Lecture 16 : Clustering

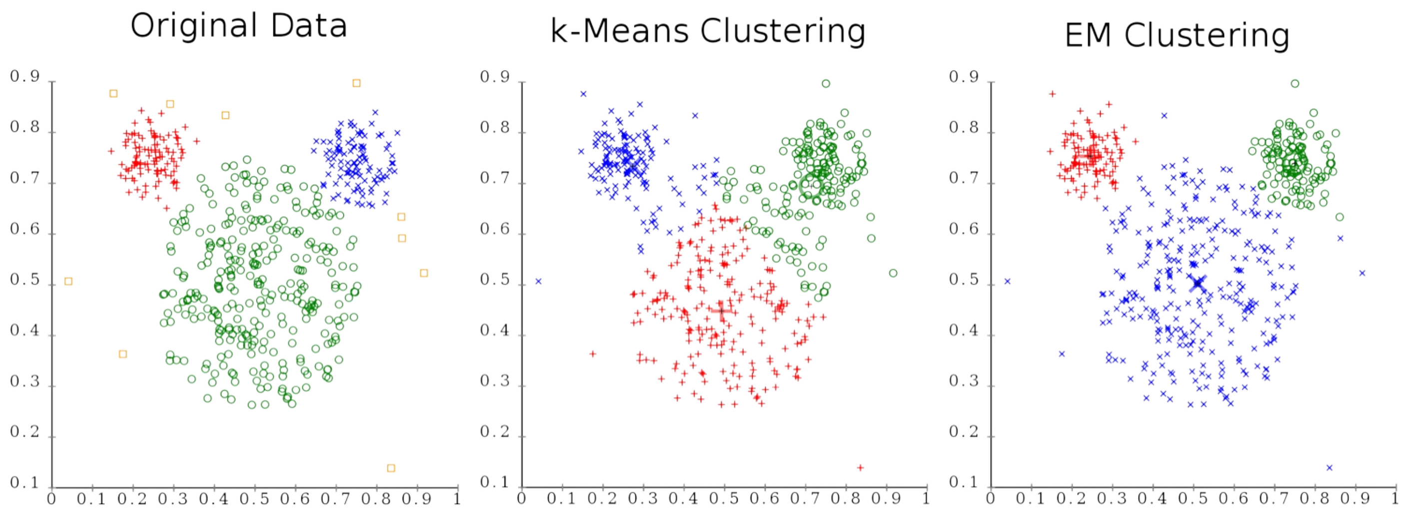
Clustering

Given points $\underline{x}_i \in \mathbb{R}^p$, $i = 1, 2, \ldots, n$, group them into clusters $C_1, \ldots, C_K$ so that

- any two points in same cluster are "close"
- any two points in different clusters are 'distant'
- "hard" clustering — each $\underline{x}_i$ is in one and only one cluster

  or "soft" clustering — each $\underline{x}_i$ may be in multiple clusters

Different cluster analysis results on "mouse" data set:



For instance, let $\underline{\mu}_k \in \mathbb{R}^p$, $k = 1, \ldots, K$, be a prototypical point for the k^{th} cluster.

Then we want to assign the $\underline{x}_i$'s to clusters so that the sum of distances (or squared distances) from each data point to its assigned cluster to be minimized

K-means clustering

$$\{\hat{C}_1, \dots, \hat{C}_K\} = \underset{\{C_1, \dots, C_K\}}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{\underline{x}_i \in C_k} \| \underline{x}_i - \mu_k \|^2$$

where $\mu_k = \frac{1}{|C_k|} \sum_{\underline{x}_i \in C_k} \underline{x}_i$ = cluster center

$$= \underset{\{C_1, \dots, C_K\}}{\operatorname{argmin}} \underbrace{\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{\underline{x}_i, \underline{x}_j \in C_k} \| \underline{x}_i - \underline{x}_j \|^2}_{=: \text{ obj (objective)}}$$

Lloyd's algorithm

Start with initial set of K means (centers) $\underline{\mu}_1^{(1)}, \underline{\mu}_2^{(1)}, \ldots, \underline{\mu}_K^{(1)}$

for $t = 1, 2, 3, \ldots$

for $i = 1, 2, \ldots, n$

find nearest (center) mean to $\underline{x}_i$

$$\hat{k}_i = \underset{k}{\operatorname{argmin}} \|x_i - \underline{\mu}_k^{(t)}\|^2$$

end

for $k = 1, \ldots, K$

$$\hat{C}_k^{(t+1)} = \{\underline{x}_i : \hat{k}_i = k\}$$ (set cluster estimates)

$$\mu_k^{(t+1)} = \frac{1}{|\hat{C}_k^{(t+1)}|} \sum_{i \in \hat{C}_k^{(t+1)}} x_i$$ (find each cluster mean)

end

end

Generally this algorithm converges, but not to the optimal clustering — results depend on initial clustering

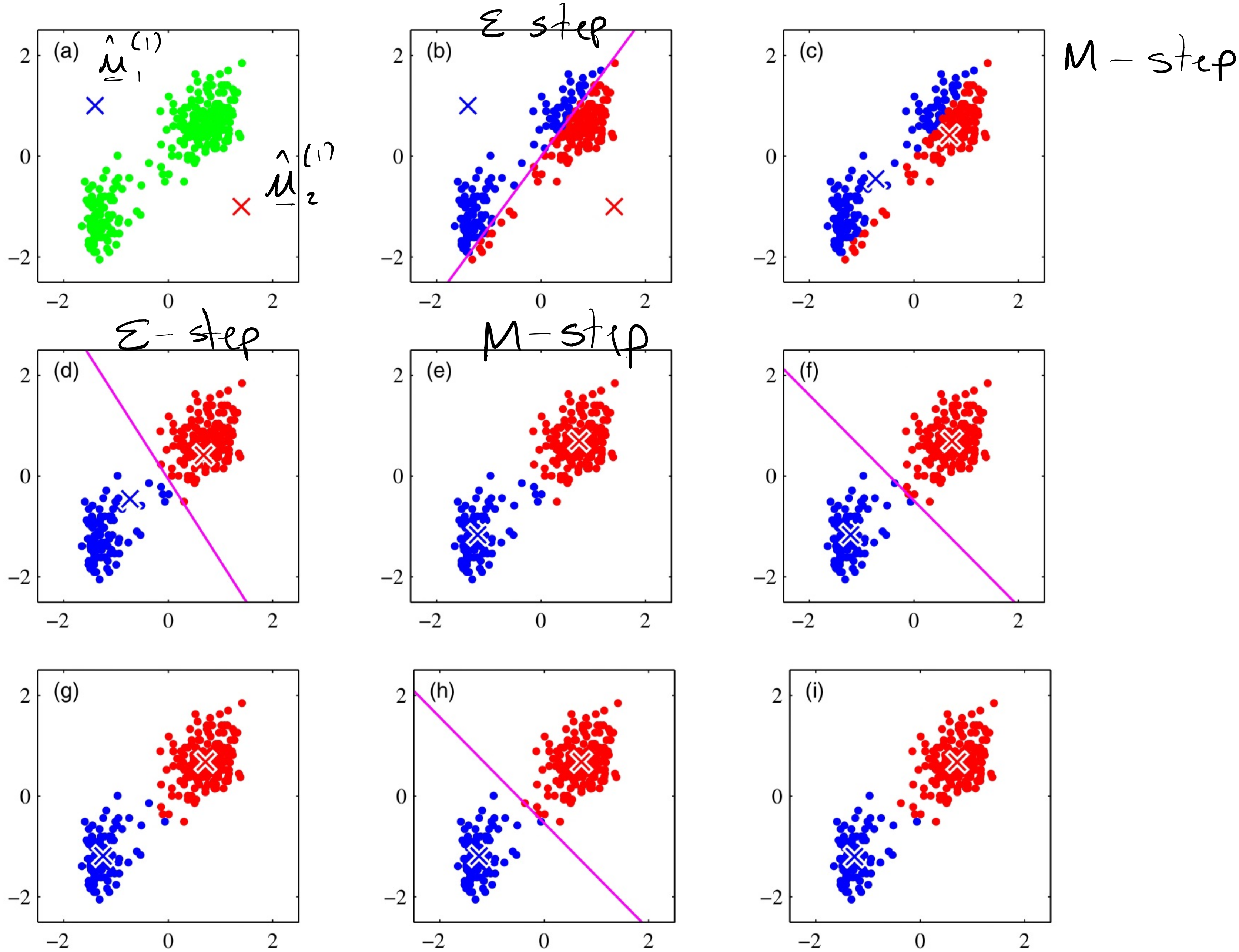
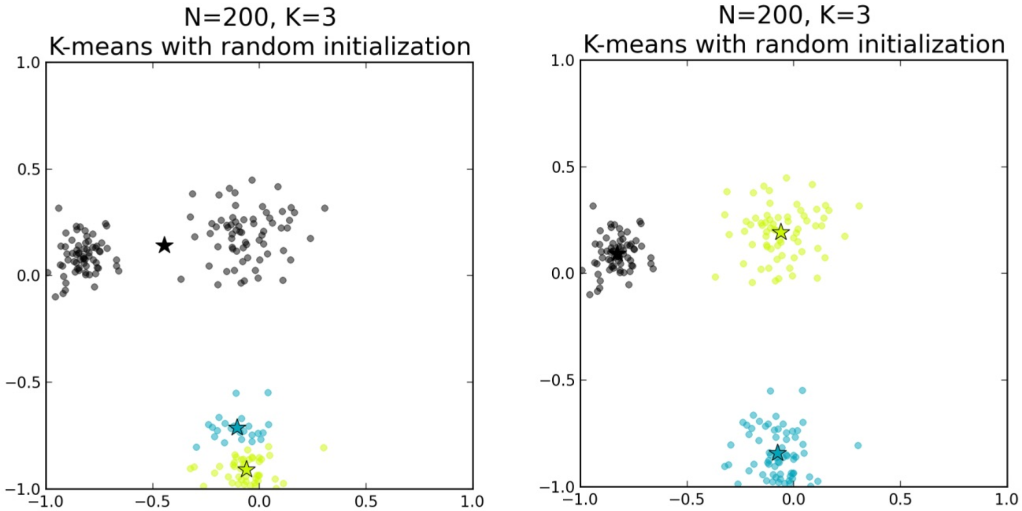


Figure 9.1 Illustration of the K-means algorithm using the re-scaled Old Faithful data set. (a) Green points denote the data set in a two-dimensional Euclidean space. The initial choices for centres $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are shown by the red and blue crosses, respectively. (b) In the initial E step, each data point is assigned either to the red cluster or to the blue cluster, according to which cluster centre is nearer. This is equivalent to classifying the points according to which side of the perpendicular bisector of the two cluster centres, shown by the magenta line, they lie on. (c) In the subsequent M step, each cluster centre is re-computed to be the mean of the points assigned to the corresponding cluster. (d)–(i) show successive E and M steps through to final convergence of the algorithm.

Initialization:



$\Longleftarrow$ Importance of good initialization

Forgy: choose k random $\underline{x}_i$'s as initial points

Kmeans++:

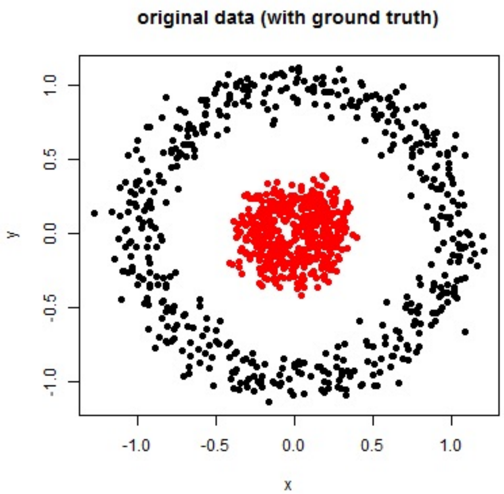
- choose one $\underline{x}_i$ uniformly at random as $\mu_1^{(1)}$
- for $k = 2, 3, \dots, K$

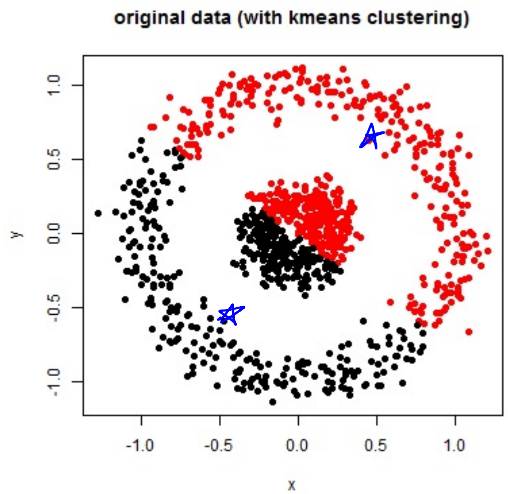
  find remaining data point furthest from any $\mu_j^{(1)}$, $j < k$, selected so far,

  make this $\mu_k^{(1)}$

Provably good! Solution found by Kmeans++ may not be optimal, but

$$\text{obj}(\text{kmeans}++) \le O(\log K)\,\text{obj}(\text{optimal})$$





⟸ k-means not always sensible.

Can we do better with kernels?

Regular k-means

Start with initial set of K means

$\underline{\mu}_1^{(1)}, \underline{\mu}_2^{(1)}, \ldots, \underline{\mu}_K^{(1)}$

for $t = 1, 2, 3, \ldots$

for $i = 1, 2, \ldots, n$

find nearest mean to $\underline{x}_i$

$$\hat{k}_i = \underset{k}{\arg\min} \; \| x_i - \underline{\mu}_k^{(t)} \|^2$$

end

for $k = 1, \ldots, K$

$$\hat{C}_k^{(t+1)} = \{ \underline{x}_i : \hat{k}_i = k \}$$

$$\underline{\mu}_k^{(t+1)} = \frac{1}{|\hat{C}_k^{(t)}|} \sum_{i \in \hat{C}_k^{(t)}} x_i$$

Towards Kernel k-means

$$\hat{k}_i = \underset{k}{\arg\min} \; \| \phi(\underline{x}_i) - \phi(\underline{\mu}_k^{(t)}) \|_2^2$$

$$= \underset{k}{\arg\min} \; \underbrace{\phi(\underline{x}_i)^T \phi(\underline{x}_i)}_{k(\underline{x}_i, \underline{x}_i)} - 2\phi(\underline{x}_i)^T \phi(\underline{\mu}_k^{(t)}) + \phi(\underline{\mu}_k^{(t)})^T \phi(\underline{\mu}_k^{(t)})$$

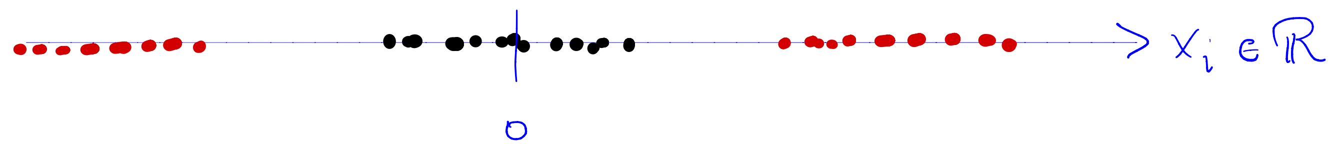
$$\phi(\underline{\mu}_k^{(t)}) = \frac{1}{|C_k^{(t)}|} \sum_{\underline{x}_j \in C_k^{(t)}} \phi(\underline{x}_j) .$$

$$\Rightarrow \phi(\underline{x}_i)^T \phi(\underline{\mu}_k^{(t)}) = \frac{1}{|C_k^{(t)}|} \sum_{\underline{x}_j \in C_k^{(t)}} \phi(\underline{x}_i)^T \phi(\underline{x}_j)$$

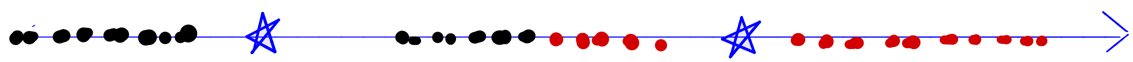
$$= \frac{1}{|C_k^{(t)}|} \sum_{\underline{x}_j \in C_k^{(t)}} k(\underline{x}_i, \underline{x}_j)$$

$$\phi(\underline{\mu}_k^{(t)})^T \phi(\underline{\mu}_k^{(t)}) = \frac{1}{|C_k^{(t)}|^2} \sum_{x_j, x_j' \in C_k^{(t)}} \phi(\underline{x}_j)^T \phi(\underline{x}_j')$$

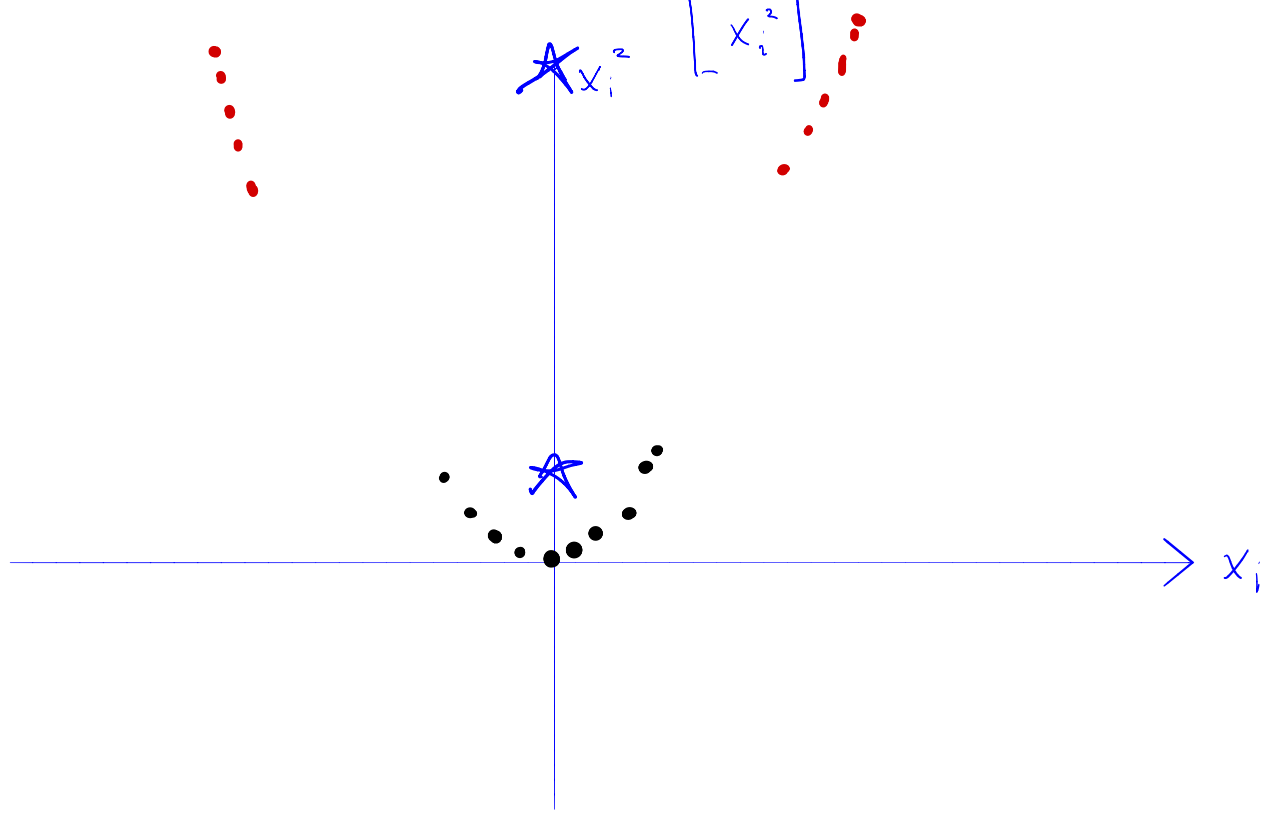
$$= \frac{1}{|C_k^{(t)}|^2} \sum_{x_j, x_j' \in C_k^{(t)}} k(\underline{x}_j, \underline{x}_j')$$



Regular k-means



Kernel k-means, $\phi(x_i) = \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix}$



Kernel k-means

Start with initial set of K cluster assignments $\hat{C}_1^{(1)}, \hat{C}_2^{(1)}, \ldots, \hat{C}_K^{(1)}$

for $t = 1, 2, 3, \ldots$

for $i = 1, 2, \ldots, n$

\# find nearest mean to $\underline{x}_i$

$$\hat{k}_i = \underset{k}{\operatorname{argmin}} \; \frac{1}{|C_k^{(t)}|^2} \sum_{\underline{x}_j, \underline{x}_{j'} \in C_k^{(t)}} k(\underline{x}_j, \underline{x}_{j'}) - \frac{2}{|C_k^{(t)}|} \sum_{\underline{x}_j \in C_k^{(t)}} k(\underline{x}_i, \underline{x}_j) + k(\underline{x}_i, \underline{x}_i)$$

end

for $k = 1, \ldots, K$

$$\hat{C}_k^{(t+1)} = \{ \underline{x}_i : \hat{k}_i = k \}$$

end

end