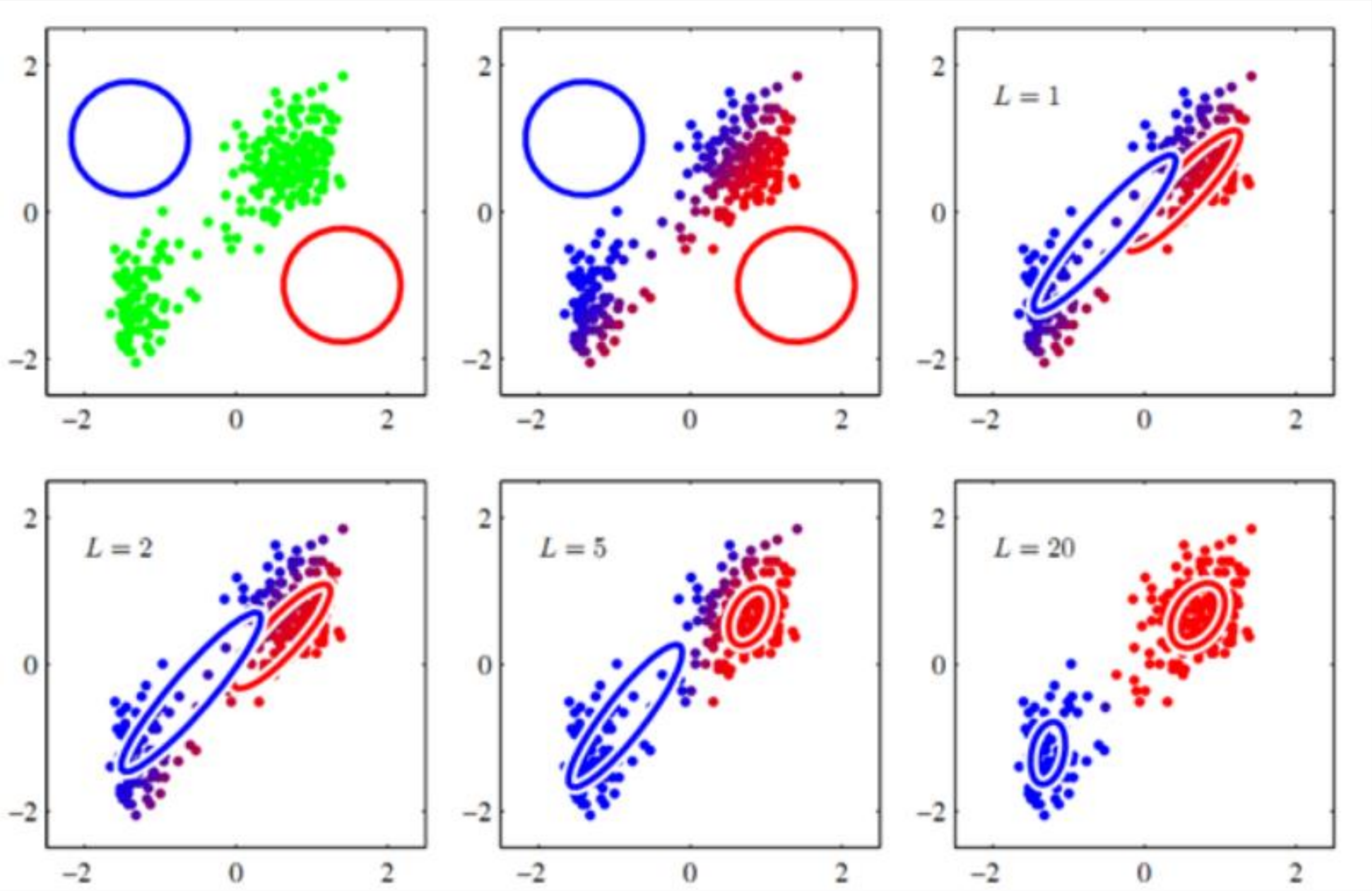
Assumption: simple sample.

w_y is prior probability of class y.

Approximate with Gaussians.

Each class is described with d-dimensional Gaussian density with diagonal covariance matrix.

EM example



***k*-means**

- ***k*-means** is an iterative algorithm that splits sets on k parts.
- Mass center of a cluster (mean intercluster distance by each feature) C_j is called *centroid*

$$c_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i \in C_j$$

k-means

It is EM-algorithm simplification with strong association with only one class.

1. Chose k points (**centroids**) $\{c_i\}_{i=1}^k$ from sample.
2. Repeat
3. For each x find nearest centroid $n(x)$.
$$C_i = \{x | n(x) = c_i\}$$
4. For each C_i find central point and claim it to be centroid.
5. Until centroid set do not change.

c-means (fuzzy clustering)

Imprecise degree of cluster belonging $u_i(x)$ of object x to cluster C_i, having $\sum_i u_i(x) = 1$.

Cluster center is chosen with

$$c_i = \frac{\sum_{x \in X^m} {u_i}^d(x) x}{\sum_{x \in X^m} {u_i}^d(x)}.$$

Re-estimate degree of belonging:

$$u_i(x) = \frac{1}{\sum_j \left(\frac{\rho(c_i, x)}{\rho\left(c_j, x\right)} \right)^{2/(d-1)}}.$$

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

Density-based approach

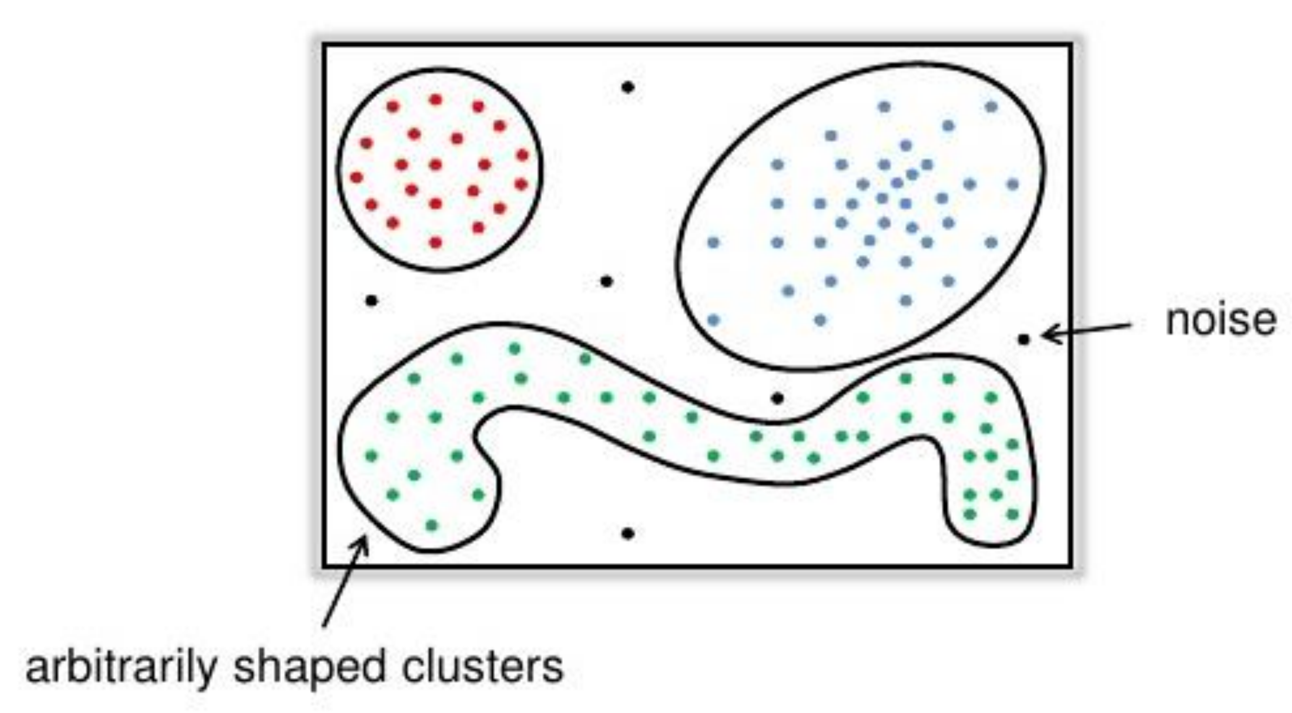
Main idea: each p point of cluster contains more than M points within eps distance:

$N_{eps}(p)$ – set of points around p within distance eps. $|N_{eps}(p)| \geq M$.

Problem with border points.

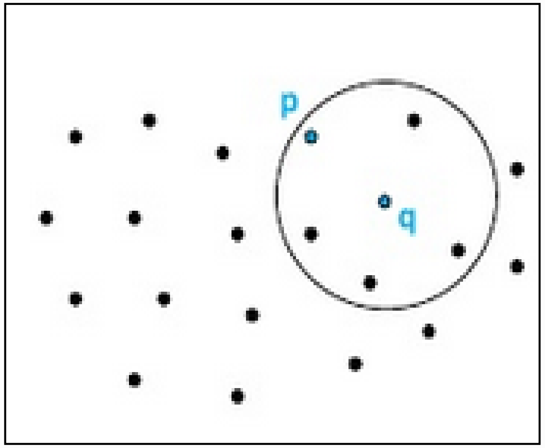
DBSCAN

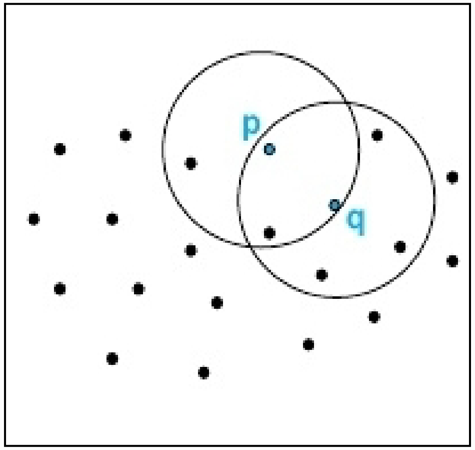
DBSCAN (Density Based Spatial Clustering of Applications with Noise)



Reachable point

p is **directly reachable** from q (given Eps and M), if $p \in N_{eps}(q)$ and $|N_{eps}(q)| \geq M$.

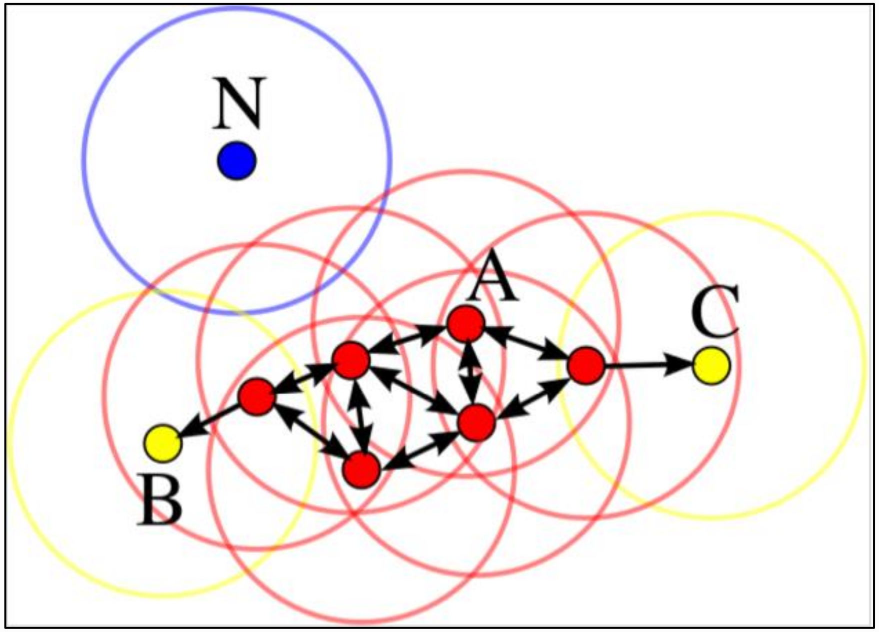




p is **reachable** from q (given Eps and M), if $\exists \{a_i\}, a_i$ is directly reachable from a_{i-1}.

Connected points

B is **connected** with C (given Eps and M), if $\exists A$, so that B and C are reachable from A (given Eps and M).



Cluster definition

Cluster C_j (given Eps and M) is non-empty set of points:

- $\forall p, q : p \in C_j, q$ is reachable $p \Rightarrow q \in C_j$
- $\forall p, q \in C_j$: p connected with q.

DBSCAN algorithm

$Input$: $D -$ data, Eps, M $-$ parameters.

foreach $d_i \in D$: $V[d_i]$ = false, $j = 0$, $Noise = \emptyset$

for all $d_i \in D$:

  if $V[d_i]$ == false **then**

    $V[d_i]$ = true, $N_i = N_{eps}(d_i)$

    if $|N_i| < M$ **then**

      $Noise = Noise + \{d_i\}$

    else

      $j = j + 1$, $\boldsymbol{Expand}(d_i, N_i, C_j, Eps, M)$

return $C = \{C_j\}$

Expand function

Input: d_i − current point, N_i − sphere, C_j − current cluster, Eps, M.

$C_j = C_j + \{d_i\}$

for all $d_k \in N_i$:

  if $V[d_k] ==$ false **then**

    $V[d_k] =$ true, $N_{ik} = N_{eps}(d_k)$

    if $|N_{ik}| \geq M$ **then**

      $N_i = N_i + N_{ik}$

  if $\nexists p : d_k \in C_p$ **then**

    $C_j = C_j + \{d_k\}$

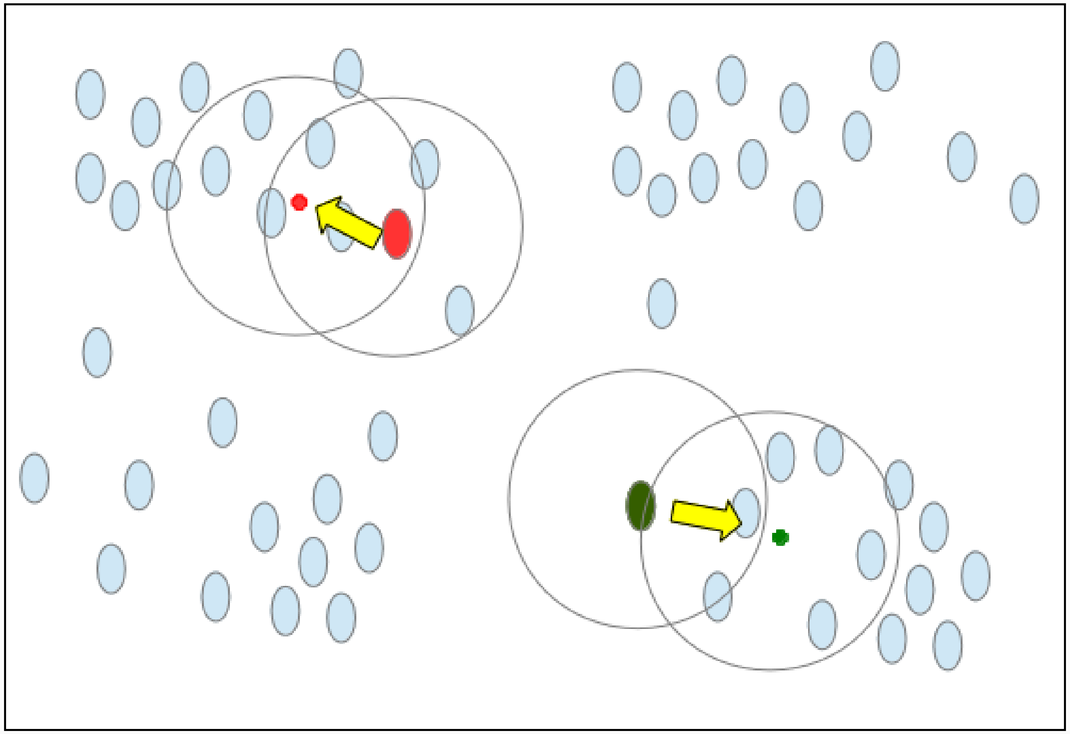
return $C = \{C_j\}$

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

Main idea

Find mass center, with maximum point density, make it a centroid

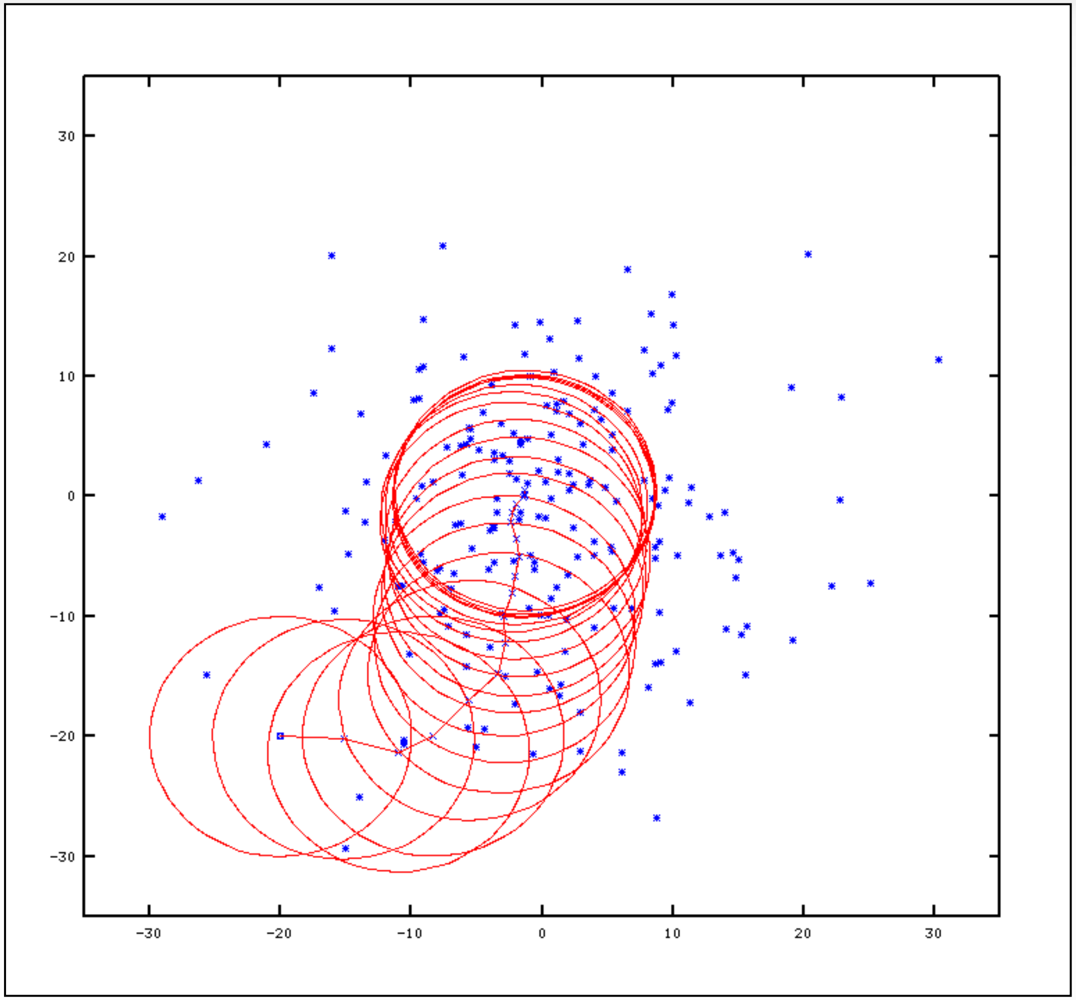


Mean-shift approach

- Set sphere around every point
- Find centroid of every sphere
- Move the center of the sphere to centroid

After each iteration centroids move to more «densed» spheres till convergence to **density modes**.

Mean-shift approach



Density modes

Mean-Shift uses gradient ascent:

$$\nabla \hat{f}(\mathbf{x}) = 1\frac{1}{nh^d}\sum_{i=1}^{n}\frac{\partial}{\partial \mathbf{x}}K(\frac{\mathbf{x} - \mathbf{x_i}}{h})$$

$$\nabla \hat{f}(\mathbf{x}) = 0$$

Gaussian kernel

$$\frac{\partial}{\partial \mathbf{x}} K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h}\right) = K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h}\right) \frac{\mathbf{x} - \mathbf{x_i}}{h} \frac{1}{h}$$

$$\Rightarrow \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h}\right)\mathbf{x} = \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h}\right)\mathbf{x_i}$$

«Ascending» direction

Vector of ascending kernel function

$$\mathrm{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} K(\frac{\mathbf{x} - \mathbf{x_i}}{h})\mathbf{x_i}}{\sum_{i=1}^{n} K(\frac{\mathbf{x} - \mathbf{x_i}}{h})}$$

Mean shift

$$\mathrm{m}(\mathbf{x}) - \mathbf{x} = \frac{\sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h}\right)\mathbf{x_i}}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h}\right)} - \mathbf{x}$$

Mean-shift algorithm

Input: $D -$ data.

do

    foreach $\mathbf{x}_i \in D$: count $\mathbf{m}(\mathbf{x_i})$

    $\nabla \hat{f}(\mathbf{x}) \to \nabla \hat{f}(\mathbf{m}(\mathbf{x}) - \mathbf{x})$

while $\nabla \hat{f}(\mathbf{x}) \neq 0$

return $C = \{C_j\}$

Lecture plan

- Clustering Problem
- Graph-based clustering
- Hierarchical clustering
- EM clustering
- Density-based clustering
- Non-parametric clustering
- Semi-supervised learning

Problem formulation

A training sample is given, which is

$$\{(x_1, y_1), \dots, (x_\ell, y_\ell), x_{\ell+1}, \dots, x_{\ell+m}\} = T^\ell \cup X^m,$$

where $\ell \ll m$.

Solve as supervised problem (on T^ℓ, "forgetting" about X^m)

Solve as unsupervised problem (on $X^\ell \cup X^m$, "forgetting" about Y^m).

Semi-supervised learning

Three approaches:

- Solve with native methods
- Solve with supervised algorithms without estimating error on unlabeled objects
- Solve with unsupervised algorithm achieving clusters which contains at least one object and all objects belonging to a cluster have the same label

Why is it important to solve this problem?

Usually it's chip to get objects and it is expensive to label objects.

Object mining is automated and object-labeling is expert-based.

Typical example: data from Internet (posts, pictures, articles) or generic data.

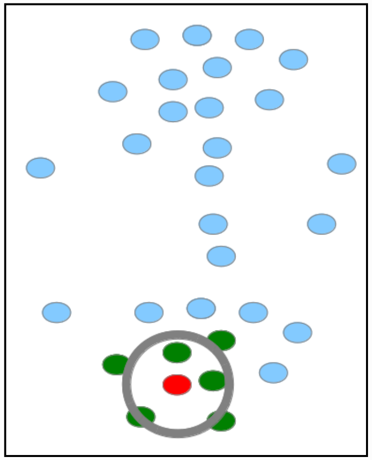
Method adaptation

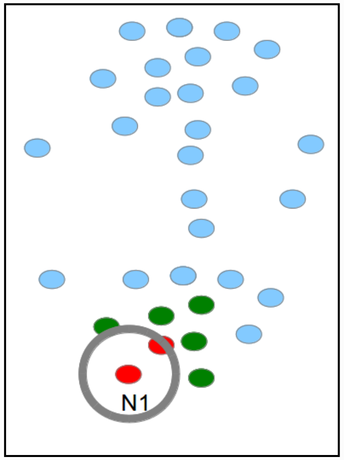
It’s much simpler to adopt unsupervised methods.

Each method can be modified just by including a certain constrain, which should not allow to get clusters with differently labeled object.

DBSCAN. Example

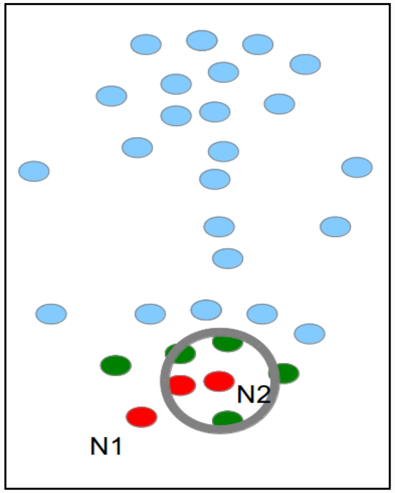
Initial parameters: $M = 4$, $Eps > 0$. We take the first point at random. It has 6 neighbors from N_0 (left figure) $\Rightarrow$ create the first cluster (red) and start the extension. The first of the neighbors N_1 turned out to be a boundary one - we add it to the cluster (right figure).

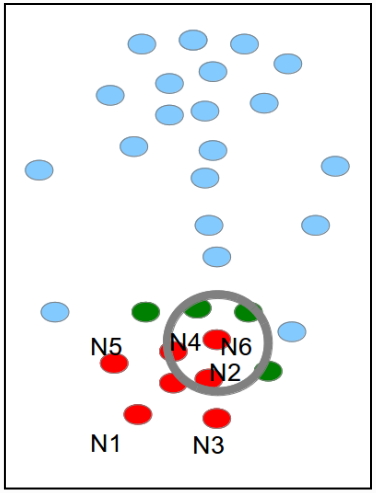




DBSCAN. Example

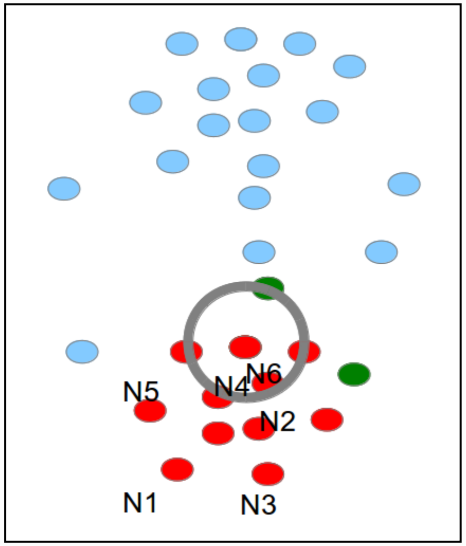
The next neighbor is N_2. He has 5 of his neighbors from N_i. (left figure) ⇒ Adding new neighbors to the old ones (another green neighbor appeared). And so on. When we went around all the original neighbors N_1 - N_6 (right figure), we continue with the new, "green" ones.

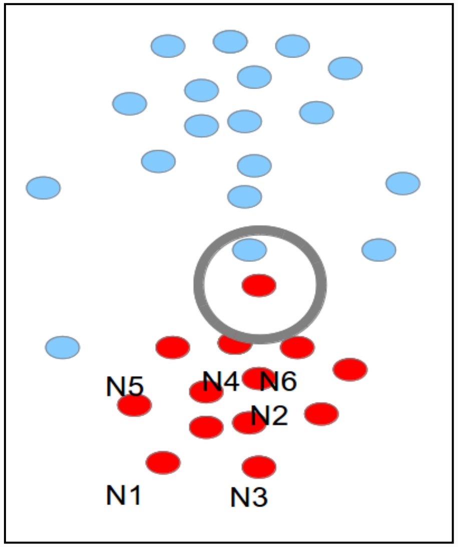




DBSCAN. Example

After scanning the neighbors of points N_1 - N_6, there are only two "green" points (left figure), after processing of which the first cluster is formed (right) and then a point from the original array is again taken at random.





DBSCAN. Example

When the choice was made on a "single point", where the number of neighbors is less than M = 4 (left figure), it is added to the noise array $Noise$, and then the next unvisited point is again randomly selected. As a result, 2 clusters are formed in this example, and 6 points are classified as noises (right figure). We note that two points between the clusters ("isthmus") also fell into the number of noise.