Data Mining
Lecture 3: Discovering Groups

Jo Grundy

ECS Southampton

28^{th} February 2022

Discovering Groups - Introduction

Understanding large datasets is hard, especially if it has high dimensional features

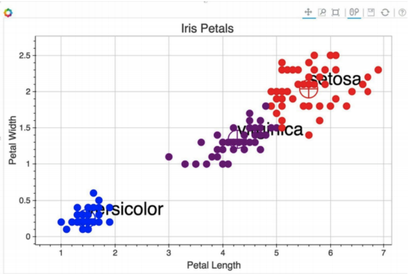
To help understand a dataset:

- Find similar data items
- Find similar features

Discovering Groups - Clustering

Grouping data, just using the feature vectors

- Unsupervised
- Similar feature vectors grouped together
- Can be
  - Soft (allow overlapping groups)
  - Hard (each item assigned to one group)



Discovering Groups - Clustering

We will cover:

- KMeans
- DBSCAN
- Hierarchical Clustering
- Mean Shift

Discovering Groups - K Means

K Means needs a fixed number of clusters **K**

It first initialises K centroids

Calculates which points are closest to each, this is the cluster.

The mean for each cluster is calculated using alll the points in the cluster.

This process is then repeated until there is no more change

Discovering Groups - K Means

Algorithm 1: K Means clustering

```
Data: X, K
initialise K centroids;
while positions of centroids change do
    for each data point do
        assign to nearest centroid
    end
    for each centroid do
        move to average of assigned data points
    end
end
return centroids, assignments;
```

A special case of Expectation Maximisation - why?

Discovering Groups - K Means

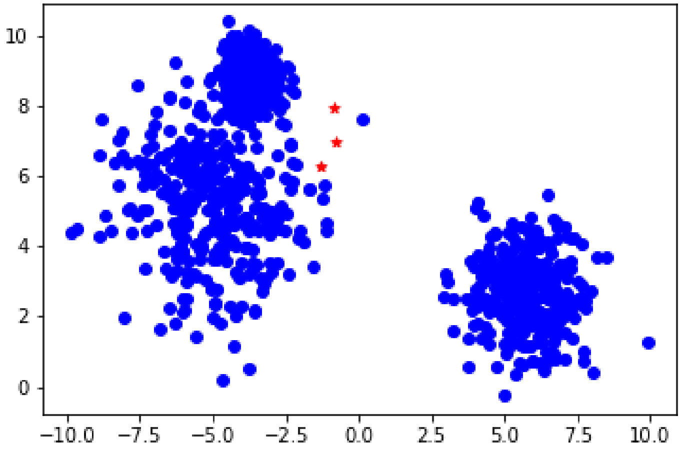
Algorithm 2: K Means clustering

```
Data: X, K
initialise K centroids;
while positions of centroids change do
    for each data point do
        assign to nearest centroid ;        // Expectation of
         associations
    end
    for each centroid do
        move to average of assigned data points ;
         // Maximisation of likelihood
    end
end
return centroids, assignments;
```

Assumes spherical clusters

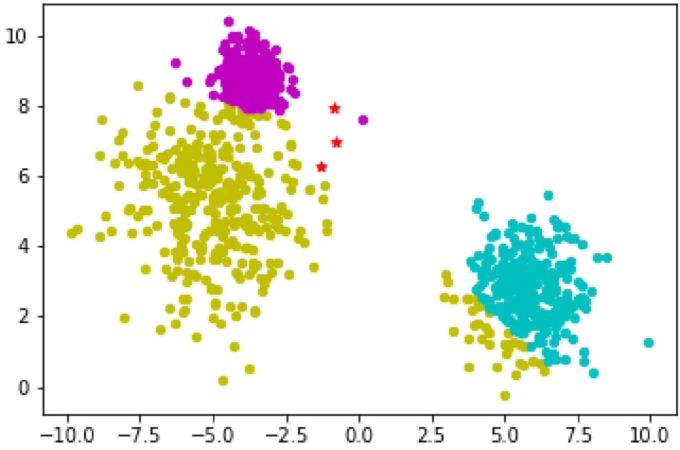
Discovering Groups - K Means

Step by step: Initialise with some random means:



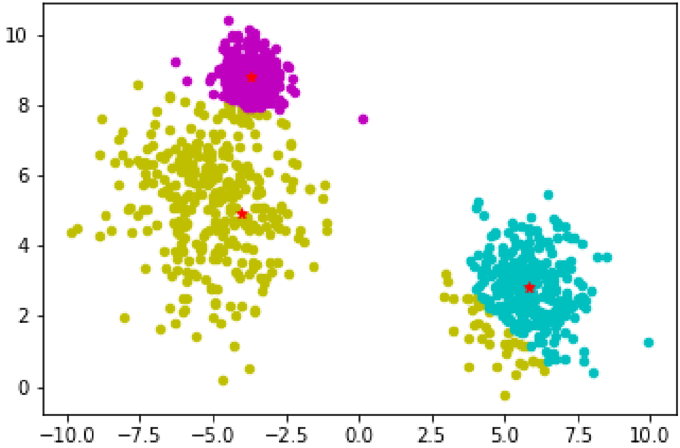
Discovering Groups - K Means

Step by step: Calculate which are closest



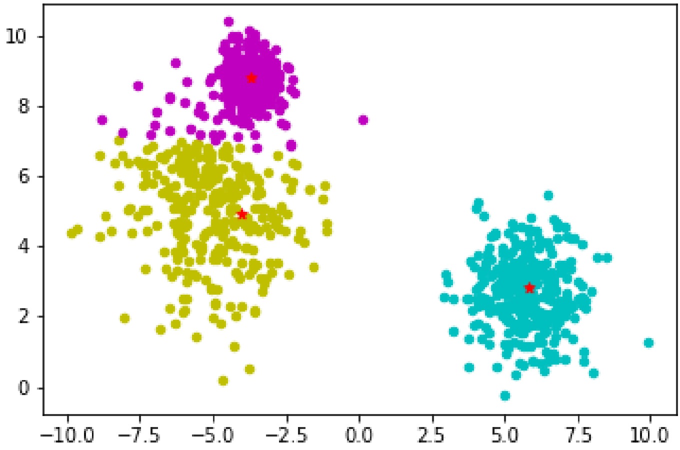
Discovering Groups - K Means

Step by step: Then calculate the new mean, that is the new centroid



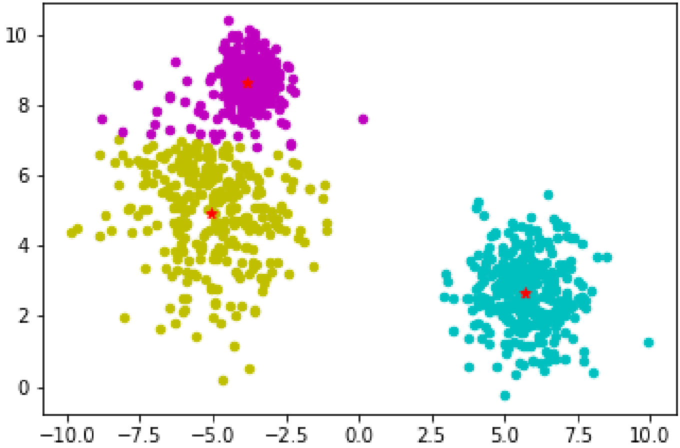
Discovering Groups - K Means

Step by step: Calculate which are closest



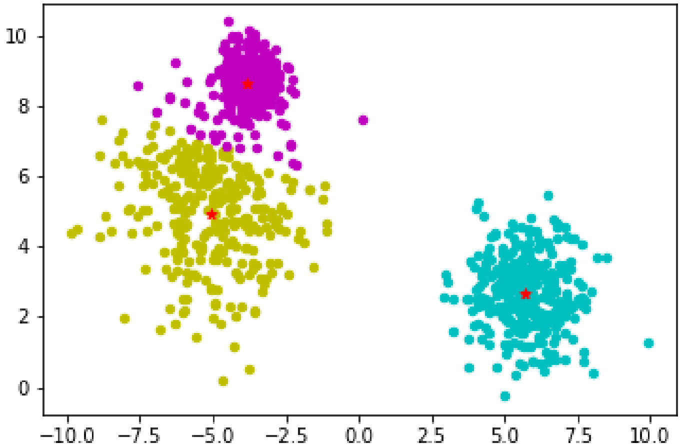
Discovering Groups - K Means

Step by step: Then calculate the new mean, that is the new centroid



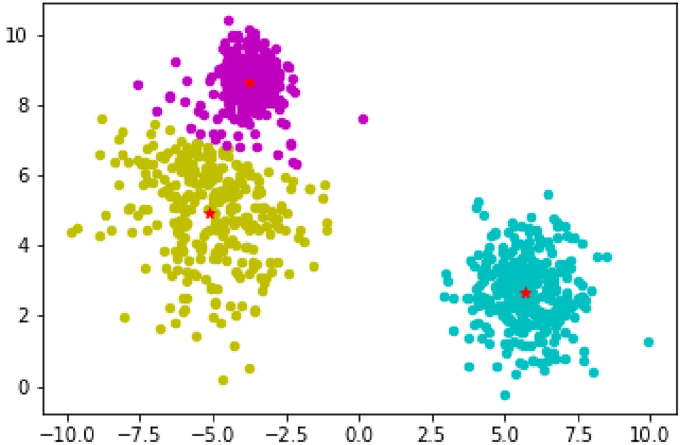
Discovering Groups - K Means

Step by step: Calculate which are closest



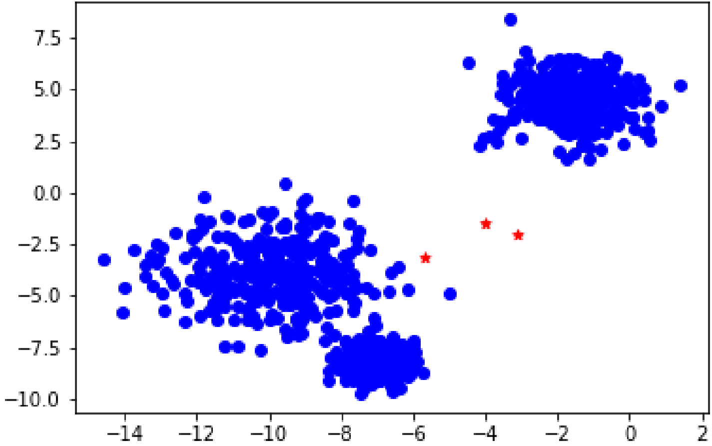
Discovering Groups - K Means

Step by step: Then calculate the new mean, that is the new centroid



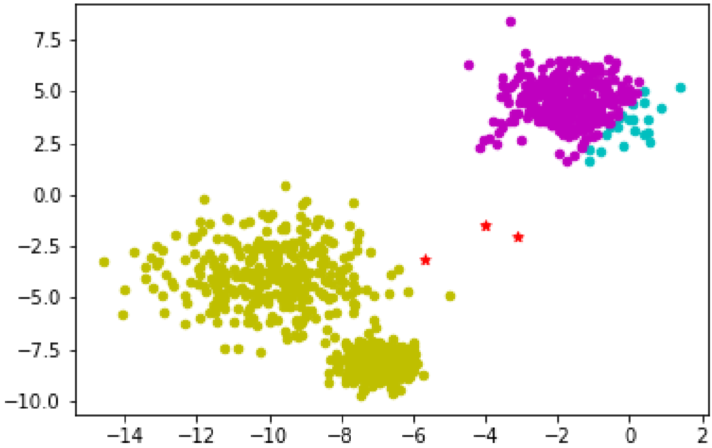
Discovering Groups - K Means

Sometimes it gets it wrong..



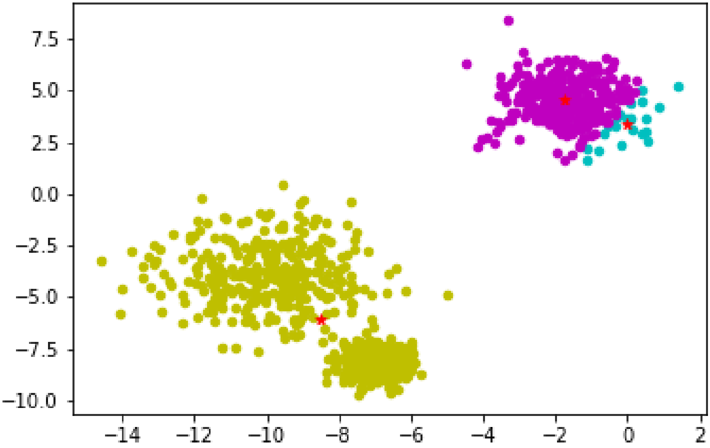
Discovering Groups - K Means

Step by step: Calculate which are closest



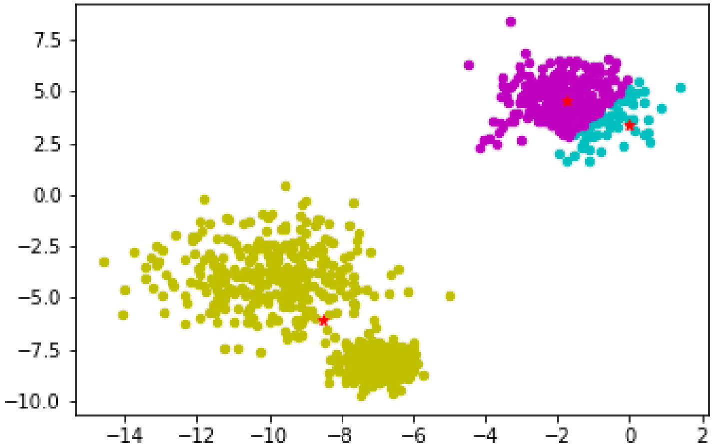
Discovering Groups - K Means

Step by step: Then calculate the new mean, that is the new centroid



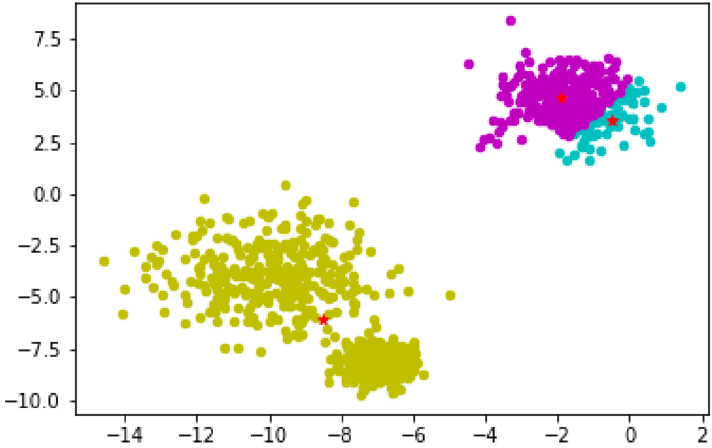
Discovering Groups - K Means

Step by step: Calculate which are closest



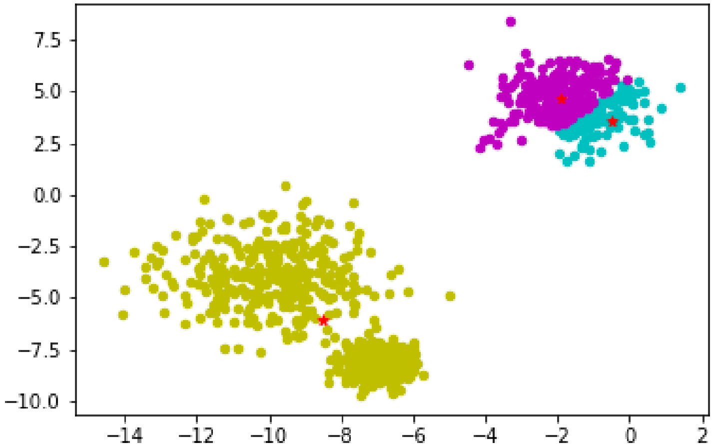
Discovering Groups - K Means

Step by step: Then calculate the new mean, that is the new centroid



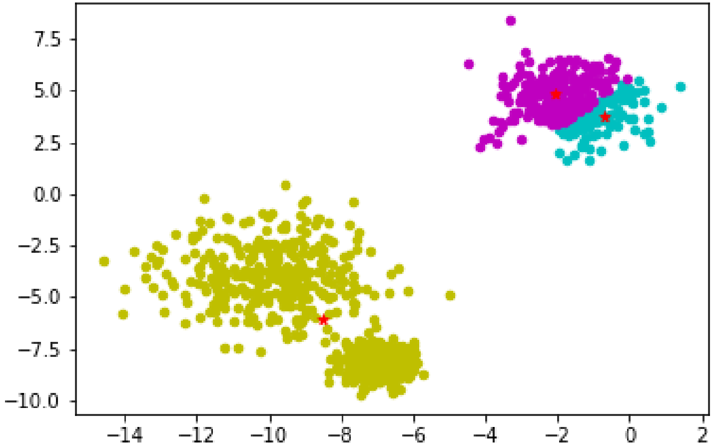
Discovering Groups - K Means

Step by step: Calculate which are closest



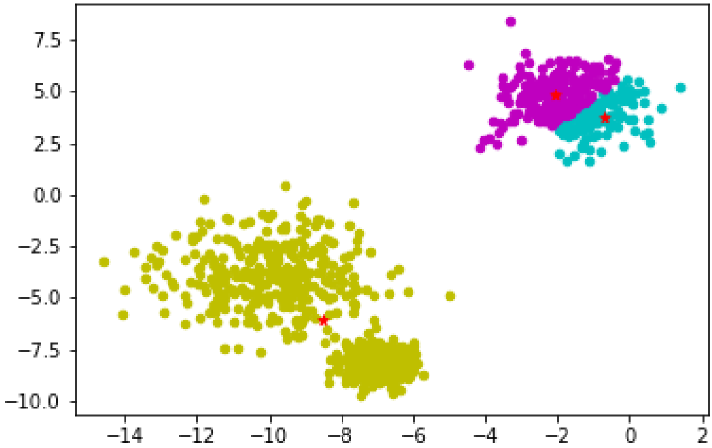
Discovering Groups - K Means

Step by step: Then calculate the new mean, that is the new centroid



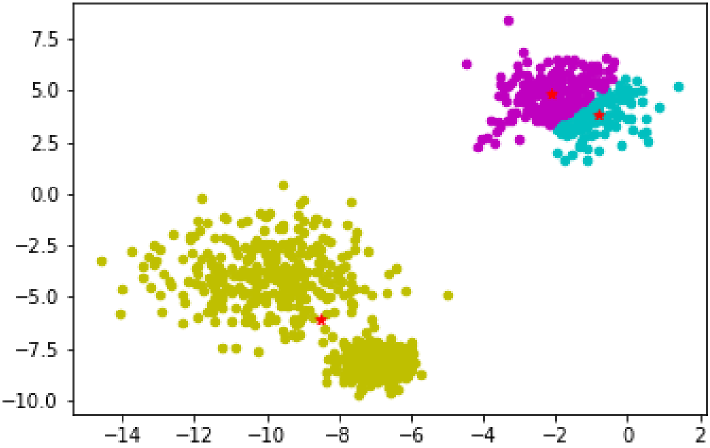
Discovering Groups - K Means

Step by step: Calculate which are closest



Discovering Groups - K Means

Step by step: Then calculate the new mean, that is the new centroid



Discovering Groups - K Means

K Means can quickly and cheaply cluster data.
Problems?

- need to specify k
- assumes spherical data
- depends on good initial centroid guesses
- may converge on local minimum

Gaussian Mixture models can work better, using a generalisation of KMeans, not discussed here.

Discovering Groups - DBSCAN

Density Based Spatial Clustering and Noise
Tries to find areas of density and follow them to generate the clusters. The number of clusters doesn't need to be specified.
But..
Does need:

- maximum radius
- minimum number

Max radius is the limit on which to look for neighbours
Min number is the lower limit on what can be in a cluster

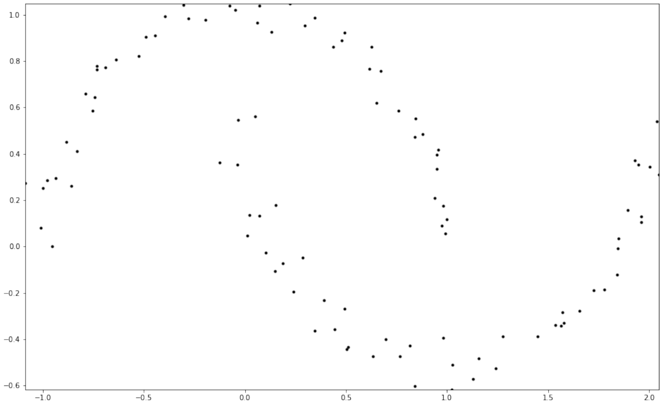
Discovering Groups - DBSCAN

Algorithm 3: DBSCAN

Data: X, *eps*, *min_pts*
initialse *labels* list as zeros, *count* list, *core* list;
Find neighbours for each point, Find core points;
class = 1;
for *each core point p* **do**
    add neighbours(p) to queue;
    while *queue not empty* **do**
        neighbours = next(queue);
        for *q in neighbours* **do**
            set label(q = *class*;
            if *label(q) is 'core'* **then**
                add neighbours(q) to queue
            end
        end
    end
    class = *class* + 1
end
return labels;

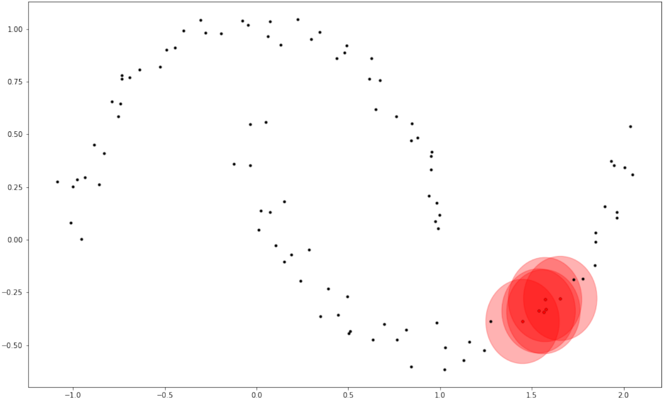
Discovering Groups - DBSCAN

Step by step:



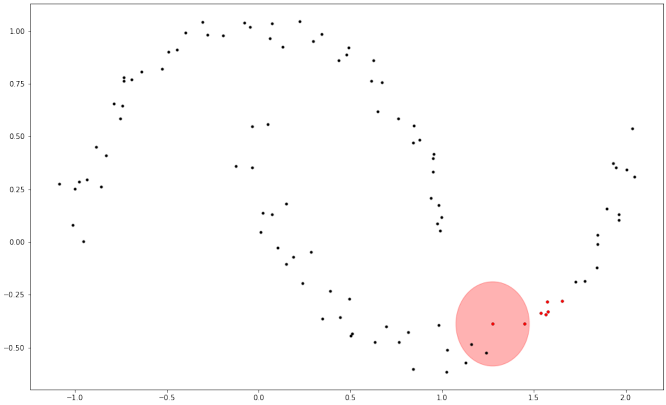
Discovering Groups - DBSCAN

Step by step:



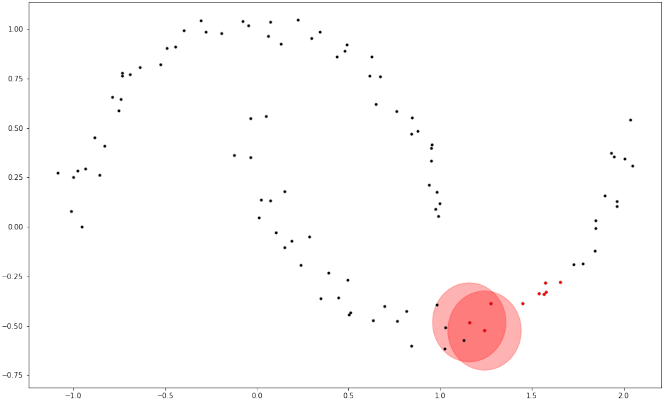
Discovering Groups - DBSCAN

Step by step:



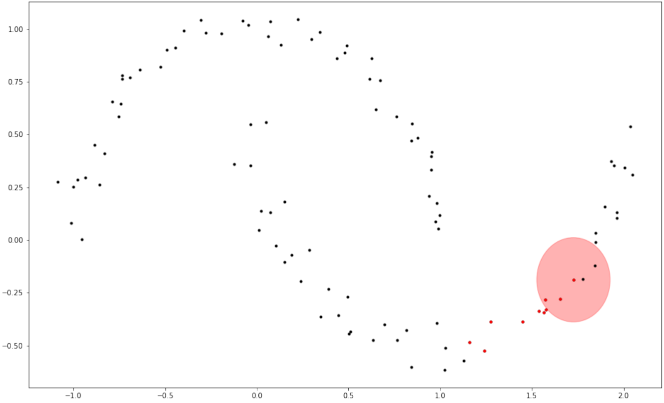
Discovering Groups - DBSCAN

Step by step:



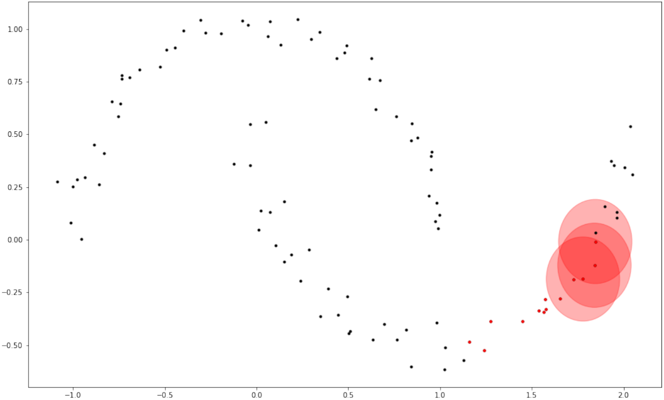
Discovering Groups - DBSCAN

Step by step:



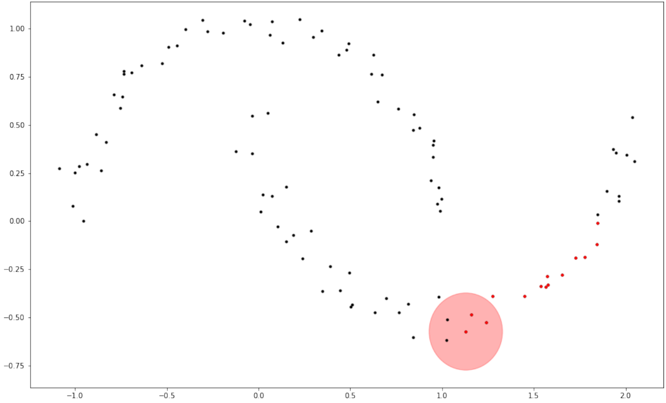
Discovering Groups - DBSCAN

Step by step:



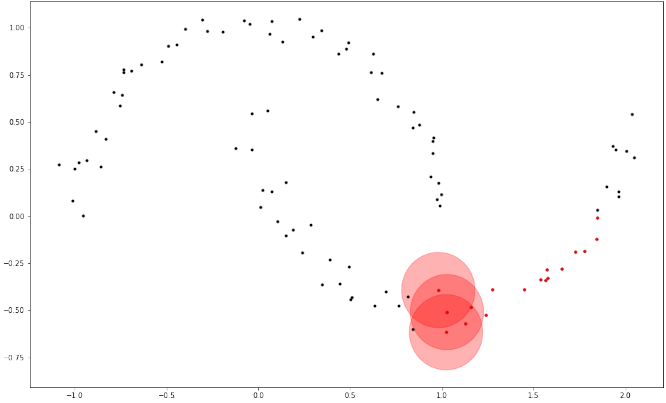
Discovering Groups - DBSCAN

Step by step:



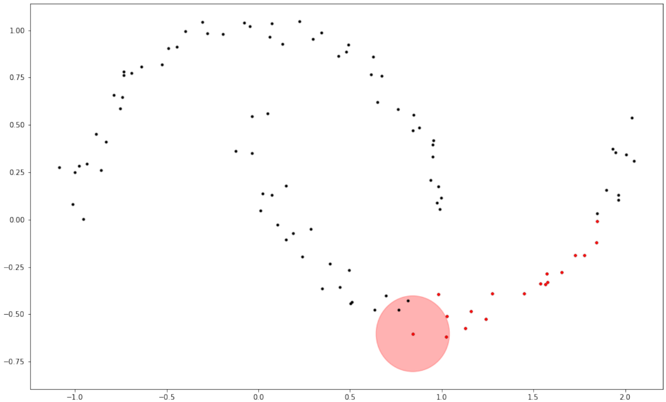
Discovering Groups - DBSCAN

Step by step:



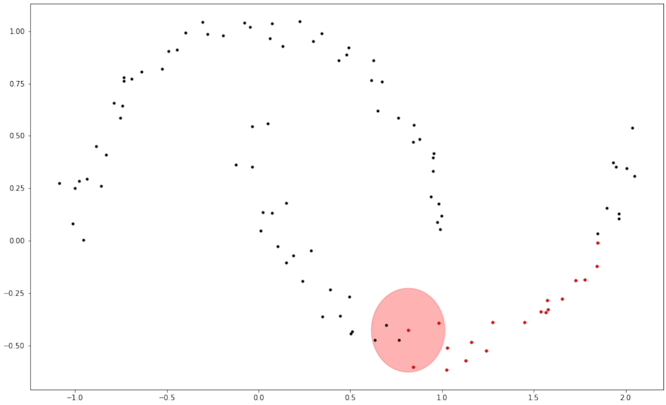
Discovering Groups - DBSCAN

Step by step:



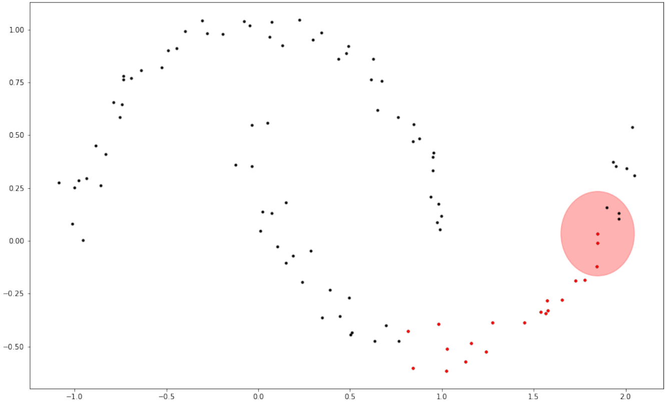
Discovering Groups - DBSCAN

Step by step:



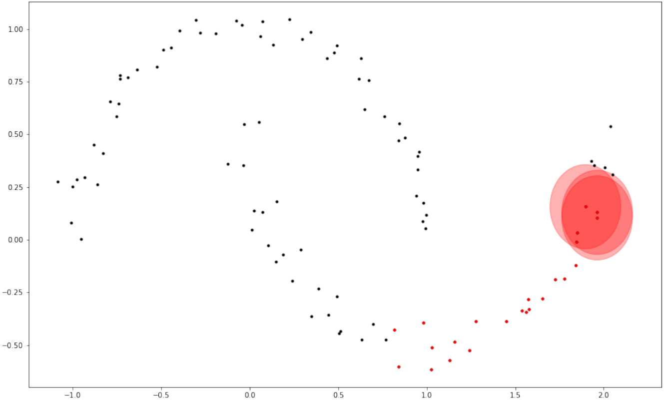
Discovering Groups - DBSCAN

Step by step:



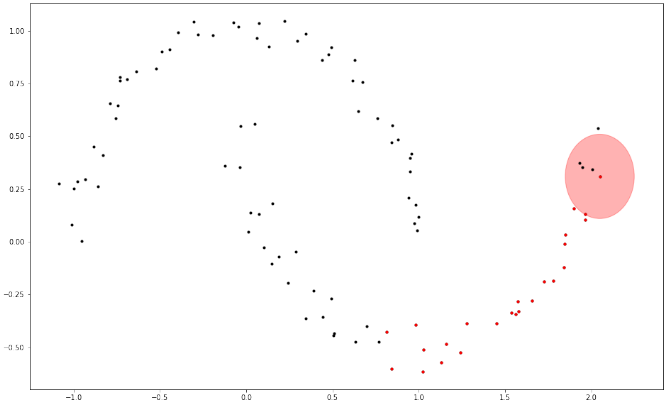
Discovering Groups - DBSCAN

Step by step:



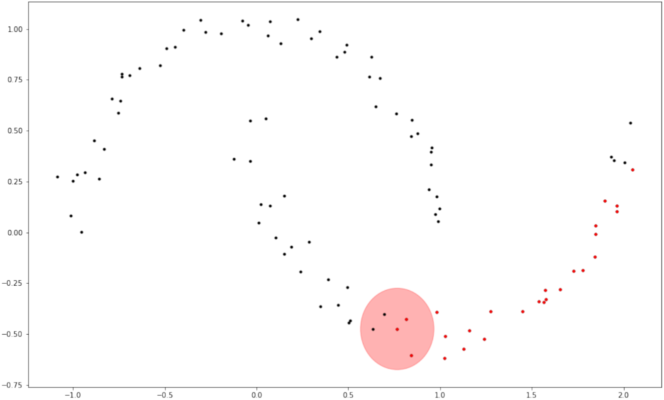
Discovering Groups - DBSCAN

Step by step:



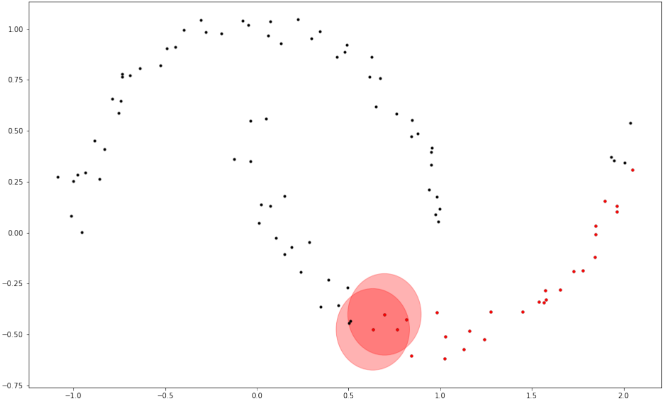
Discovering Groups - DBSCAN

Step by step:



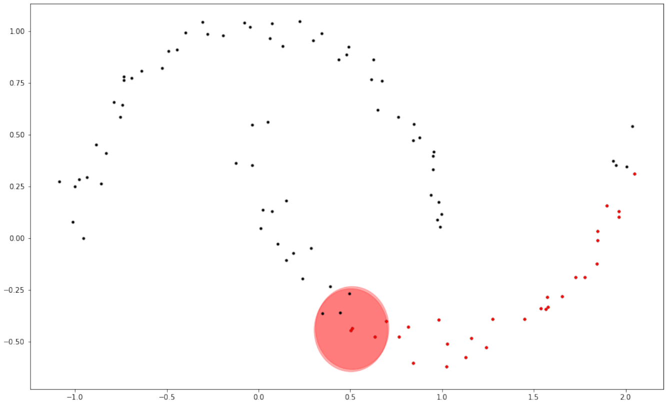
Discovering Groups - DBSCAN

Step by step:



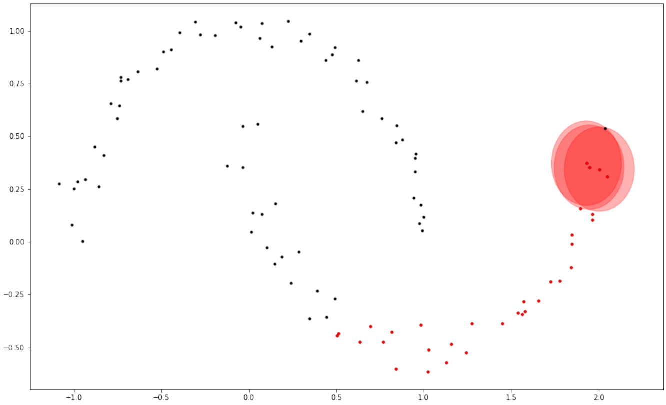
Discovering Groups - DBSCAN

Step by step:



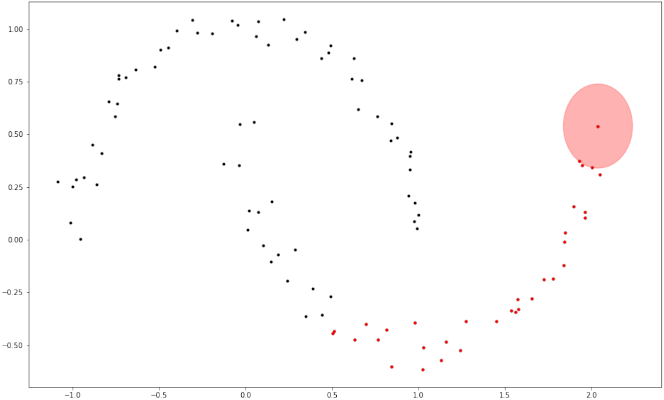
Discovering Groups - DBSCAN

Step by step:



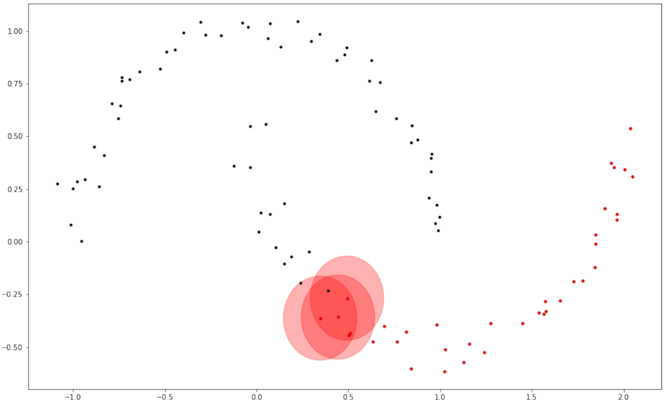
Discovering Groups - DBSCAN

Step by step:



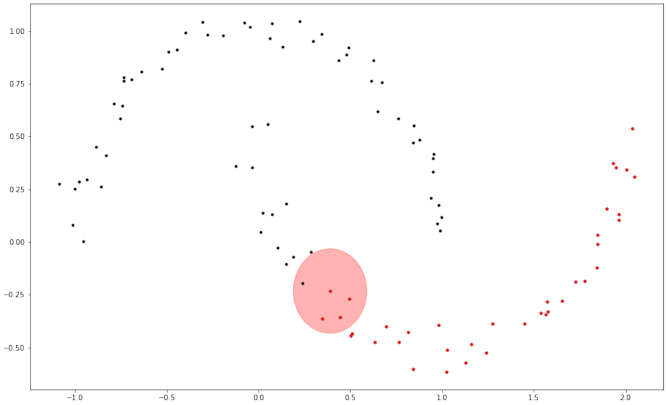
Discovering Groups - DBSCAN

Step by step:



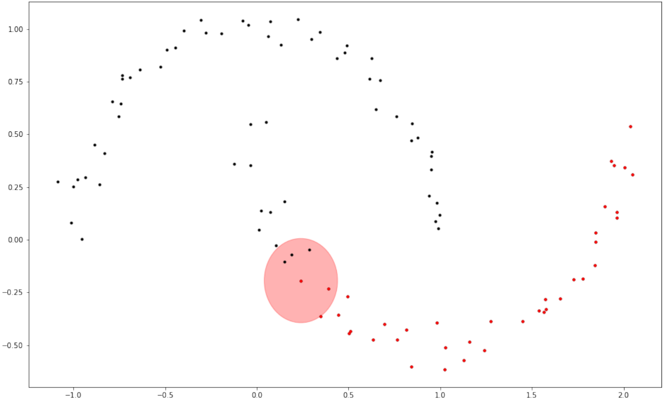
Discovering Groups - DBSCAN

Step by step:



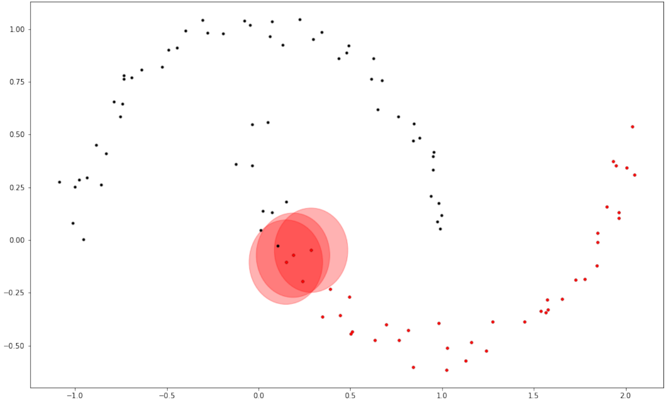
Discovering Groups - DBSCAN

Step by step:



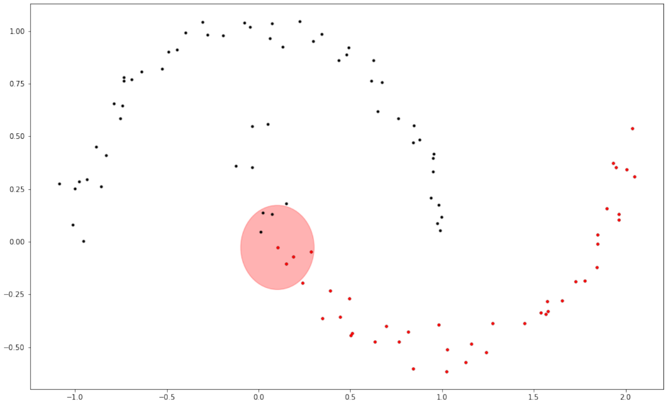
Discovering Groups - DBSCAN

Step by step:



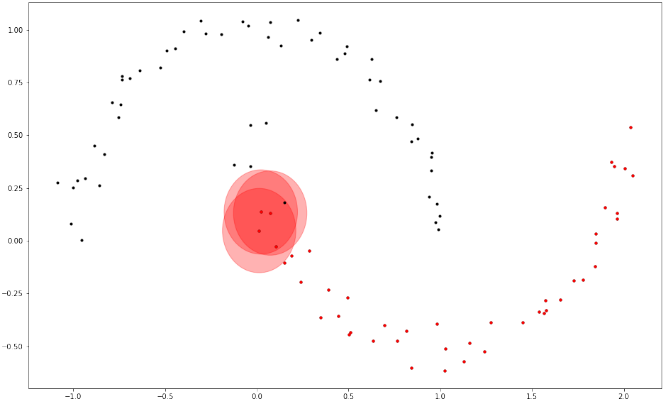
Discovering Groups - DBSCAN

Step by step:



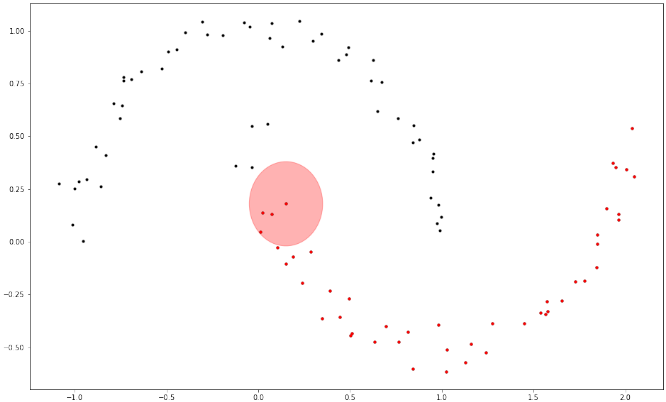
Discovering Groups - DBSCAN

Step by step:



Discovering Groups - DBSCAN

Step by step:



Discovering Groups - DBSCAN

DBSCAN works well on any shape of data, and is robust to outliers.
problems?

- can struggle in high dimensions
- needs a distance parameter
- same parameter may not work for different cluster density
- need also minimum number specified

Discovering Groups - Hierarchical Clustering

Hierarchical Clustering:

Creates a binary tree that recursively groups pairs of similar items or clusters

Can be:

- Agglomerative (bottom up)
- Divisive (top down)

We will look at Agglomerative clustering. Needs a distance measure.

Discovering Groups - Hierarchical Clustering

Algorithm 4: Hierarchical Agglomerative Clustering

Data: N data points with feature vectors X_i $i = 1 \ldots N$

$numClusters = N$;

while $numClusters > 1$ **do**

    cluster1, cluster2 = FindClosestClusters();

    merge(cluster1, cluster2);

end

The distance between the clusters is evaluated using a linkage criterion.

If each merge is recorded, a binary tree structure linking the clusters can be formed.

This gives a **dendrogram**

Discovering Groups - Hierarchical Clustering

Linkage criterion: A measure of dissimilarity between clusters

Centroid Based:

- Dissimilarity is equal to distance between centroids
- Needs numeric feature vectors

Distance-Based:

- Dissimilarity is a function of distance between items in clusters
- Only needs precomputed measure of similarity between items

We could compute a distance matrix between points

Discovering Groups - Hierarchical Clustering

Centroid based linkage:

- WPGMC: Weighted Pair Group Method with Centroids
  When two clusters are combined into a new cluster, the average of the two centroids is the new centroid
- UPGMC: Unweighted Pair Group Method with Centroids
  When two clusters are combined into a new cluster, the new centroid is recalculated based on the positions of the items

Discovering Groups - Hierarchical Clustering

Distance based linkage:

- **Minimum**, or **single-linkage clustering** Distance between two closest members

  $$\min d(a, b) : a \in A, b \in B$$

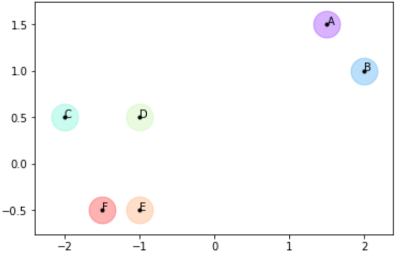
  Produces long, thin clusters
- **Maximum**, or **complete-linkage clustering** Distance between two most distant members

  $$\max d(a, b) : a \in A, b \in B$$

  Finds compact clusters, approximately equal diameter
- **Mean** or **Average Linkage Clustering** (**UPGMA**: Unweighted Pairwise Group Method with Arithmetic Mean):

  $$\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$$

Discovering Groups - Hierarchical Clustering



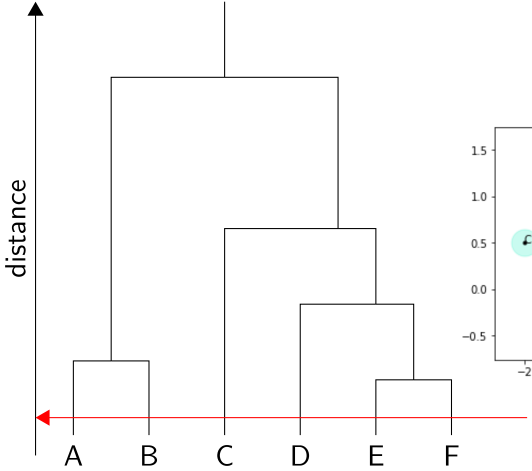
With sample data:

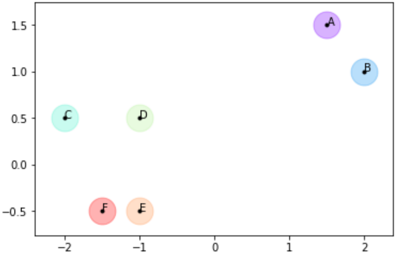
$$X = \begin{bmatrix} 1.5 & 1.5 \\ 2.0 & 1.0 \\ 2.0 & 0.5 \\ -1.0 & 0.5 \\ -1.5 & -0.5 \\ -1 & 0.5 \end{bmatrix}$$

Distance matrix:

	A	B	C	D	E	F
A	0	0.7	2.7	1.8	...	
B	0.7	0	...			
C	2.7		0	...		
D	1.8			0	...	
E	⋮				0	

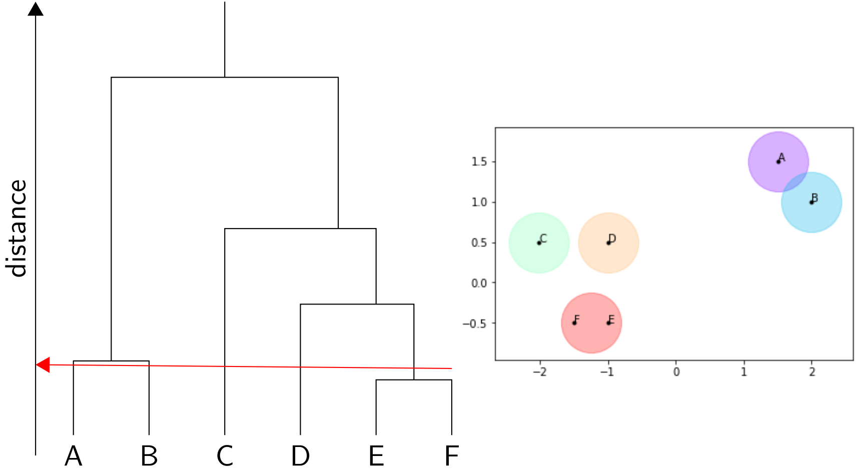
Discovering Groups - Centroid Clustering





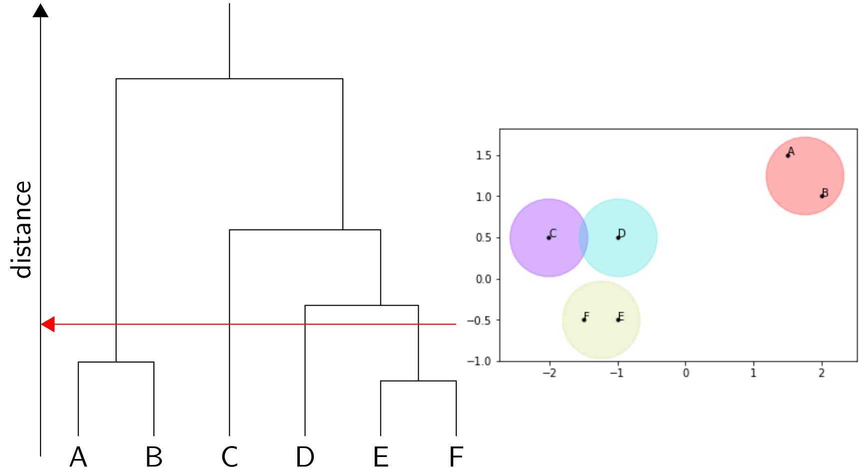
Using minimum or single-linkage clustering

Discovering Groups - Centroid Clustering



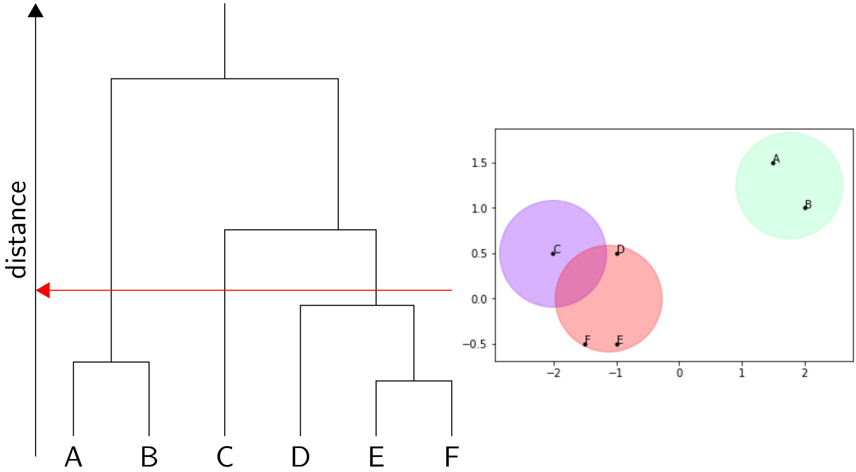
Using minimum or single-linkage clustering

Discovering Groups - Centroid Clustering



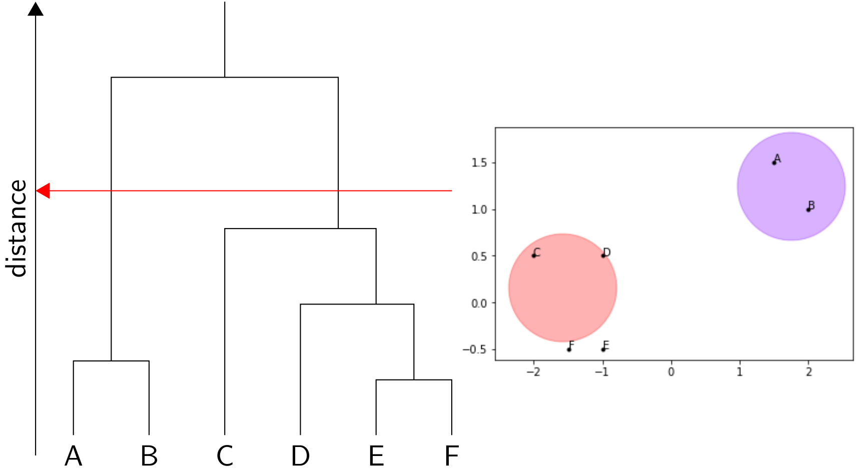
Using minimum or single-linkage clustering

Discovering Groups - Centroid Clustering



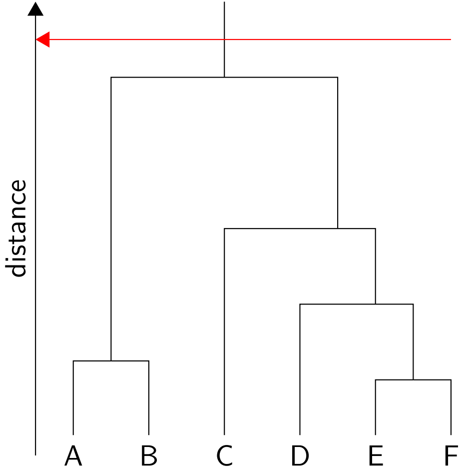
Using minimum or single-linkage clustering

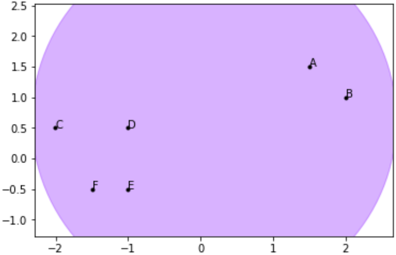
Discovering Groups - Centroid Clustering



Using minimum or single-linkage clustering

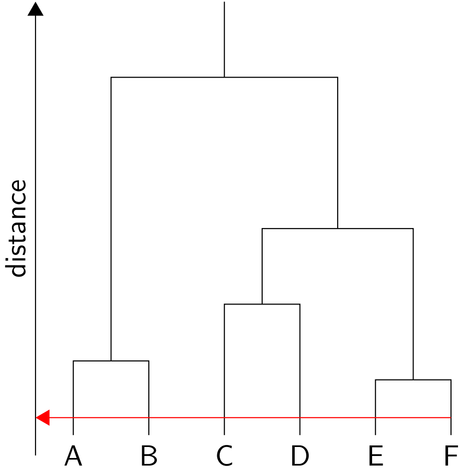
Discovering Groups - Centroid Clustering

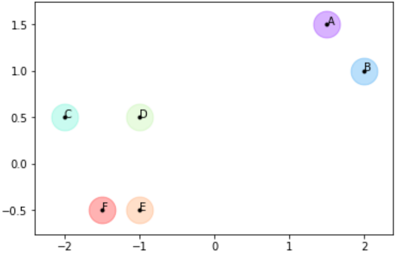




Using minimum or single-linkage clustering

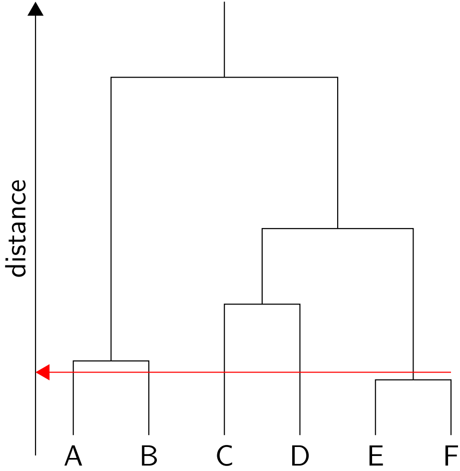
Discovering Groups - Centroid Clustering

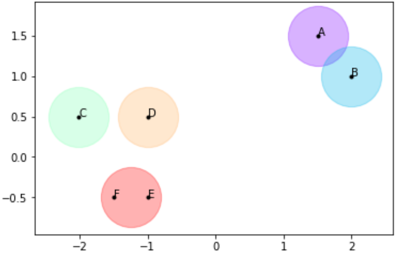




Using maximum or complete-linkage clustering

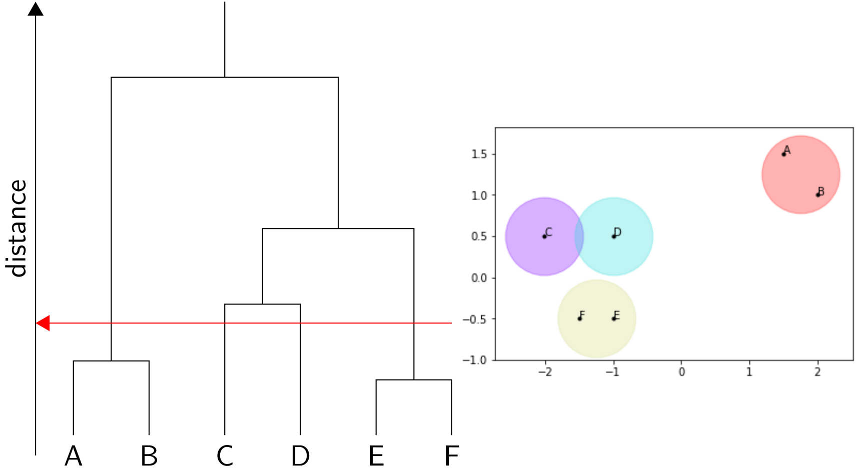
Discovering Groups - Centroid Clustering





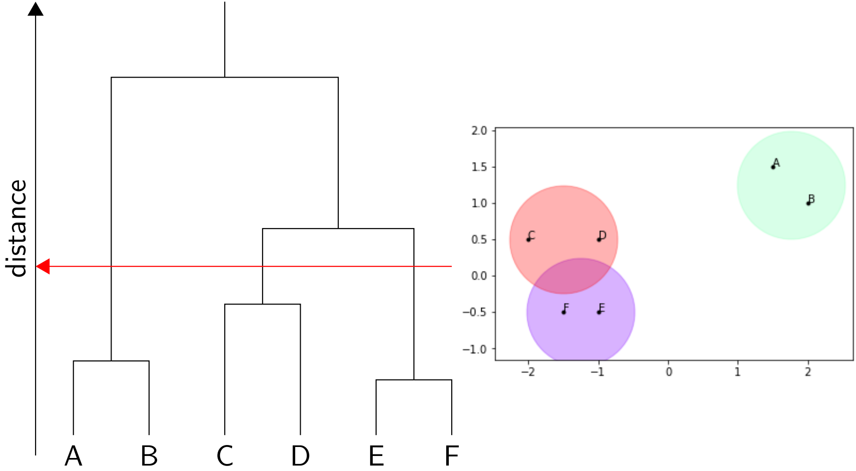
Using maximum or complete-linkage clustering

Discovering Groups - Centroid Clustering



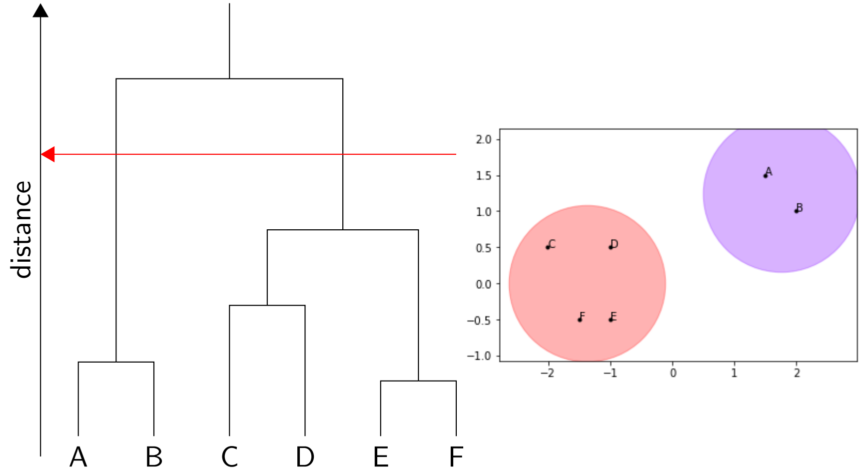
Using maximum or complete-linkage clustering

Discovering Groups - Centroid Clustering



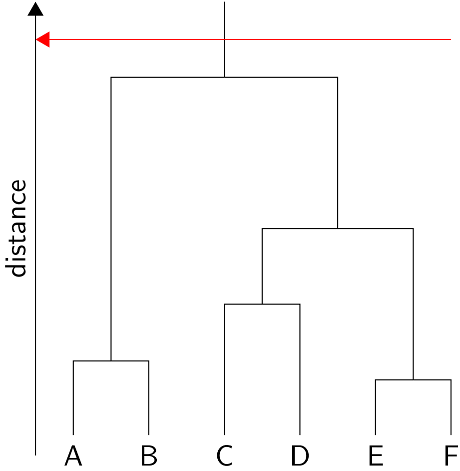
Using maximum or complete-linkage clustering

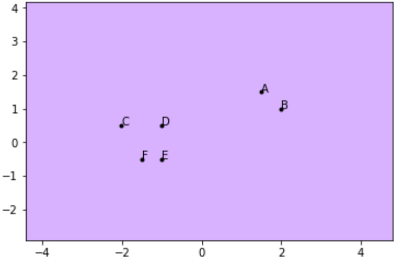
Discovering Groups - Centroid Clustering



Using maximum or complete-linkage clustering

Discovering Groups - Centroid Clustering





Using maximum or complete-linkage clustering

Discovering Groups - Hierarchical Agglomerative Clustering

Java HAC Demo

Minimum distance linkage tends to give long thin clusters

maximum distance linkage tends to give rounded clusters

Discovering Groups - Mean Shift Clustering

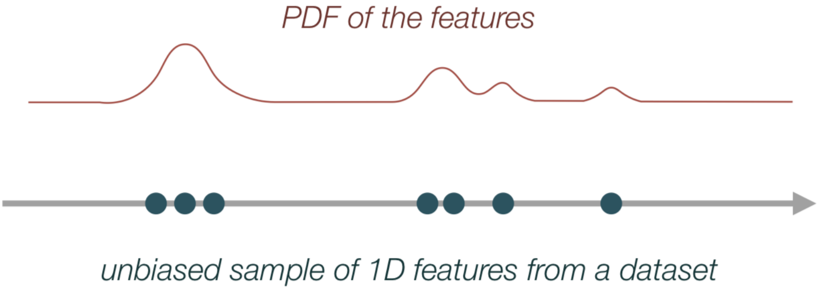
Mean shift [1] finds the *modes* of a probability density function.

This means it finds the points in feature space with the highest feature density, i.e. are the most likely given the dataset
Needs a kernel and a kernel bandwidth.

It is a hill climbing algorithm that follows the gradient of increasing density of the data

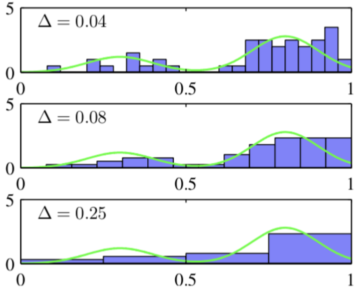
[1] Fukunaga and Hostetler IEEE Trans. Inf. Theory. 21 (1): 32–40, 1975

Discovering Groups - Mean Shift Clustering



Discovering Groups - Mean Shift Clustering

How can we estimate the PDF?
Could use a histogram, need to guess number of bins



Changing bin size affecting accuracy of probability density estimation[2]

Can be too crude

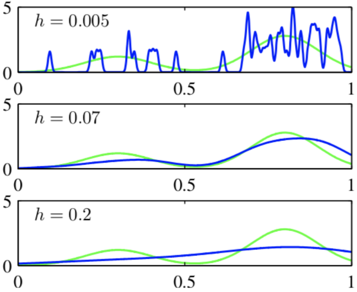
[2] C. Bishop, Pattern Recognition and Machine Learning

Discovering Groups - Mean Shift Clustering

Kernel Density Estimation (aka Parzen Window)
Gives a smooth continuous estimate

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

Where *nh* is the number of items, *d* is the dimensionality of the feature space, *K* is the kernel function, *x* is an arbitrary position in feature space, *h* is the kernel bandwidth

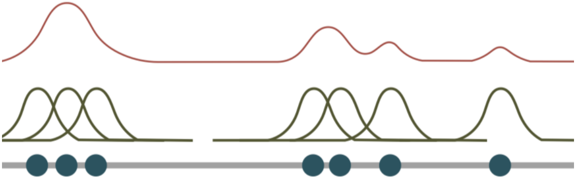


Changing bandwidth affecting accuracy of probability density estimation

Discovering Groups - Mean Shift Clustering

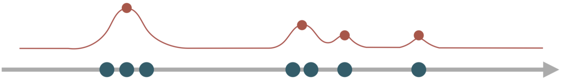
Usually use a Gaussian kernel with $\sigma = 1$

If kernel is radially symmetric, then only need profile of kernel, $k(x)$ that satisfies $K(x) = C_{k,d}k(||x||^2)$



Discovering Groups - Mean Shift Clustering

Find the modes of the probability density function (PDF), i.e. where the gradient is zero. $\Delta f(x) = 0$



Discovering Groups - Mean Shift Clustering

We model the probability density using the parzen window.

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \quad K(x) = c_{k,d} k(||x||^2)$$

We then substitute for $K(x)$, where K(x) is the kernel function, and $c_{k,d}$ is a normalisation constant. Assumes radial symmetry.

$$f(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\big(||\frac{x - x_i}{h}||^2\big)$$

Differentiating, and substituting $g(x)$ for $-k'(x)$:

$$\Delta f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) g\big(||\frac{x - x_i}{h}||^2\big) \quad g(x) = -k'(x)$$

Gives:

$$\Delta f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} g\big(||\frac{x - x_i}{h}||^2\big) \frac{\sum_{i=1}^{n} x_i g\big(||\frac{x-x_i}{h}||^2\big)}{\sum_{i=1}^{n} g\big(||\frac{x-x_i}{h}||^2\big)} - x$$

Discovering Groups - Mean Shift Clustering

$$\Delta f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} g\left(||\frac{x-x_i}{h}||^2\right) \frac{\sum_{i=1}^{n} x_i g(||\frac{x-x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x-x_i}{h}||^2)} - x$$

The first part is a probability density estimate with kernel $G(x) = x_{g,d} g(||x||^2)$

Discovering Groups - Mean Shift Clustering

$$\Delta f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} g\left(||\frac{x - x_i}{h}||^2\right) \boxed{\frac{\sum_{i=1}^{n} x_i g(||\frac{x-x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x-x_i}{h}||^2)} - x}$$

The first part is a probability density estimate with kernel $G(x) = x_{g,d} g(||x||^2)$

The second part is the mean shift, the vector that always points in the direction of maximum density.

$$m(x) = \frac{\sum_{i=1}^{n} x_i g(||\frac{x-x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x-x_i}{h}||^2)} \tag{1}$$

where m(x) is the weighted mean of the density of the data in the window determined by K and h

Discovering Groups - Mean Shift Clustering

Mean shift algorithm:

Algorithm 5: Mean Shift Procedure

Data: N data points with feature vectors X_i $i = 1 \ldots N$

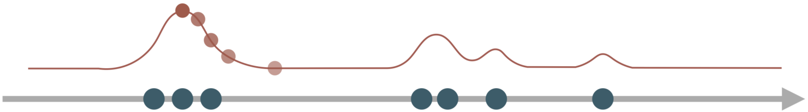
Start with initial estimate - x;

while $x_t not = x_{t+1}$ **do**

  $m_h(x_t) = \text{computeMeanShiftVect}()$;

  $x_{t+1} = x_t + m_h(x_t)$;

end

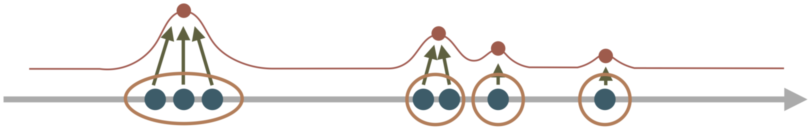


Discovering Groups - Mean Shift Clustering

For each feature vector:

- apply mean shift procedure until convergence
- store resultant mode

Set of feature vectors that converge to the same mode define the basin of attraction of that mode



Discovering Groups - Mean Shift Clustering

Mean shift is used for visual tracking and smoothing, doesn't assume a shape in the data. Only needs one parameter, h. Problems?

- it can be very difficult to select h

Discovering Groups - Summary

Clustering is a key way to understand your data.

There are many different approaches

- K Means - Need to chose K
- DBSCAN - need to choose min points and radius
- Hierarchical Agglomerative Clustering - needs a threshold or number of clusters
- Mean Shift Clustering - needs bandwidth

They are a very good way to start exploring a dataset