Lecture 20 - Mar 18

Clustering
Expectation-Maximization (EM) Algorithm

References

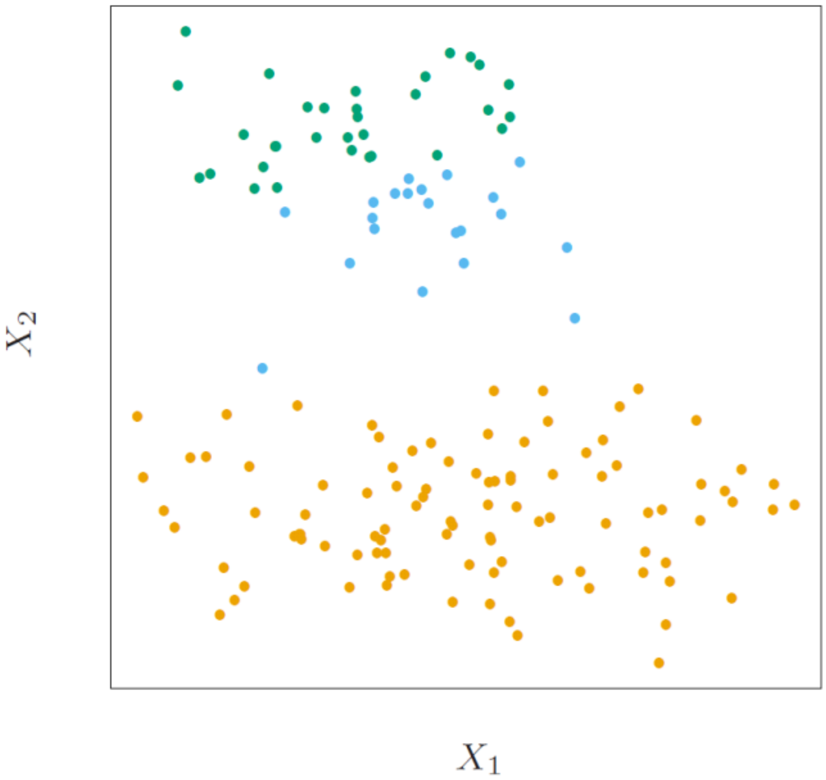
Data Mining and Machine Learning
Ch 13 - Representation-based Clustering

Elements of Statistical Learning
14.3.6 K-means
14.3.7 Gaussian Mixtures as Soft K-Means Clustering

Cluster Analysis

AKA data sesmentation has a variety of goals.

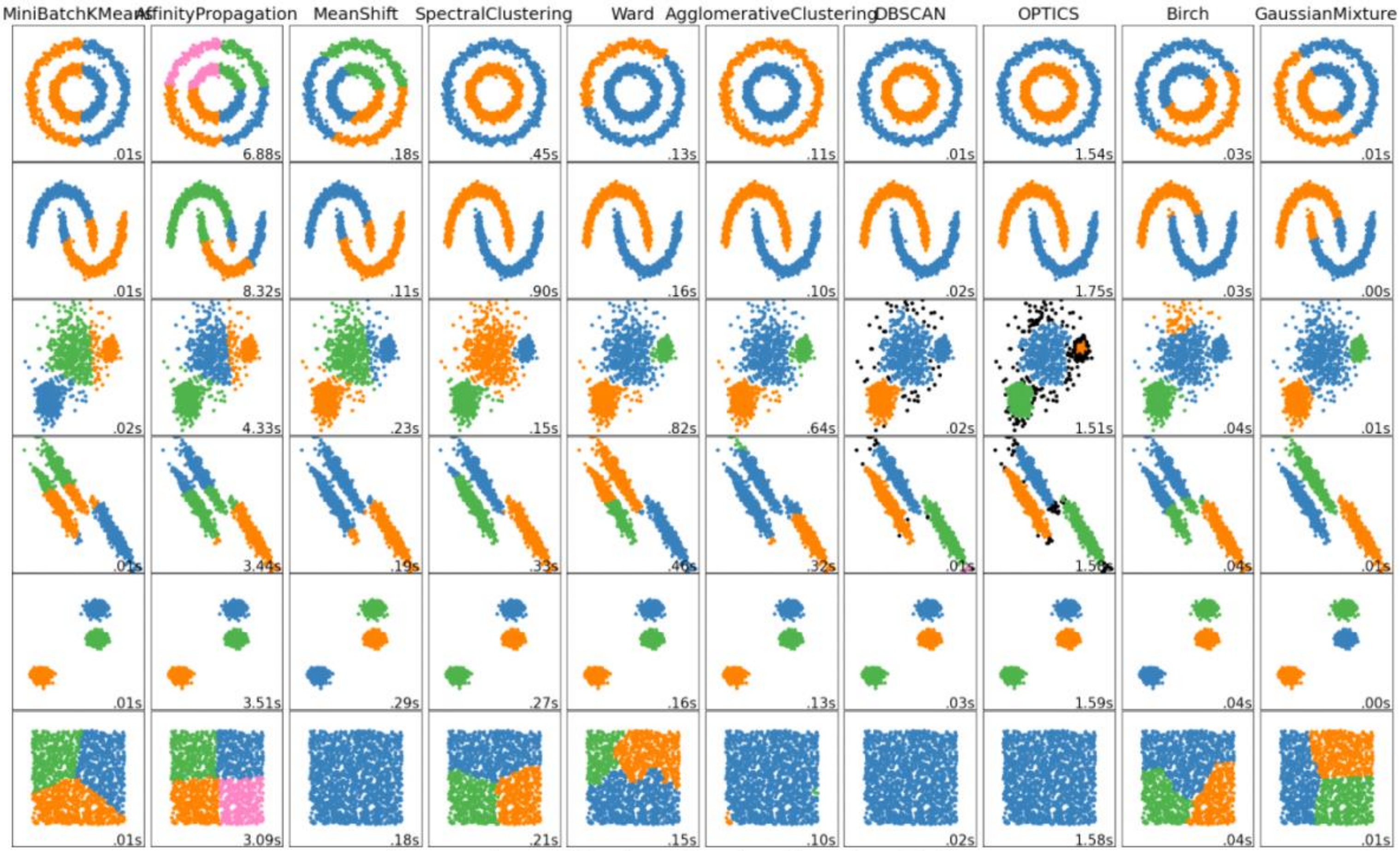
All group or segment objects into "clusters" where points inside are similar but different from points in other clusters.



- Sometimes the goal is to find a natural hierarchy of clusters

- Also used for descriptive statistics regarding whether or not there exist distinct subgroups with substantially different properties

Clustering Algorithms



A comparison of the clustering algorithms in scikit-learn

https://scikit-learn.org/stable/modules/clustering.html

Types: Combinatorial, Mixture modeling, mode seeking

work directly on the observed data

assumes data is an i.i.d. sample from a population described by a pdf characterized by a parameterized model taken as a mixture of component density functions: each describing one cluster

"bump hunters" attempt to directly estimate distinct modes of the pdf as centers of clusters

Representation-based Clustering

Let $X \in \mathbb{R}^{n \times d}$ be a dataset of points $x_i \in \mathbb{R}^d$

Representative-based clustering partitions X into k clusters...

$$\mathcal{C} = \{C_1, \dots, C_k\}$$

For each cluster C_i, there exists a representative point that summerizes the cluster

↳ e.g. the centroid (mean) μ_i of points in C_i

$$\mu_i = \frac{1}{n_i} \sum_{x_j \in C_i} x_j \quad \text{where } n_i = |C_i|$$

Two common methods: K-means + Expectation-Maximization (EM) algorithms

For a given clustering C, we need a scoring function to evaluate it

The SSE scoring function is

$$SSE(C) = \sum_{i=1}^{k} \underbrace{\sum_{x_j \in C_i} \|x_j - \mu_i\|^2}_{\text{SS of difference between cluster pts + cluster centroid}}$$

Sum over all clusters

$$C^* = \underset{C}{\arg\min} \{ SSE(C) \}$$

K-means minimizes SSE via a greedy iterative approach (risks reaching local min)

K-means initializes centroids randomly + iterates 2 steps

1. Assign points to nearest centroid's cluster
2. Compute centroid as mean of points in each cluster

(2) Compute centroid as mean of points in each cl

Stop when centroids stop moving.

Often, people run K-means several times due to random initializations + keep the best result (wrt. SSE)

K-means creates convex-shaped clusters.

Algorithm 13.1

```
Input: $X, k, \varepsilon$
1. Randomly initialize k centroids $\mu_1^0, \dots, \mu_k^0 \in \mathbb{R}^d$
while $\sum_{i=1}^{k} \|\mu_i^t - \mu_i^{t-1}\| > \varepsilon$:
    for $x_j \in X$:
        $i^* = \arg\min_i \{\|x_j - \mu_i^{t-1}\|^2\}$
        $C_{i^*} = C_{i^*} \cup \{x_j\}$
    for $C_i \in C$
        $\mu_i^t = \frac{1}{n_i} \sum_{x_j \in C_i} x_j$
```

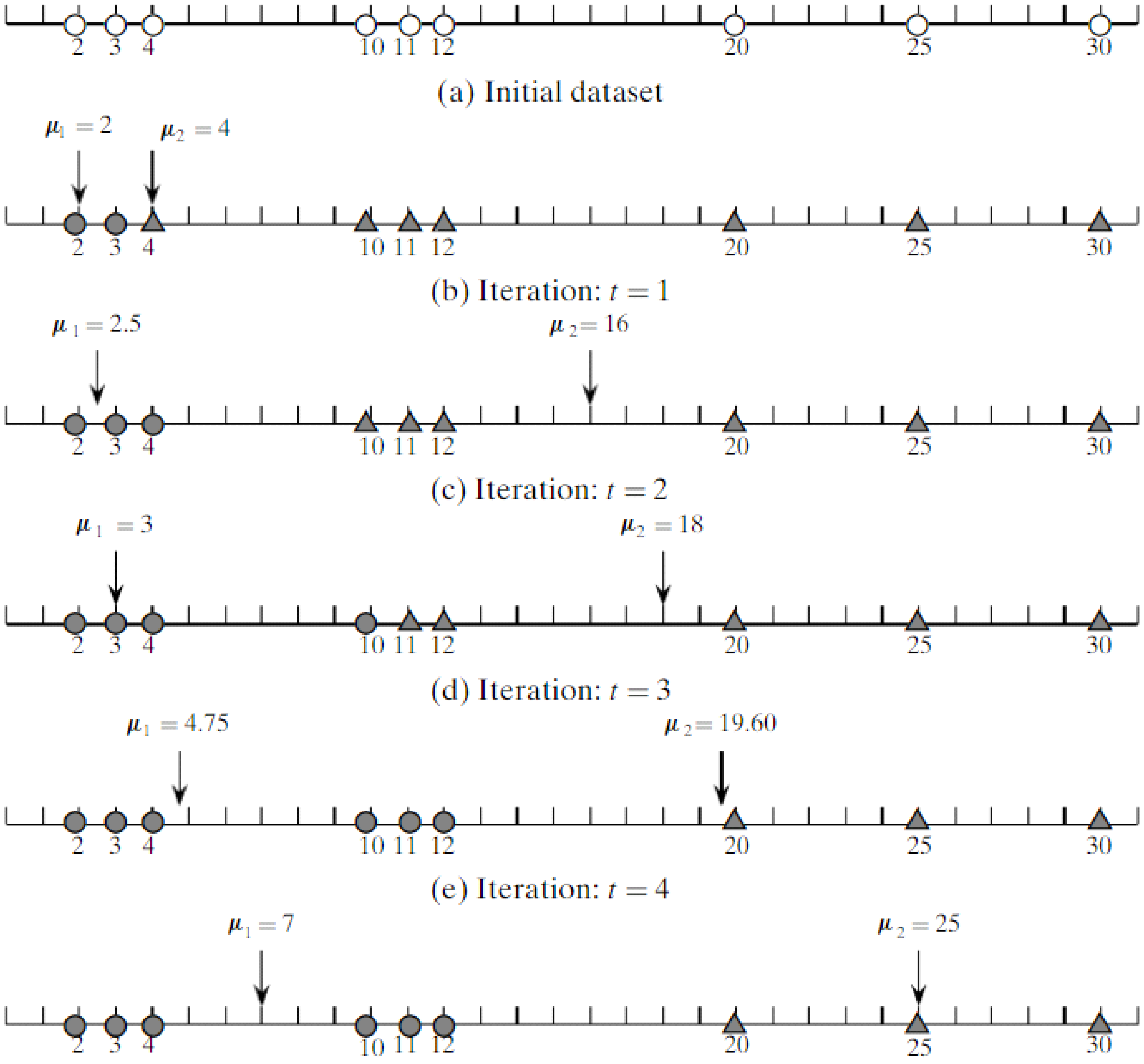
often, sample uniformly in each feature

while centroids move substantially in re-calculations

assign points to cluster with nearest centroid

re-calculate centroids with new clusters

K-Means in 1D



(a) Initial dataset

(b) Iteration: $t = 1$

(c) Iteration: $t = 2$

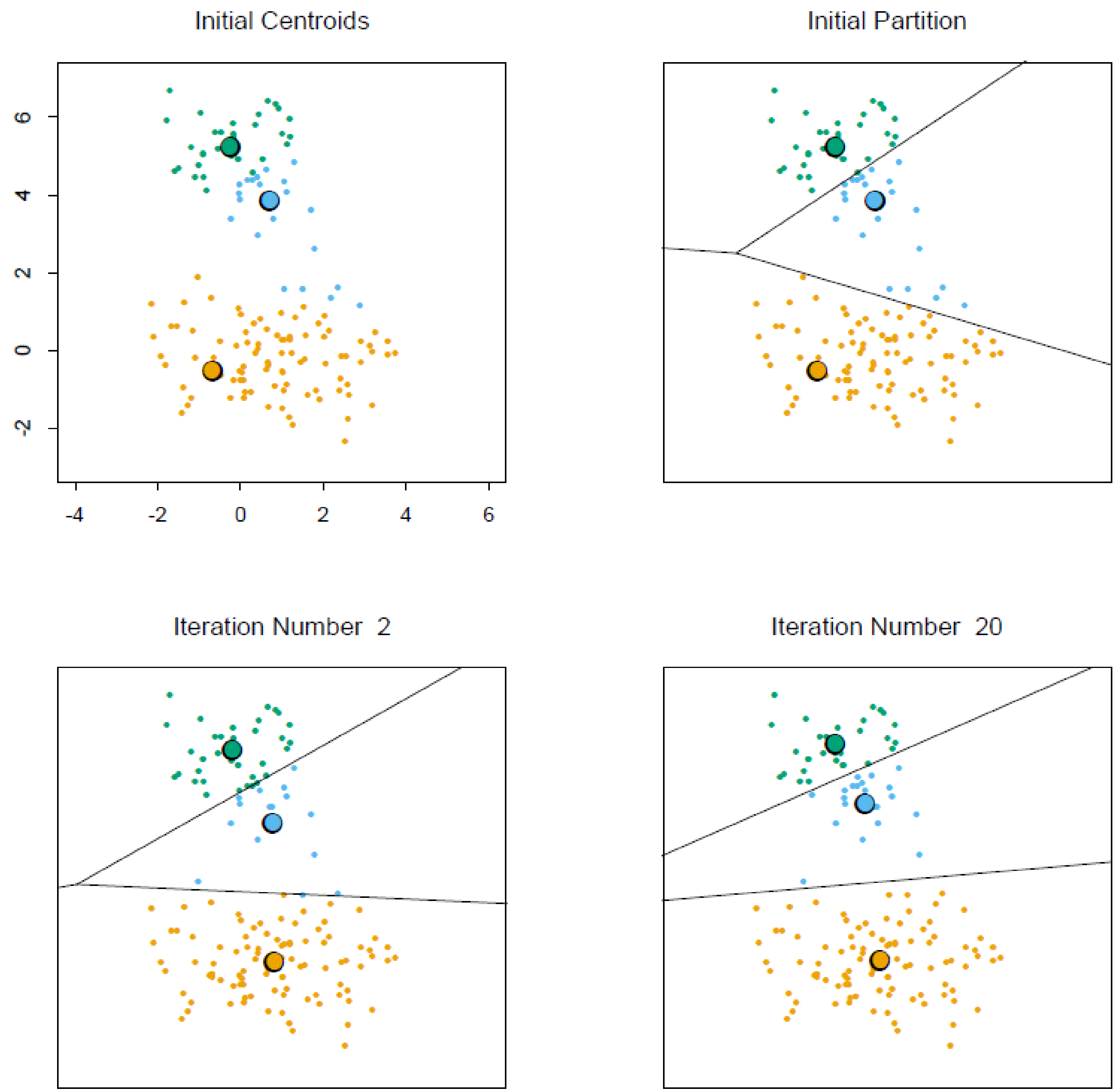
(d) Iteration: $t = 3$

(e) Iteration: $t = 4$

(f) Iteration: $t = 5$ (converged)

Figure 13.1. K-means in one dimension.

K-Means in 2D


Initial Centroids
Initial Partition
Iteration Number 2
Iteration Number 20
6
4
2
0
-2
-4
-2
0
2
4
6