Machine Learning

Session 2 Unsupervised methods

-Unsupervised Methods

-Clustering

-K-means analysis and validation

Bishop Chap 9 , other in the slides

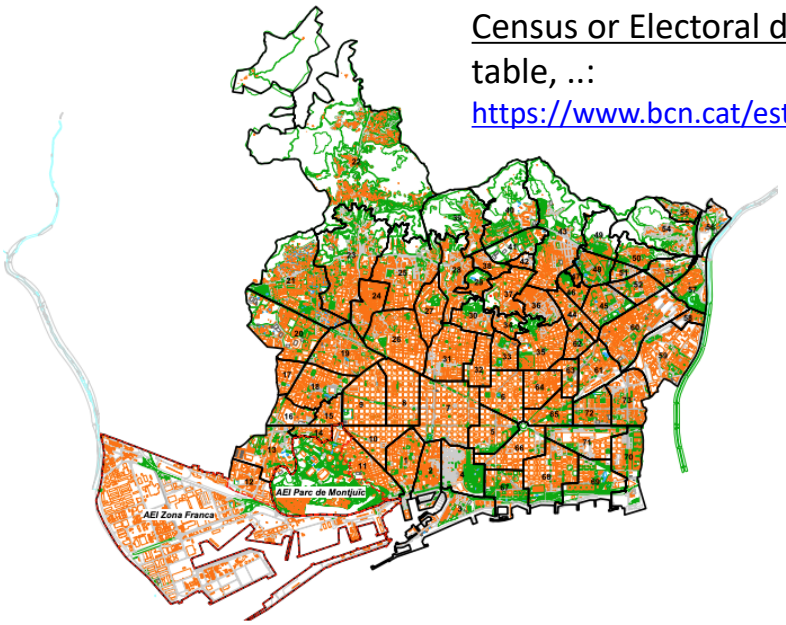
https://scikit-learn.org/dev/

Introduction

We want to place a network of shops (or Social Services Centers ..) covering all the people of BCN

Where do we place each one?

- Each citizen will be assigned (theoretically) to a shop
- This is a partition of the space



Where could we get data for doing that?

Census or Electoral data (electors by neighborhood –'barri', election table, ..:
https://www.bcn.cat/estadistica/catala/dades/telec/loc/loc19/t39.htm

This is an example of unsupervised learning

Introduction

Are the stations of pollution control placed accordingly to know how pollution affects maximum people?

https://beteve.cat/medi-ambient/contaminacio-barcelona-avui-mapa/

This is an example of unsupervised learning

Introduction

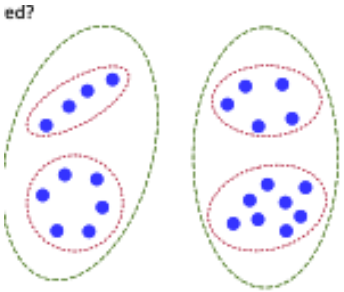
Other examples:

- Groups of people based on: gender, style, age, etc

- Moving objects on the video:

From Zemel, Urtasun, Fidelr, UofT

- How many clusters?



Unsupervised learning methods

- In the next four lectures we investigate methods that operate on unlabeled data.
- These methods are called unsupervised (training without a "teacher"- also known as ***label***) since there is no correct output
- Given a collection of *N* observations:

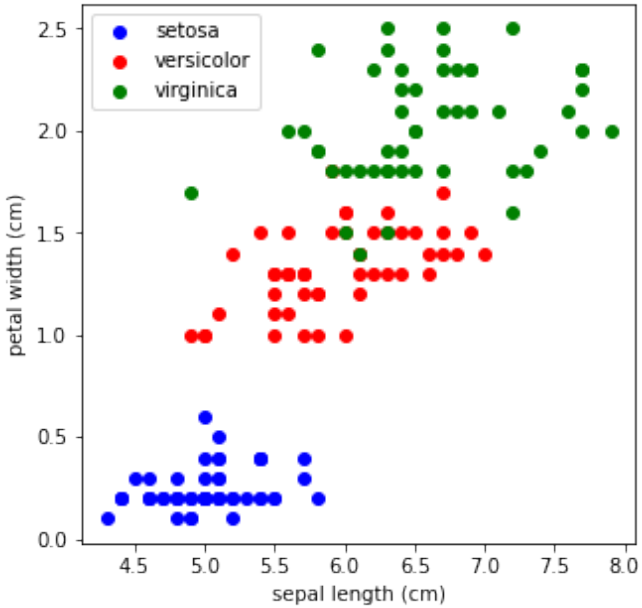
$$\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}, \qquad \text{each } \mathbf{x}^{(N)} \in \mathbb{R}^D$$

these methods attempt to build a model that captures the structure of the data

- Tasks to consider:
  - **Clustering (today)**: discover groups of similar examples
  - **Density estimation**: learn a statistical model of the input
  - **Dimensionality reduction**: useful for visualization

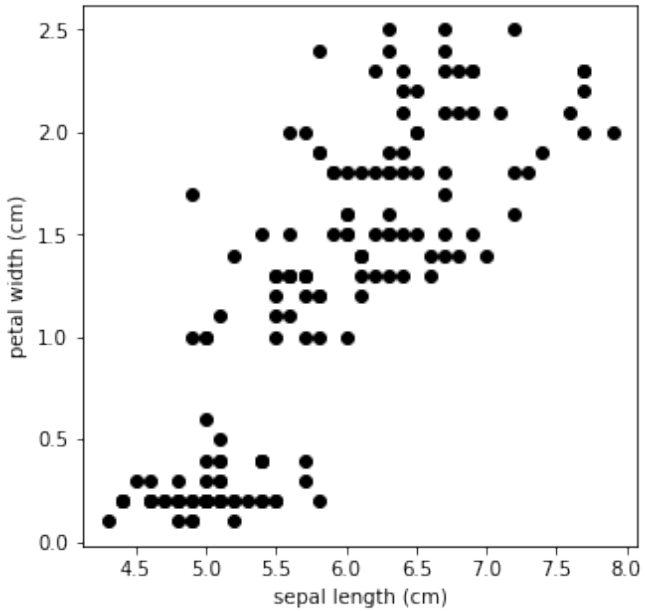
Clustering (Motivation)

- Consider the 2D feature space of the Iris dataset from last lecture



Clustering (Motivation)

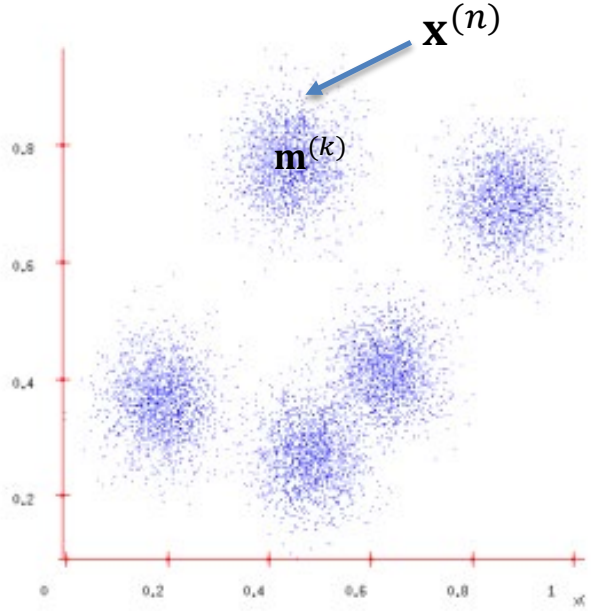
- What if we do not have information of the class memberships but know that there are three classes?



Intuition: similar data points should appear in the same class (or cluster)

Clustering

Task: group N examples into (K) clusters



Motivation:

- prediction
- lossy compression
- outlier detection

Notation:

$r_k^{(n)}$ means that the point $\mathbf{x}^{(n)}$ belongs to cluster k

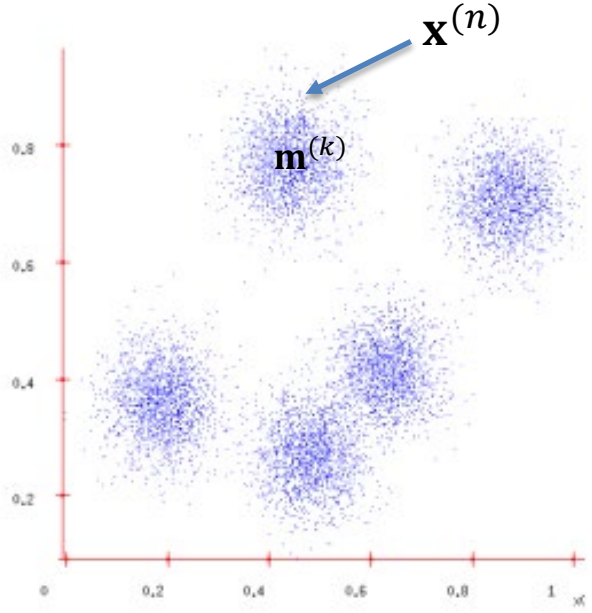
- **Goal:** Find an assignment of data points to K **clusters**

✓ Each cluster is represented by its center (or **prototype**), a vector of the same dimension $\mathbf{m}^{(k)}, k = 1, \dots K$

✓ We want that the sum of squared distances from each point to its assigned is cluster is the minimum

Clustering

Task: group N examples into (K) clusters



Motivation:

- prediction
- lossy compression
- outlier detection

Notation:

$r_k^{(n)}$ means that the point $\mathbf{x}^{(n)}$ belongs to cluster k

- Questions:
  - How many clusters? Assume K for the moment (5 in our graph)
  - What objective function should be optimized?

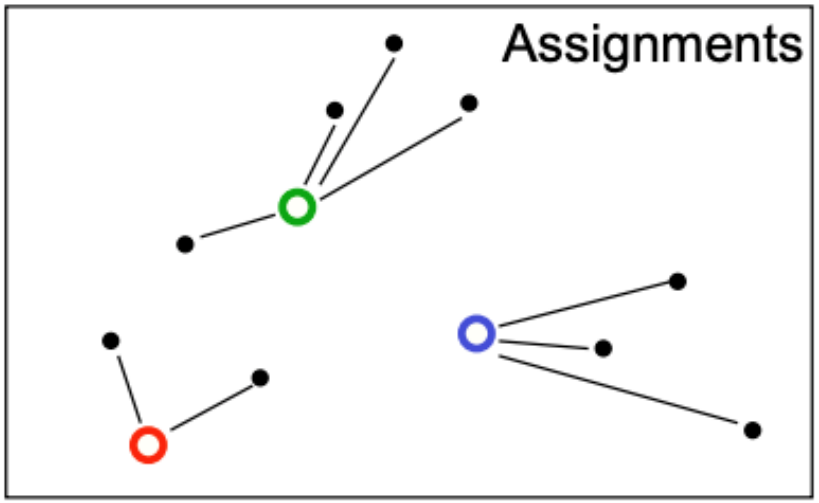
Clustering: K-means algorithm

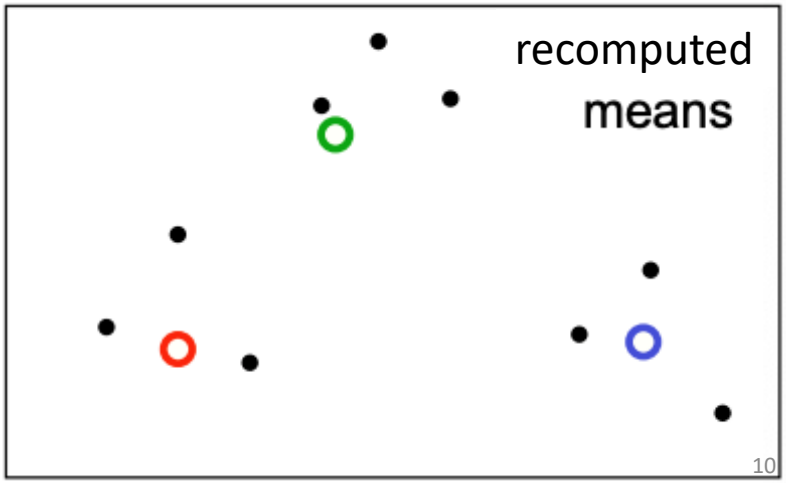
K-means: a simple iterative procedure that minimizes the sum of squared distances between each datapoint to its corresponding prototype

Inicialization: randomly initialize cluster centers

Alternate between these two steps until convergence:

- **Assignment**: Assign each data point to the closest cluster
- **Recompute**: Move each cluster center to the center of gravity of the data assigned to it

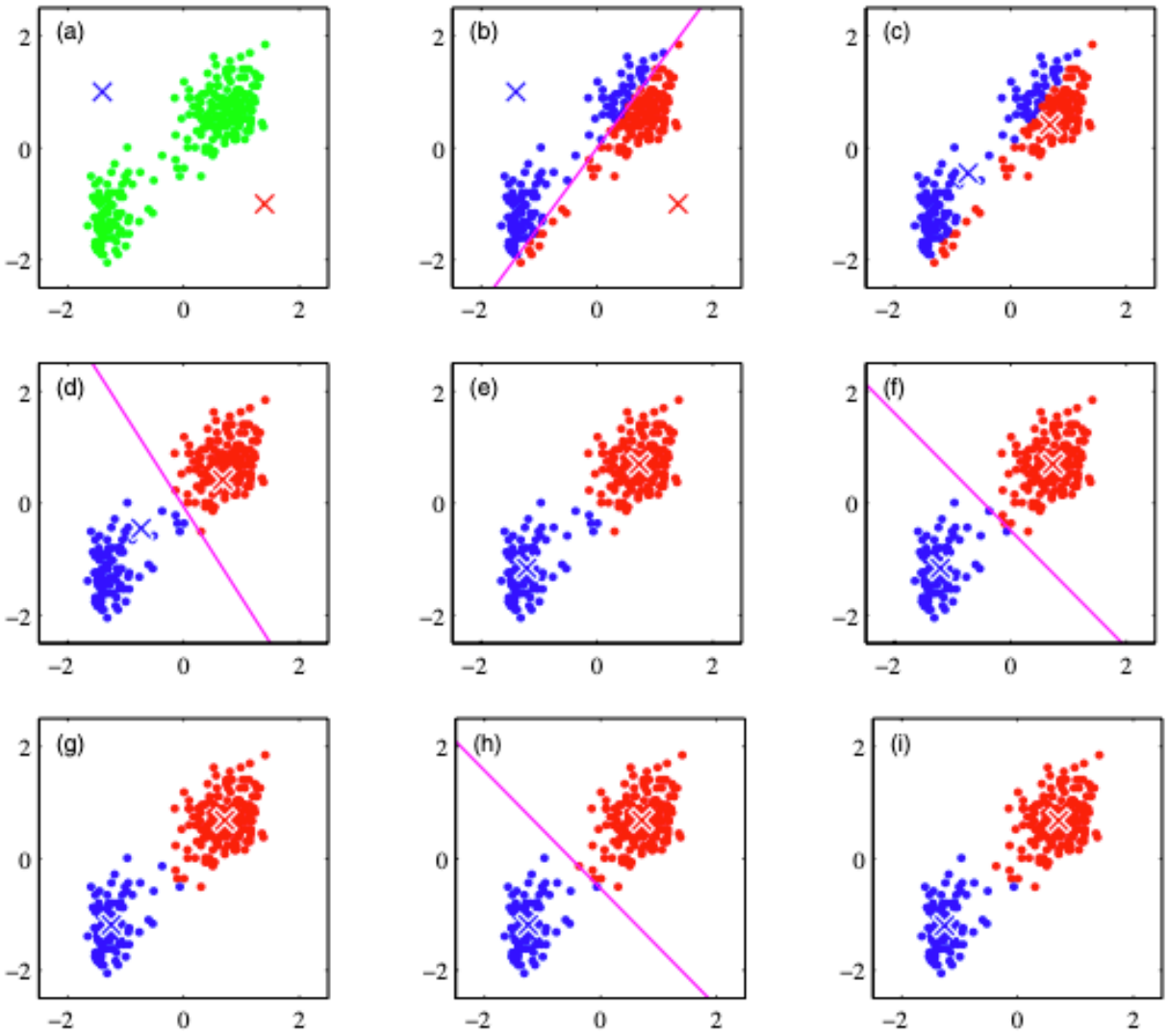




https://www.youtube.com/watch?v=Np9VuEg_aqo

Clustering: K-means algorithm

Why do this work? Take two separate points and see what happens



(From Bishop: fig 9.1)

Note: divide-line separates the points closest to one center or to the other

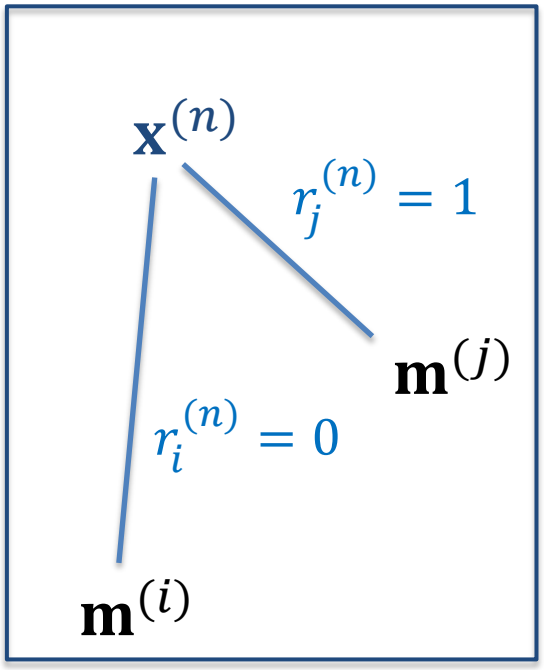
a) Dataset in 2D and initialization of cluster centers
b) Assignment step
c) Recomputed means step
…

At the end - h) and i) steps- clusters centers are stable (convergence)

Remark: data should be normalized in each coordinate or feature

Clustering: K-means algorithm

Mathematical formulation



Indicator variables, or Responsibilities:

$$r_k^{(n)} = \begin{cases} 1 & \text{if } \mathbf{x}^{(n)} \text{ assigned to cluster } k, \\ 0 & \text{otherwise} \end{cases}$$

Squared Euclidean distance between $\mathbf{a}$ and $\mathbf{b}$

$$d(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{D} (a_i - b_i)^2$$

Objective function (to be minimized):

$$J_{SSE} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathbf{m}_k} d\left(\mathbf{x}, \mathbf{m}^{(k)}\right)$$

where the means are $\mathbf{m}^{(k)} = \frac{1}{N_k} \sum_{\mathbf{x} \in \mathbf{m}_k} \mathbf{x}$.

N_k = number of elements assigned to the cluster $\mathbf{m}^{(k)}$

Clustering: K-means algorithm

- **Initialization**: Set K cluster means $\mathbf{m}^{(1)}, \mathbf{m}^{(2)}, \ldots, \mathbf{m}^{(K)}$ randomly
- Repeat until convergence (until assignments do not change):
  - **Assignment**: Each data point $\mathbf{x}^{(n)}$ assigned to nearest mean

$$\hat{k}_n = \text{argmin}_k \{ d(\mathbf{m}^{(k)}, \mathbf{x}^{(n)}) \},$$

and **Responsibilities**

$$r_k^{(n)} = \begin{cases} 1 & \text{if } \hat{k}_n = k \\ 0 & \text{if } \hat{k}_n \neq k \end{cases}$$

($\mathbf{m}^{(k)}$ will be the closest mean to $\mathbf{x}_n$)

  - **Update**: the means $\mathbf{m}^{(k)}$ are adjusted to match sample means of the data points that are responsible for:

$$\mathbf{m}^{(k)} = \frac{\sum_n r_k^{(n)} \mathbf{x}^{(n)}}{N_k}, \quad N_k = \sum_n r_k^{(n)}$$

Clustering: K-means algorithm

Points	Initial Means $\mathbf{m}^1$	...	$\mathbf{m}^K$	Assignments/Update			NewMeans $\mathbf{m}_1^1$	...	$\mathbf{m}_1^K$
$\mathbf{x}^{(1)}$	$d^2(\mathbf{x}^{(1)}, \mathbf{m}^1)$	...	$d^2(\mathbf{x}^{(1)}, \mathbf{m}^K)$	$r_1^{(1)}$	...	$r_K^{(1)}$			
...	...	...	...	...	...	...			
$\mathbf{x}^{(N)}$	$d^2(\mathbf{x}^{(N)}, \mathbf{m}^1)$	...	$d^2(\mathbf{x}^{(N)}, \mathbf{m}^K)$	$r_1^{(N)}$	...	$r_K^{(N)}$			
				$\sum_n r_1^{(n)} = N_1$	...	$\sum_n r_K^{(n)} = N_K$			
				$\mathbf{m}_1^1 = \frac{\sum_n r_1^{(n)} \mathbf{x}^{(n)}}{N_1}$	...	$\mathbf{m}_1^K = \frac{\sum_n r_K^{(n)} \mathbf{x}^{(n)}}{N_K}$			

For an efficient version using kd-trees: T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881-892, July 2002.

Application: k-means for Vector Quantization

RGB- 3Dplot

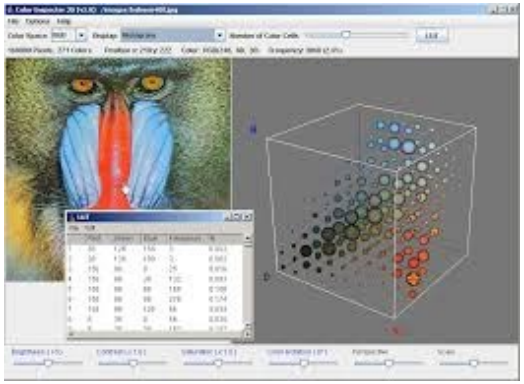


Figure from Bishop

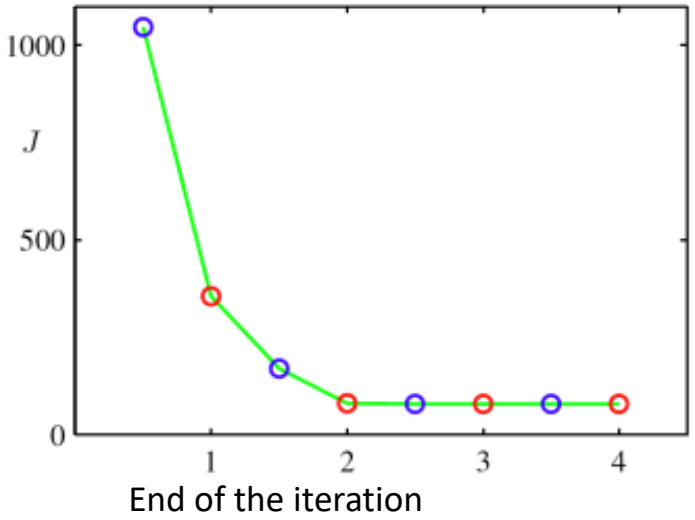
- Data set is the set of pixels of the image
- Each pixel is a vector in $\mathbf{R}^3$, the (r, g, b) of each pixel
- K-means is applied to the data set of pixels in $\mathbf{R}^3$
- Compressed representation= color of each mean plus cluster index for each pixel

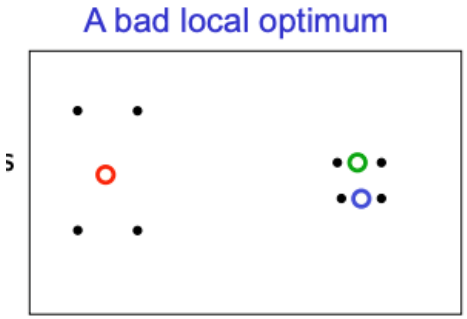
K-means: analysis

- Convergence of the algorithm:
  - Whenever an assigment is changed (blue color) , we reduce:

$$J_{SSE} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathbf{m}^{(k)}} d\left(\mathbf{x}, \mathbf{m}^{(k)}\right)$$

  - Whenever a cluster center is moved (red color), J_{SSE} is reduced
  - Test for convergence: if the assignments do not change in the assignment step, we have converged (to a local minimum at least)
  - The plot below shows the algorithm converges after the 3rd iteration



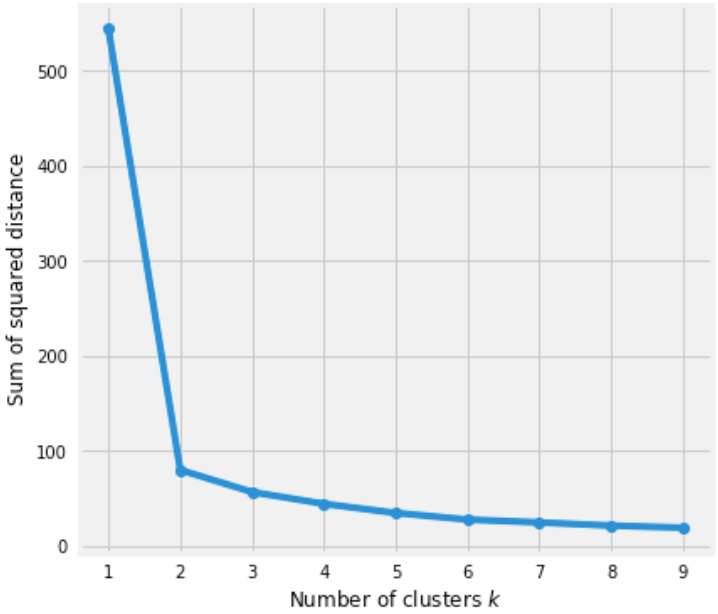


- Solutions:
  - Try many starting points
  - Merge two nearby clusters and split the big cluster into two

K-means: evaluation

- We do not have a ground truth so there is not a right answer. We need to explain why our solution is good.
- Two criteria:
  - Elbow Method: gives an idea about the number K of clusters.
  - Silhouette analysis https://orange.biolab.si/blog/2017/03/17/k-means-silhouette-score/

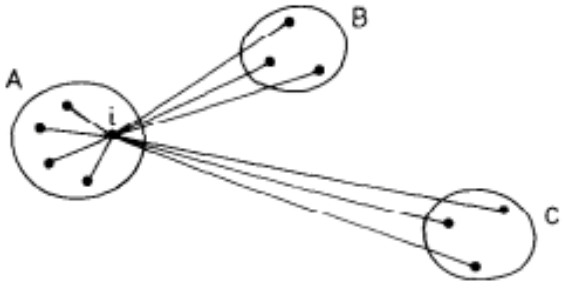
➢ **Elbow Method**: We calculate J_{SSE} in the convergence for different values of K.



- In this case $K = 2$ is not a bad choice.
  Additional clusters have little additional value
- Sometimes finding the optimal K is not easy (e.g. if the curve is smooth)

K-means: evaluation

- **Silhouette Analysis**: provides a succinct graphical representation of how well each element $\mathbf{x}^{(i)}$ has been assigned to the K clusters



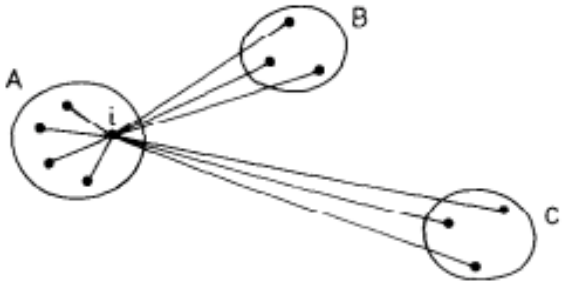
Let I denote the cluster of $\mathbf{x}^{(i)}$ and C_i the set of point contained in it

$$a(i) = \frac{1}{|C_I| - 1} \sum_{\mathbf{x}^{(j)} \in \mathbf{m}^{(I)}} d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

mean dissimilarity between $\mathbf{x}^{(i)}$ and the other points of **the same cluster**. The smaller the value, the better the assignment

K-means: evaluation

- **Silhouette Analysis**: provides a succinct graphical representation of how well each element $\mathbf{x}^{(i)}$ has been assigned to the K clusters



Let I denote the cluster of $\mathbf{x}^{(i)}$ and C_i the set of point contained in it

$$b(i) = \min_{J \neq I} \frac{1}{|C_J| - 1} \sum_{\mathbf{x}^{(j)} \in \mathbf{m}^{(J)}} d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

minimum mean dissimilarity between $\mathbf{x}^{(i)}$ and the other points of **different clusters**.

K-means: evaluation

- **Silhouette Analysis**: provides a succinct graphical representation of how well each element $\mathbf{x}^{(i)}$ has been assigned to the K clusters

  - $a(i) = \frac{1}{|\mathbf{m}^{(k)}|-1} \sum_{\mathbf{x}^{(j)} \in \mathbf{m}^{(k)}} d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$
  - $b(i) = \min_{k \neq i} \frac{1}{|\mathbf{m}^{(k)}|-1} \sum_{\mathbf{x}^{(j)} \in \mathbf{m}^{(k)}} d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$

  We want $b(i) - a(i)$ to be as large as possible

  - Silhouette of $\mathbf{x}^{(i)}$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) \ll b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) \gg b(i) \end{cases}$$

K-means: evaluation

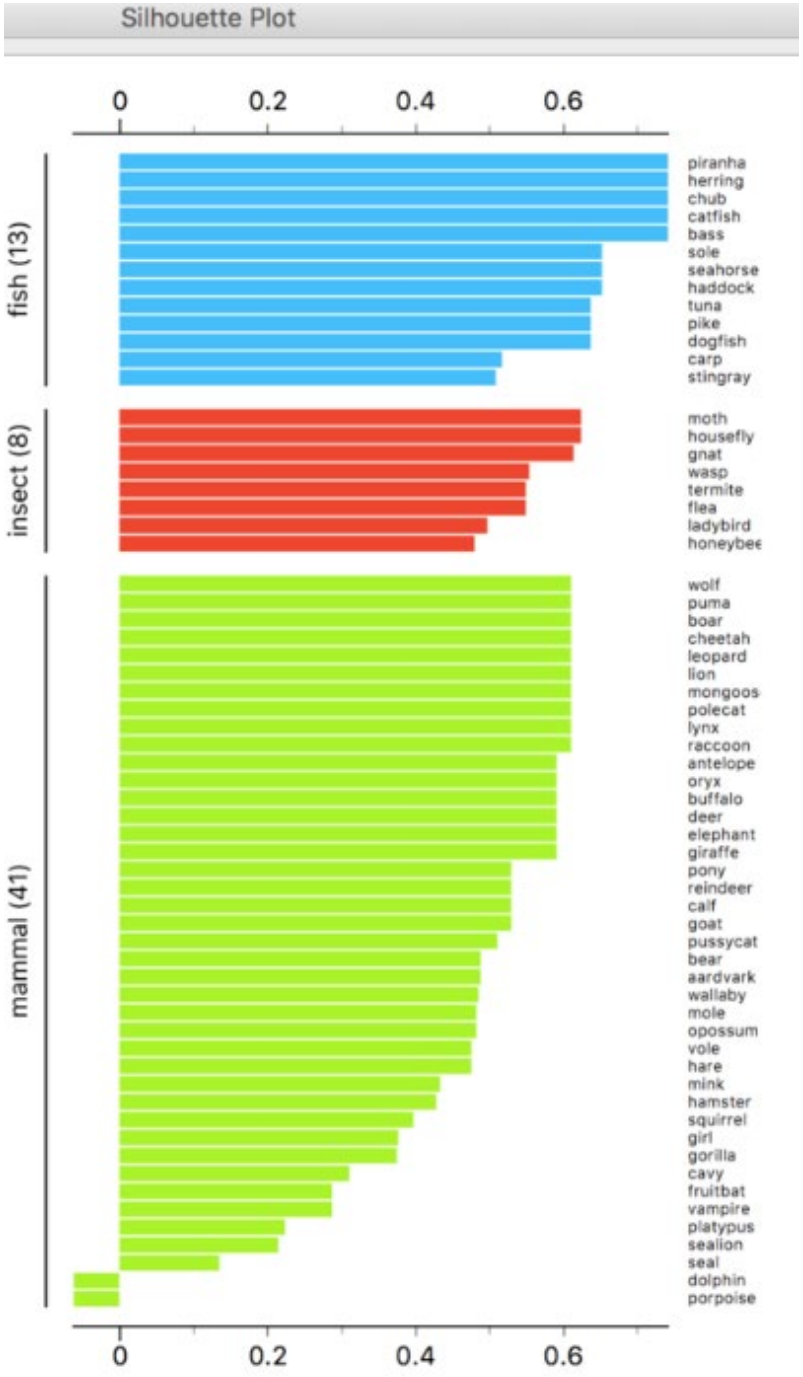
- **Silhouette Analysis**: provides a succinct graphical representation of how well each element $\boldsymbol{x}^{(i)}$ has been classified in K clusters.

  Silhouette of $\mathbf{x}^{(i)}$

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) \ll b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) \gg b(i) \end{cases}$$

Then $-1 \leq s(i) \leq 1$:

- $s(i) \approx 1$: Cluster i well matched and different from the neighboring
- $s(i) \approx 0$: Point $\boldsymbol{x}^{(i)}$ alone in the cluster
- $s(i) \approx -1$: Point i should belong to a neighbouring cluster



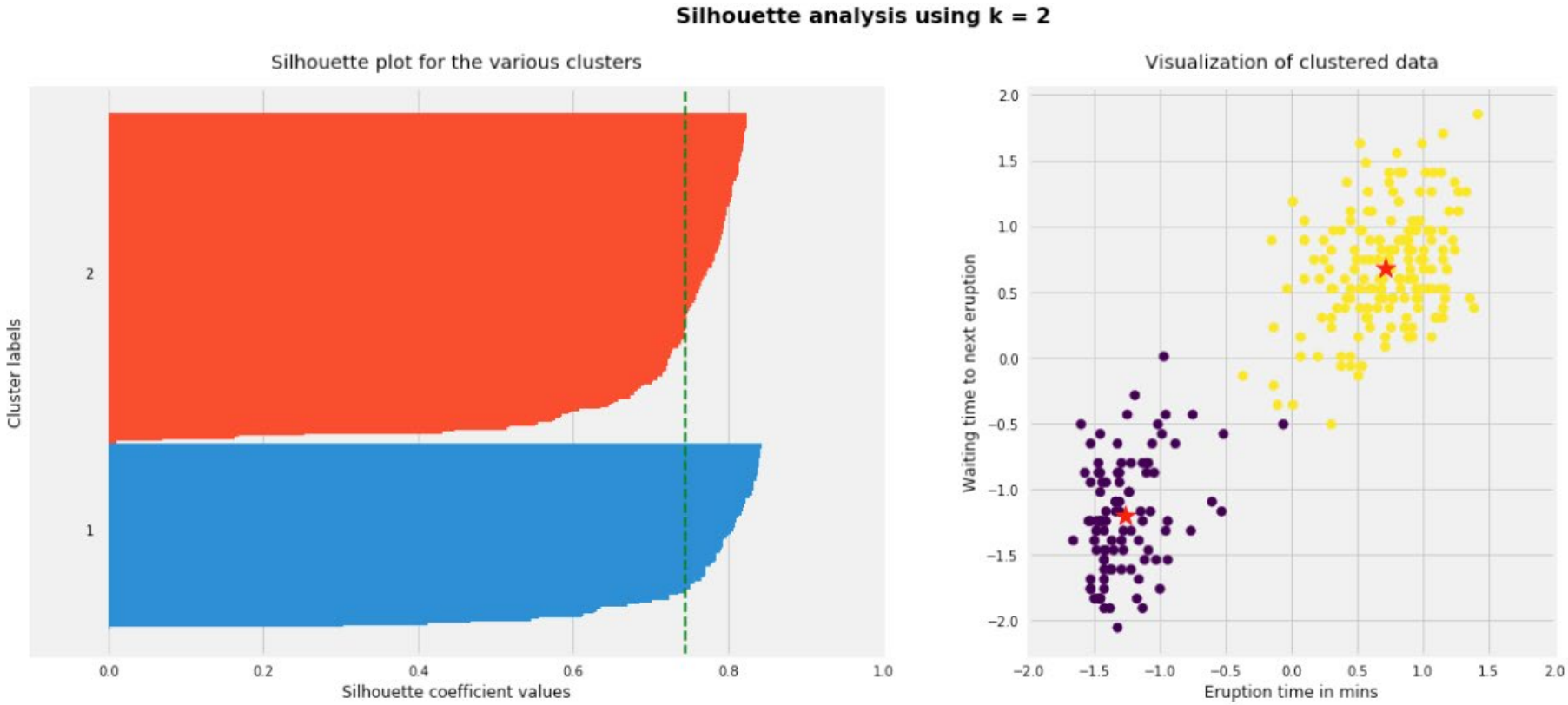
K-means: evaluation

- **Silhouette scores**:
  - Elements are animals from the zoo
  - 3 clusters: fish, insects and mammals
  - At the bottom: silhouette identifies dolphin and porpoise as outliers in the group of mammals.
  - In general, scores larger than 0.6 are good.

See the video: https://orange.biolab.si/blog/2017/03/17/k-means-silhouette-score/

K-means: evaluation

Geyser's Eruptions Segmentation

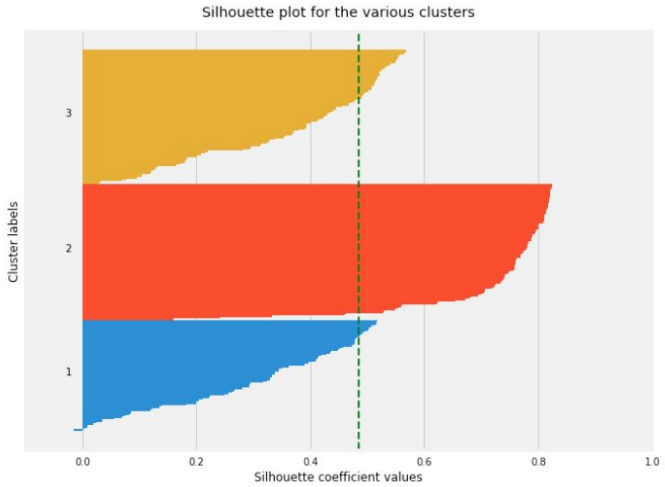


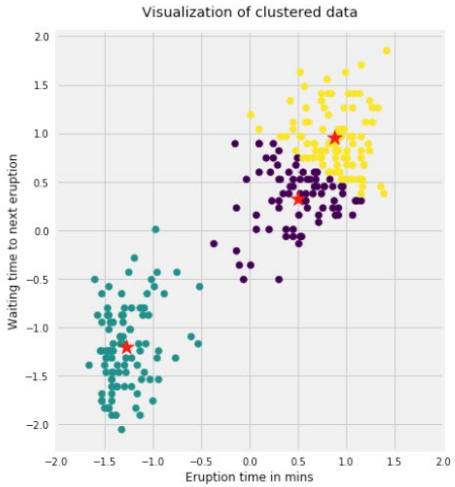
FROM: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

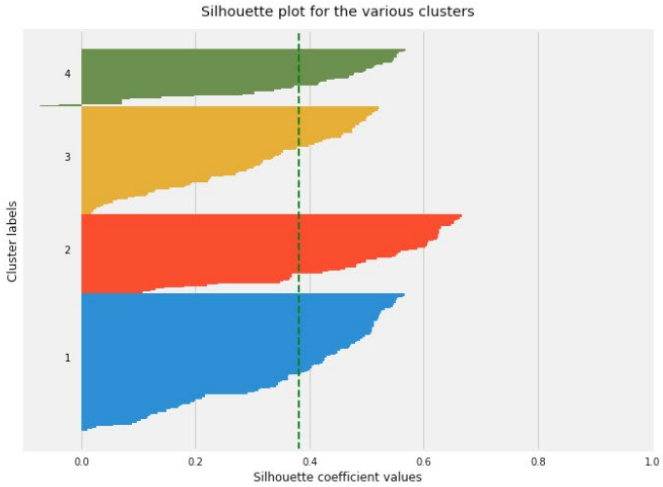
Mean Silhouette score is 0.75

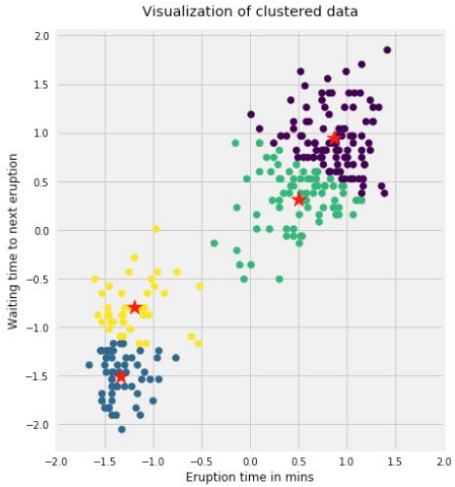
K-means: evaluation

Geyser's Eruptions Segmentation









More clusters result in smaller mean Silhouette scores (worse clusterings)

FROM: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

K-means: what can go wrong?

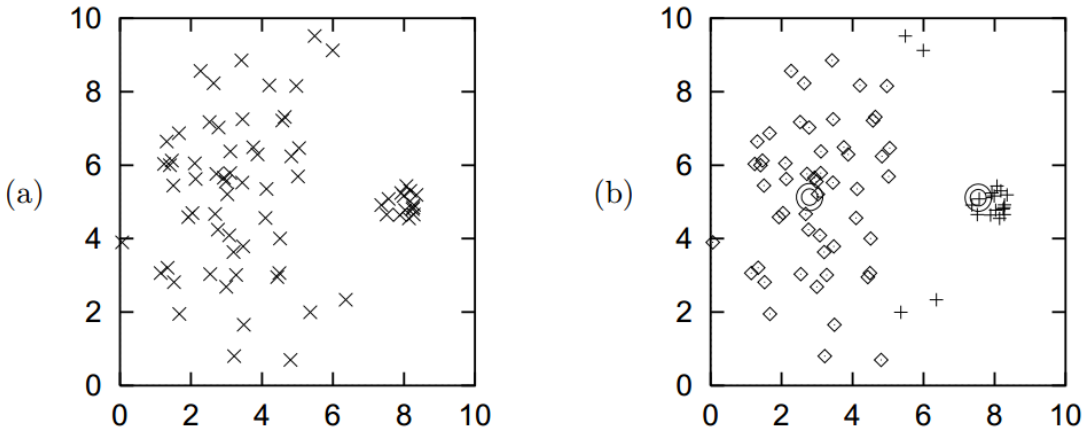


Figure 20.5. K-means algorithm for a case with two dissimilar clusters. (a) The "little 'n' large" data. (b) A stable set of assignments and means. Note that four points belonging to the broad cluster have been incorrectly assigned to the narrower cluster. (Points assigned to the right-hand cluster are shown by plus signs.)

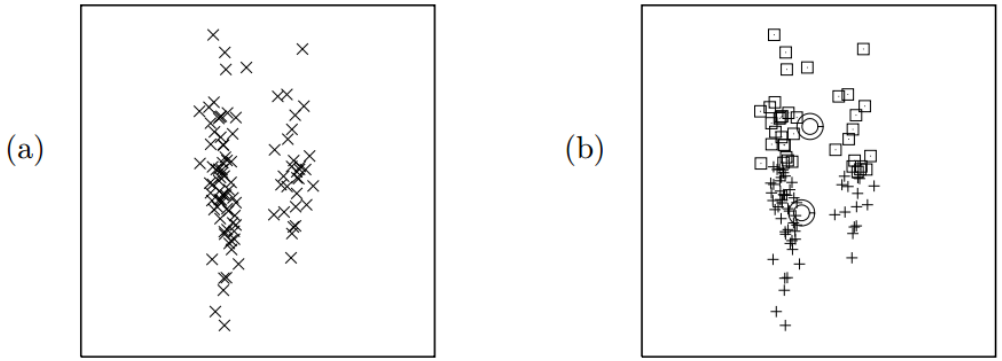


Figure 20.6. Two elongated clusters, and the stable solution found by the K-means algorithm.

Generalizing K-means

- A basic clustering algorithm : K-means
  - K-means is a 'hard' rather than a 'soft' algorithm:
  - Points are assigned to exactly one cluster
  - All points assigned to a cluster are equals in that cluster, regardless of their location

- ***Soft* K-means** introduces a soft degree of assignment to each of the means
  - Requires a new parameter β (*stiffness*)
  - In the limit of infinite stiffness, the algorithm becomes the same as the original K-means

Generalizing K-means

- **Soft K-means**
  - Assignment step
    - Each data point $\mathbf{x}_n$ is given a soft degree of assignment to the nearest mean

$$r_k^{(n)} = \frac{\exp(-\beta d(\boldsymbol{m}^k, \mathbf{x}_n))}{\sum_{k'} \exp(-\beta d(\boldsymbol{m}^{k'}, \mathbf{x}_n))}$$

    - The sum of the K responsibilities for the n-th point is 1
    - Instead of a *'min'* over distances, the assignment is a *'soft-min'*

Generalizing K-means

- **Soft K-means**
  - Update step (identical to the original K-means)
    - The means $\mathbf{m}_k$ are adjusted to match the sample means of the data points that they are responsible for

$$\boldsymbol{m}^{(\boldsymbol{k})} = \frac{\sum_n r_k^{(n)} \boldsymbol{x}^{(n)}}{N_k},$$

where R_k is the total responsibility of mean k

$$N_k = \sum_n r_k^{(n)}$$

Generalizing K-means

- **Soft K-means**
  - We can associate a length scale $\sigma \equiv 1/\sqrt{\beta}$ with it Example:

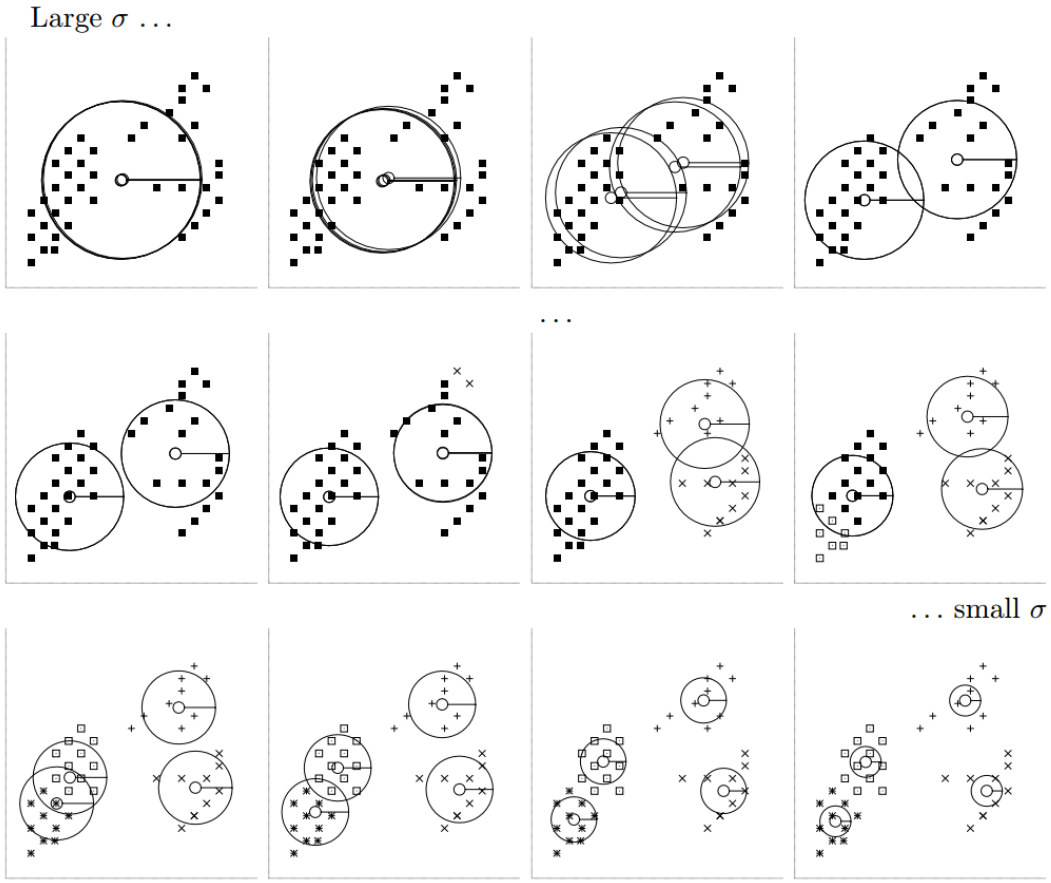


Figure 20.8. Soft K-means algorithm, version 1, applied to a data set of 40 points. $K = 4$. Implicit lengthscale parameter $\sigma = 1/\beta^{1/2}$ varied from a large to a small value. Each picture shows the state of all four means, with the implicit lengthscale shown by the radius of the four circles, after running the algorithm for several tens of iterations. At the largest lengthscale, all four means converge exactly to the data mean. Then the four means separate into two groups of two. At shorter lengthscales, each of these pairs itself bifurcates into subgroups.

Conclusions

- **Unsupervised learning**: data in the absence of labels / target
- **Clustering**: grouping N examples into K groups. Similar examples belong to the same group
  - Useful for prediction, lossy compression and outlier detection
  - Several algorithms: K-means, hierarchical clustering, etc (next chapter)
- K-means: iteration of assignment and update steps
  - Objective function to minimize: $J_{SSE} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathbf{m}^{(k)}} d\left(\mathbf{x}, \mathbf{m}^{(k)}\right)$ requires distances of datapoints to their assigned cluster centers
  - Convergence when assignments are stable
  - Evaluation: Elbow and Silhouette analysis
- K-means fails when
  - Very dissimilar clusters
  - Elongated clusters
- soft K-means as a generalization