Machine Learning

Chapter 2 Clustering

Dr. Minhhuy Le

EEE, Phenikaa University

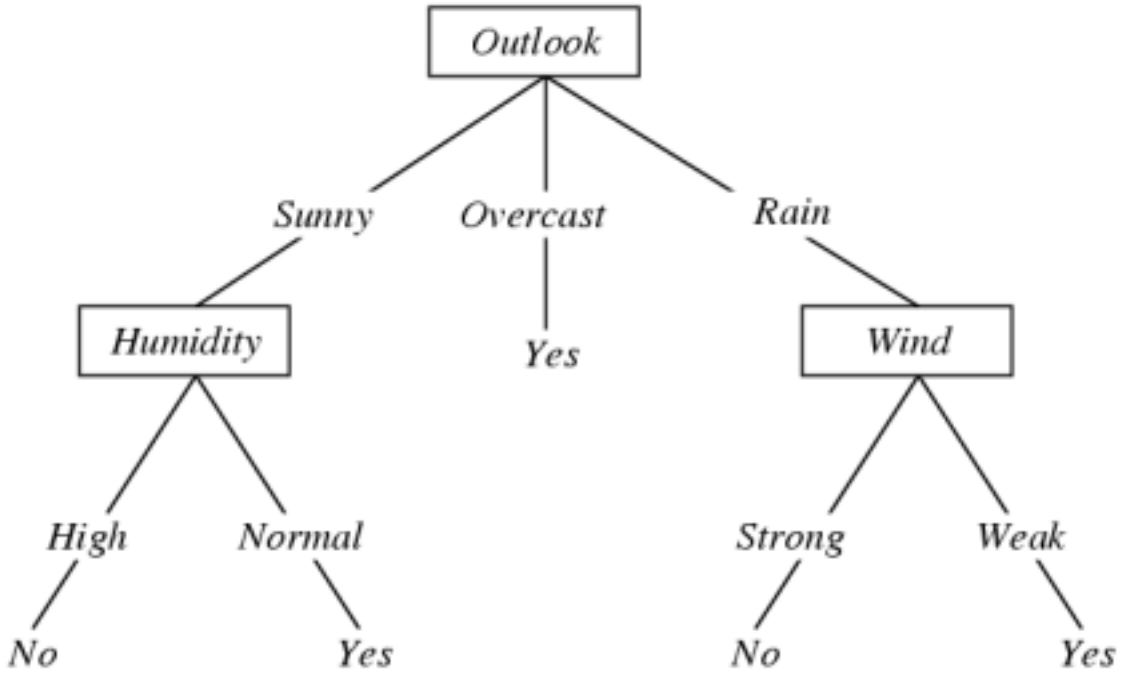
Chapter 2: Decision Tree

1. Decision tree review
2. Clustering intuition
3. K-means algorithm
4. Summary

1. Random Forest Review

Example:

f: <Outlook, Temperature, Humidity, Wind> => PlayTennis?

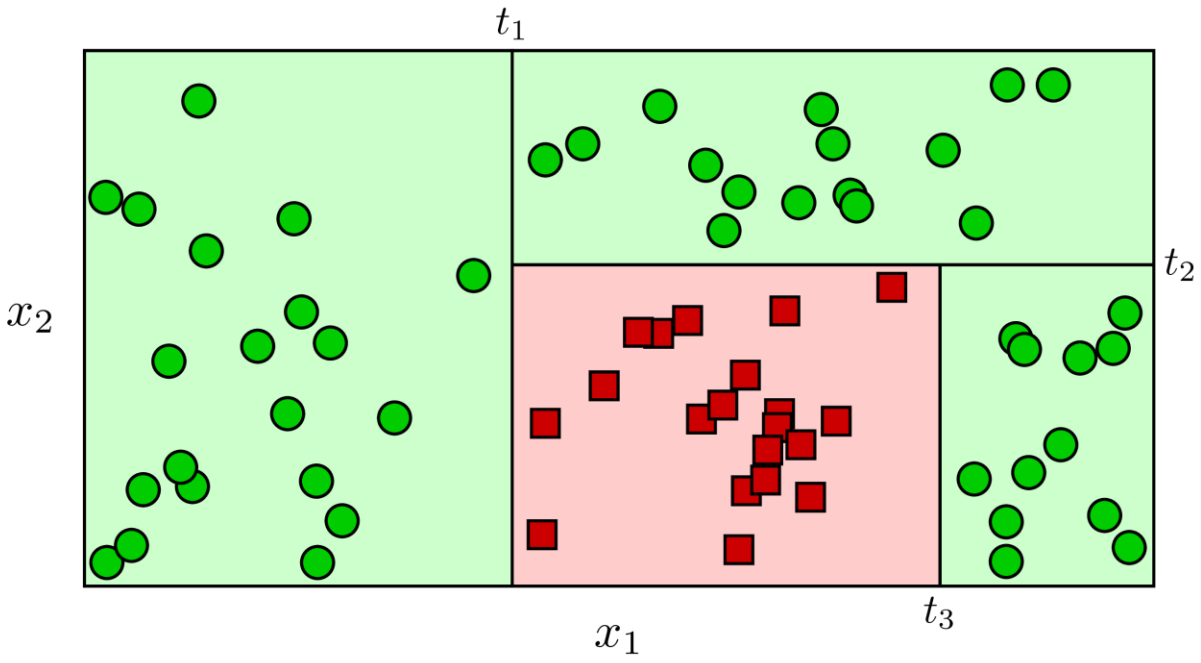


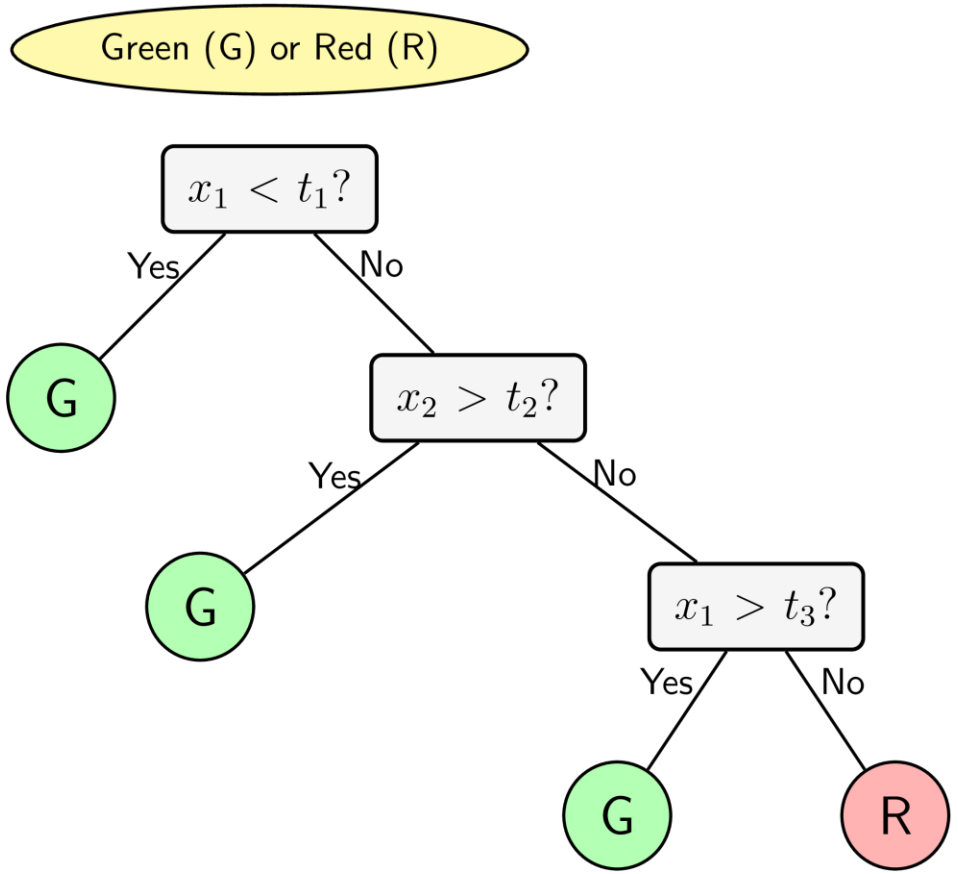
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

E.g., x=(Outlook=sunny, Temperature-Hot, Humidity=Normal,Wind=High), f(x)=Yes.

1. Random Forest Review

Example:



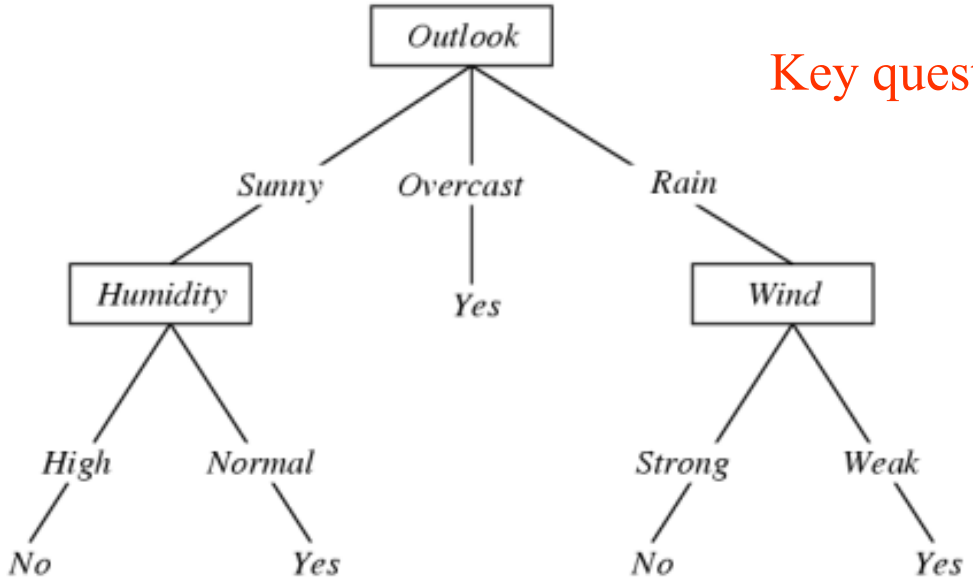


1. Random Forest Review

ID3 approach: Natural greedy approach to growing a decision tree top-down *(from the root to the leaves by repeatedly replacing an existing leaf with an internal node.).*

Algorithm:

- Pick "best" attribute to split at the root based on training data.
- Recurse on children that are impure (e.g, have both Yes and No)



Key question: Which attribute is best?

1. Random Forest Review

ID3 approach: Select attribute with highest information gain (IG)

Information Gain of A is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

$$Gain(S, A) = H_S(Y) - H_S(Y|A)$$

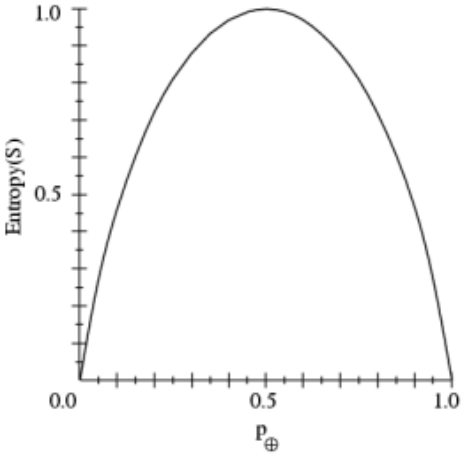
1. Random Forest Review

ID3 approach: Select attribute with highest information gain (IG)

$$Gain(S, A) = H_S(Y) - H_S(Y|A)$$

- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S.
- $p_{\ominus}$ is the proportion of negative examples in S.
- Entropy measures the impurity of S.

$$H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$



- E.g., if all negative, then entropy=0. If all positive, then entropy=0.
- If 50/50 positive and negative then entropy=1.
- If 14 examples with 9 positive and 5 negative, then entropy=.940

1. Random Forest Review

ID3 approach: Select attribute with highest information gain (IG)

$$Gain(S, A) = H_S(Y) - H_S(Y|A)$$

Information Gain of A is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

$$Gain(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

entropy of original collection

Expected entropy after S is partitioned using attribute A

Gain(S,A) information provided about the target function, given the value of some other attribute A.

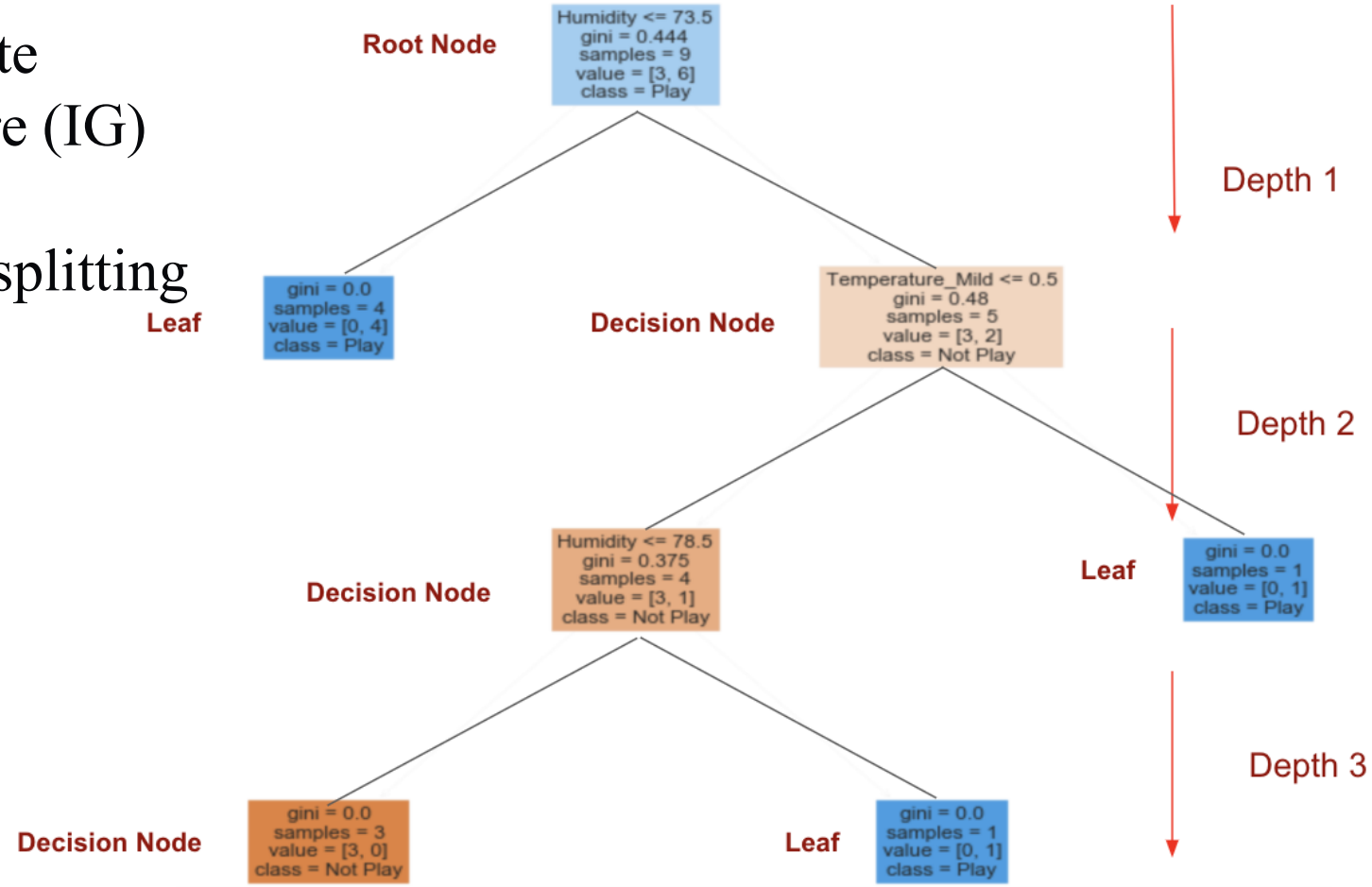
1. Random Forest Review

ID3 steps:

1. Calculate Entropy of one attribute
2. Calculate Entropy of each feature (IG)
3. Choose largest IG as Root Node
4. Entropy = 0 -> Leaf, ≠0 will be splitting
5. Repeat until all data classified

Hyper parameters

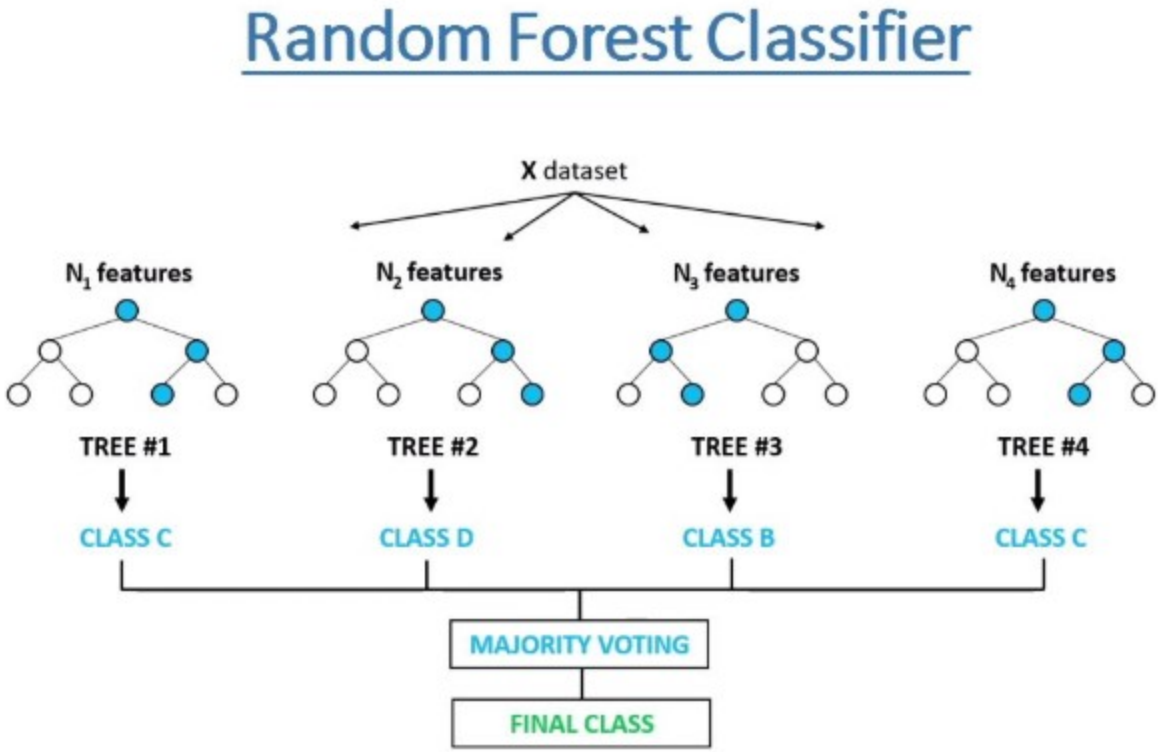
1. Depth
2. Min leaf size
3. Min no size to split
4. Max number of leaves
5. Min impurity decrease



1. Random Forest Review

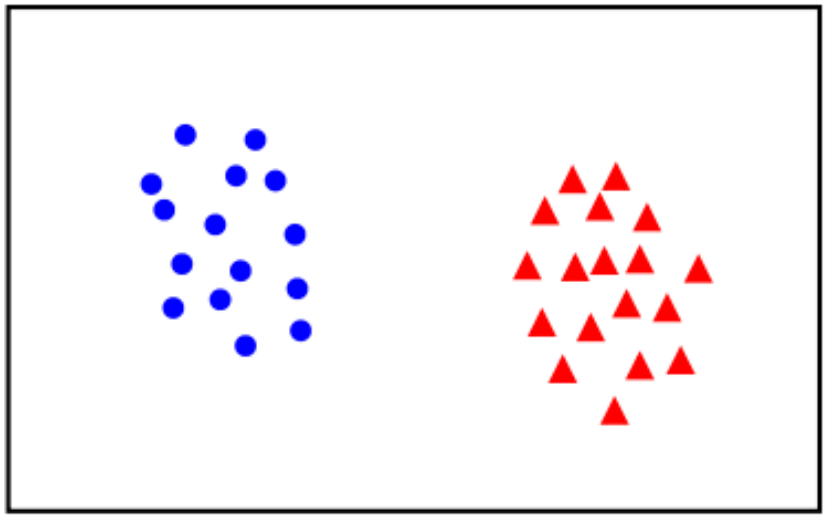
Random Forest Steps:

1. Select random samples from a given dataset.
2. Construct a decision tree for each sample and get a prediction result from each decision tree.
3. Perform a vote for each predicted result.
4. Select the prediction result with the most votes as the final prediction.



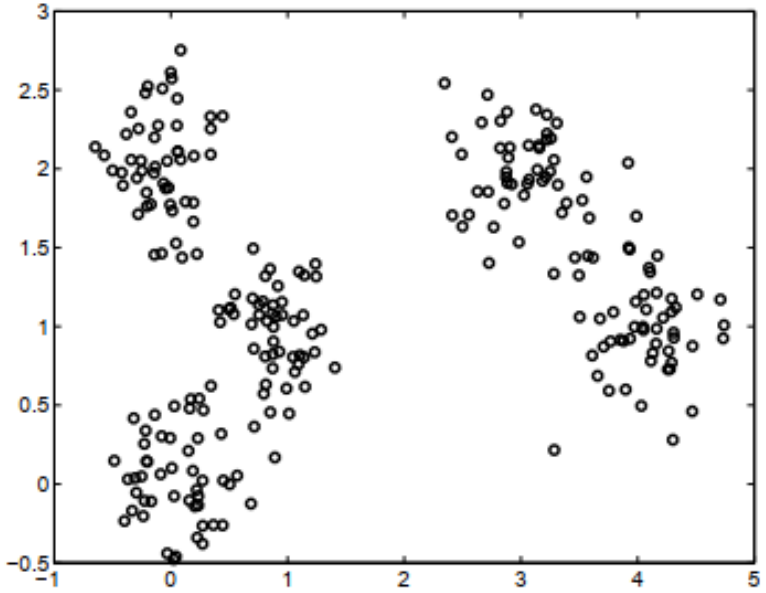
2. Clustering Intuition

Supervised learning



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Unsupervised learning



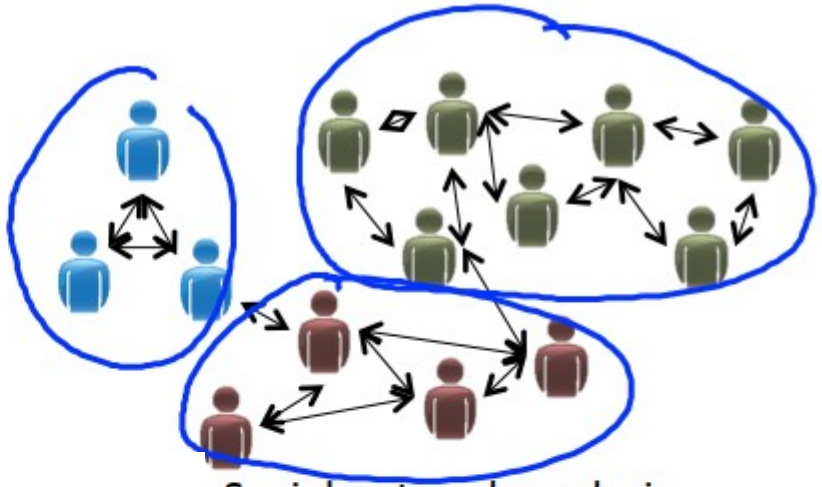
Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

No label data (y)

2. Clustering Intuition

Applications of clustering

Market segmentation



Social network analysis

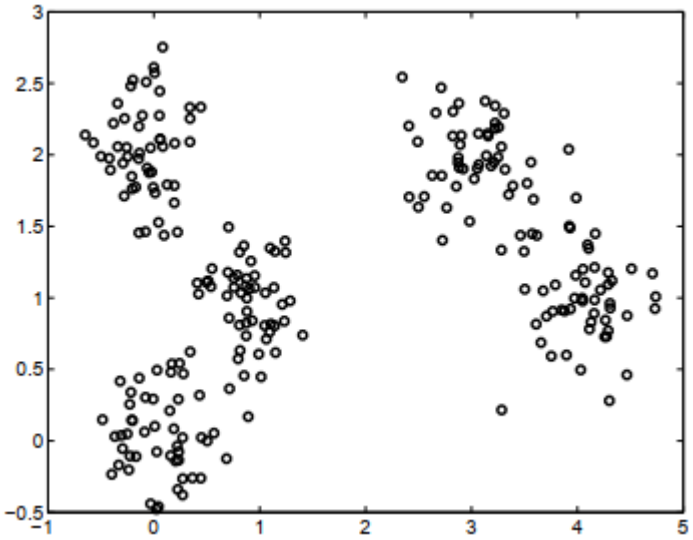
Organize computing clusters

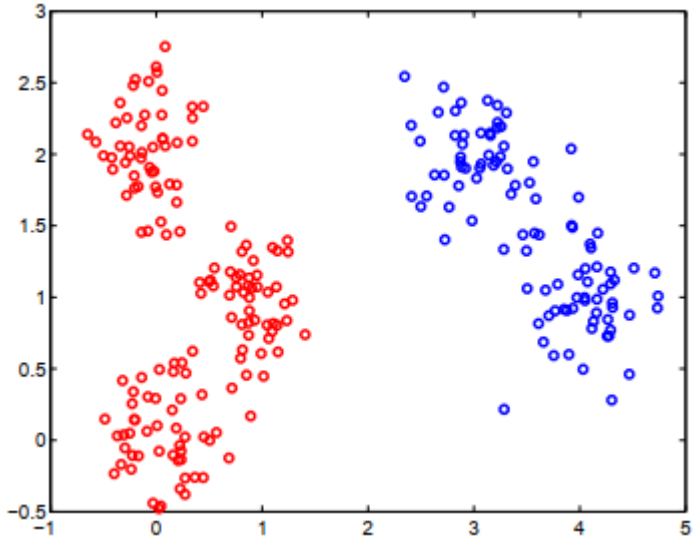
Astronomical data analysis

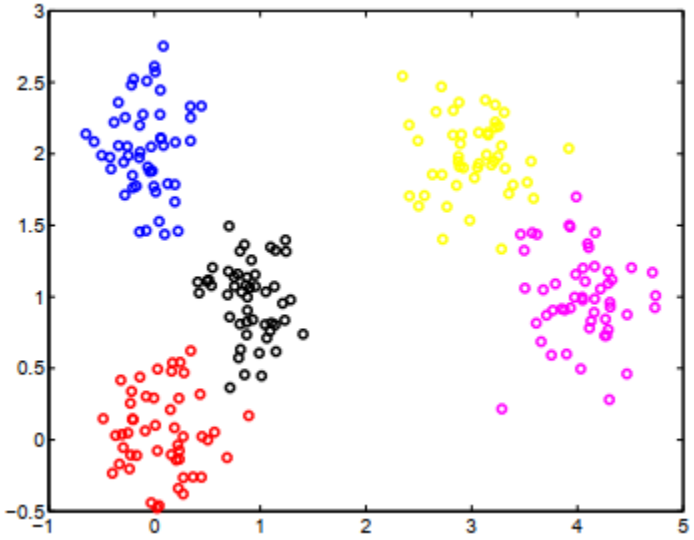
2. Clustering Intuition

Clustering: Finding structure in the data

- By isolating groups of examples that are similar in some well-defined sense
- Unsupervised learning algorithm: **only input data, no label information**







How many classes is the best?

Depend on the measure of similarity (or distance) between the data points to be clustered

3. K-means algorithm

Clustering methods:

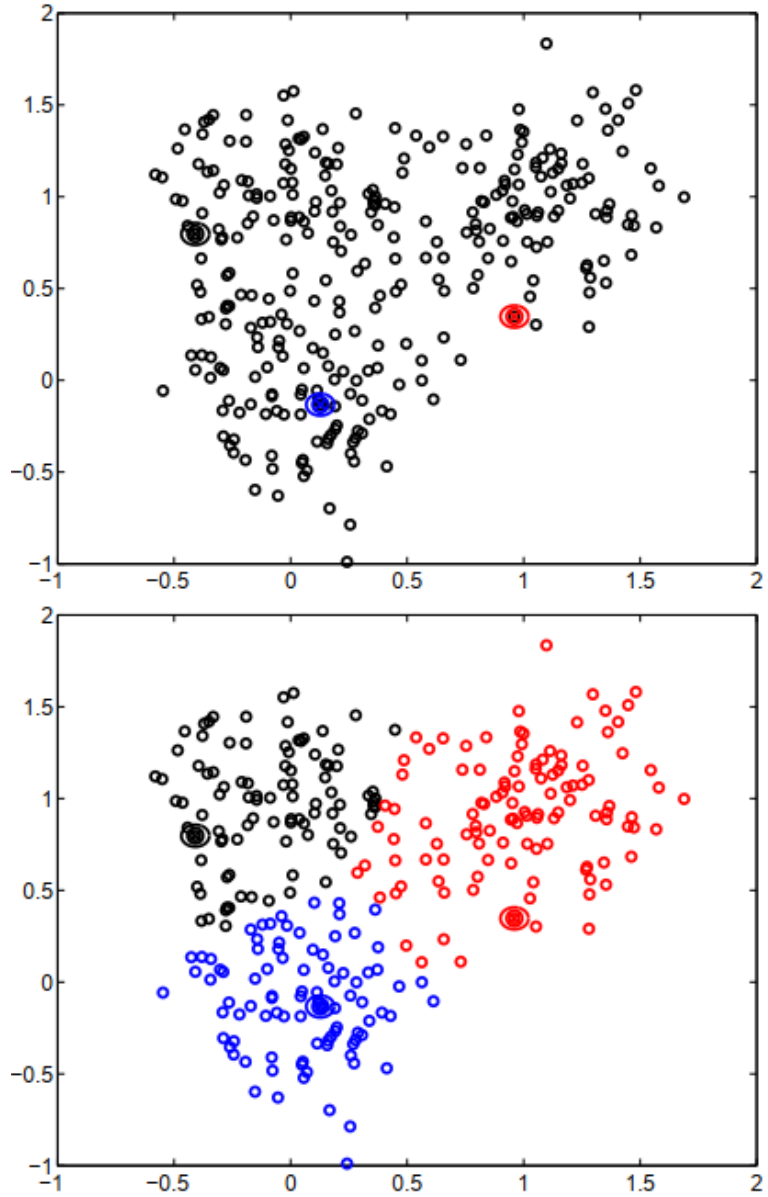
- Hierarchical clustering methods
- Spectral clustering
- Semi-supervised clustering
- Clustering by dynamics
- Flat clustering methods: k-means clustering
- Etc.

3. K-means algorithm

Procedure:

1. Pick *k* arbitrary centroids (cluster means)
2. Assign each sample to its closest centroid
3. Adjust the centroids to be the means of the examples assigned to them
4. Repeat step 2 until no change

K-means algorithm is guaranteed to converge in a finite number of iterations



3. K-means algorithm

training set $\{x^{(1)}, \ldots, x^{(m)}\}$

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set

   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

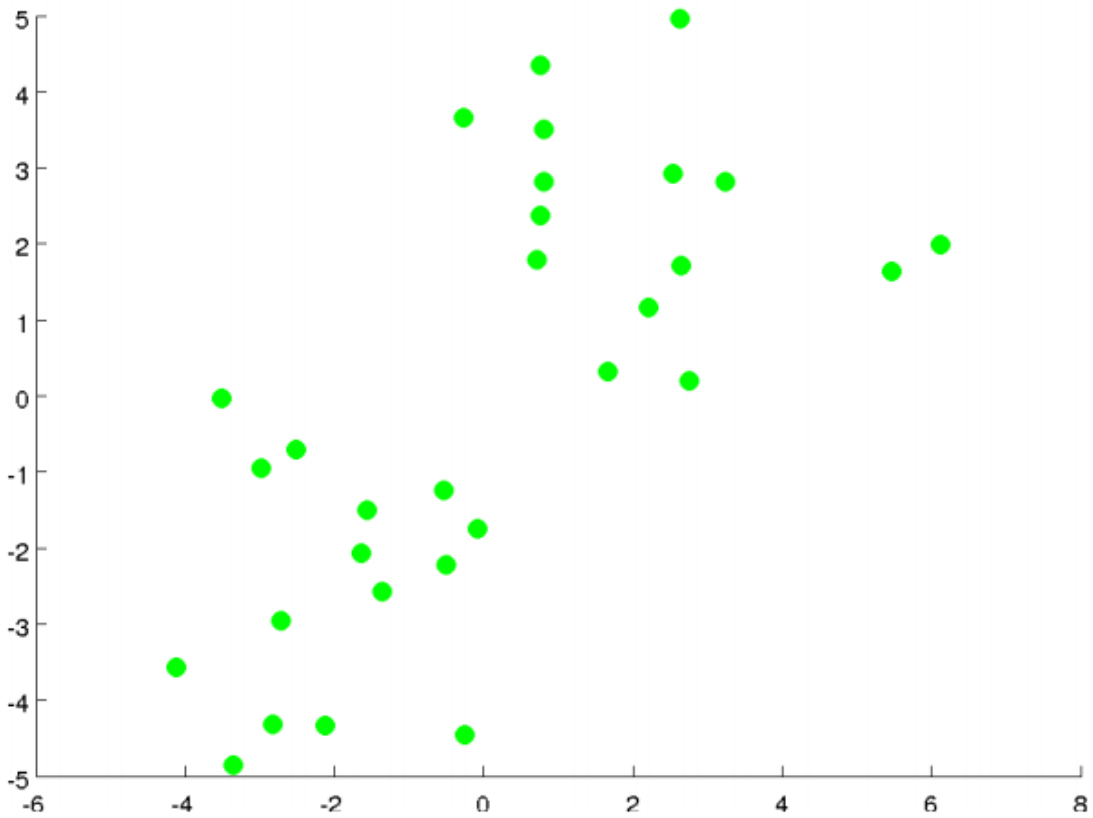
   For each j, set

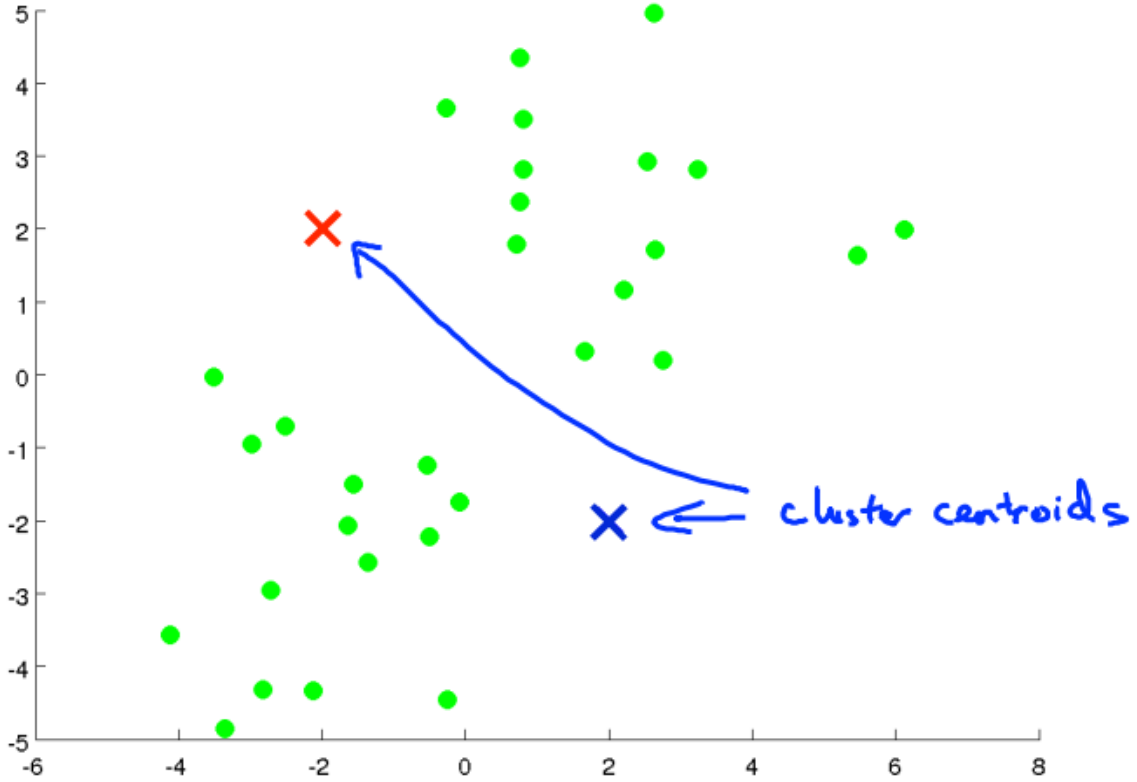
   $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

   }

3. K-means algorithm

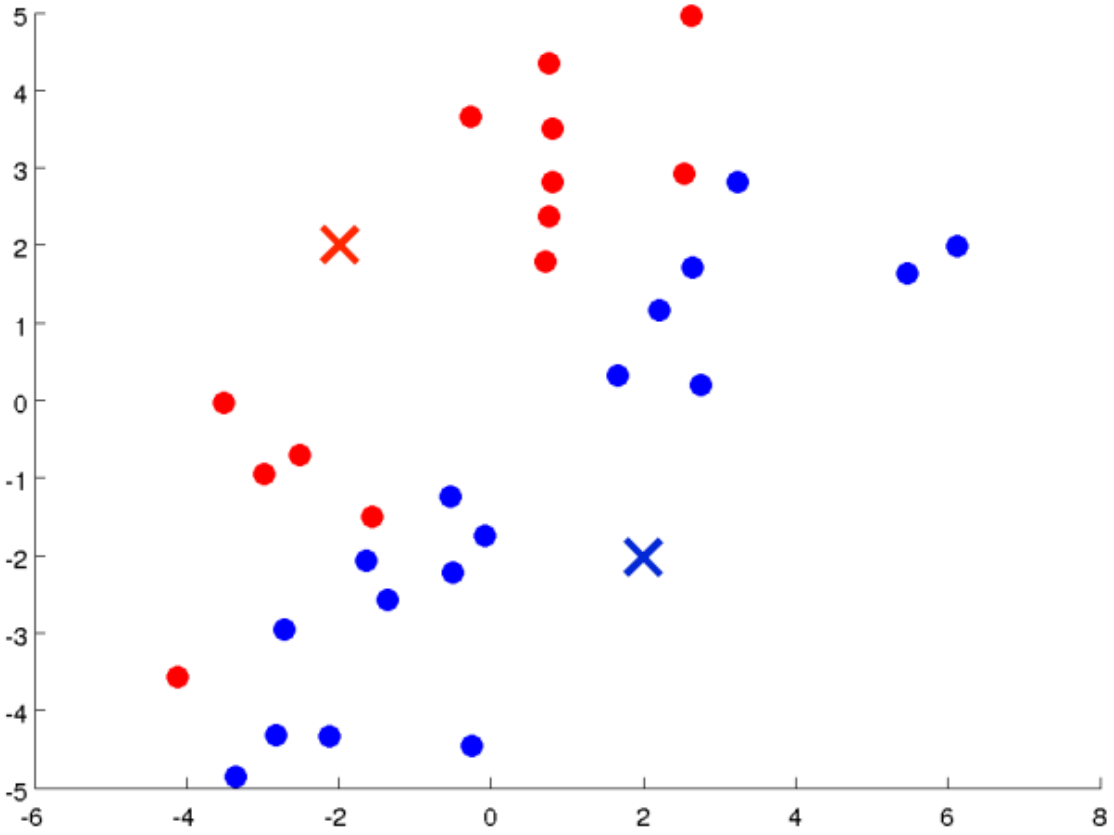
Procedure illustration:

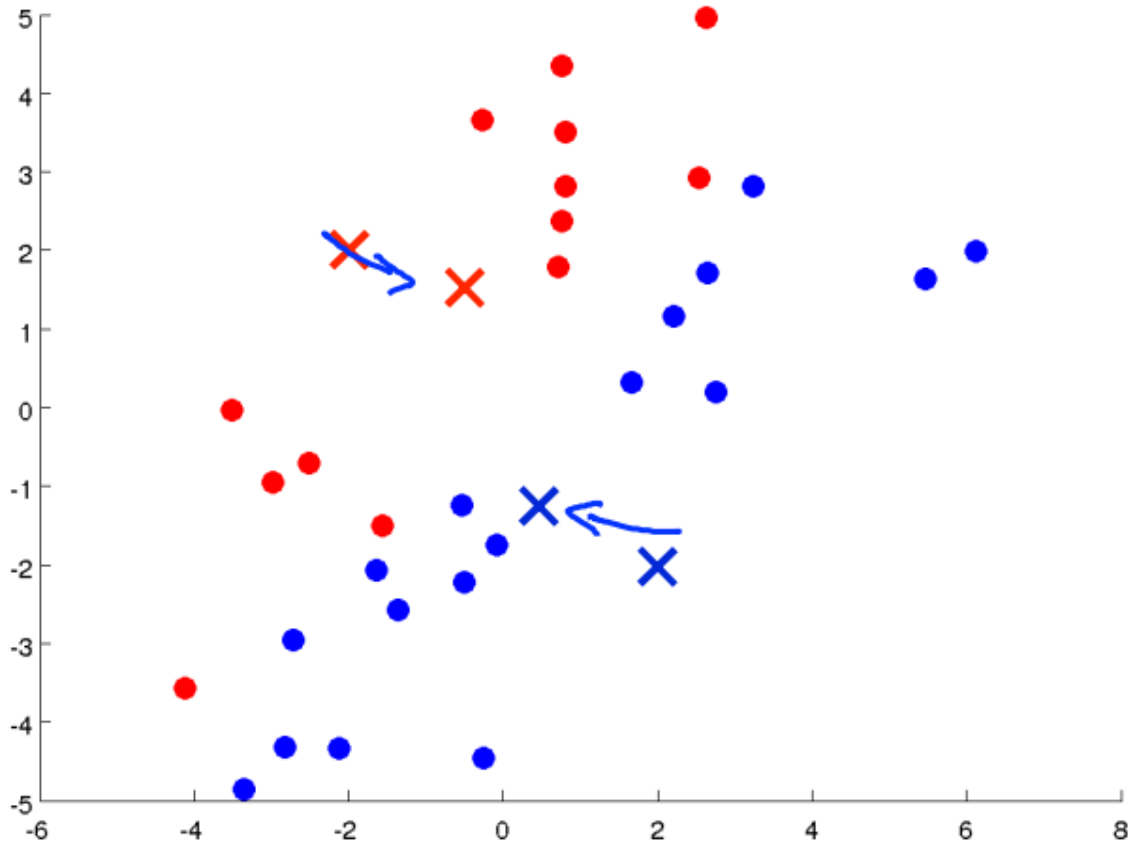




3. K-means algorithm

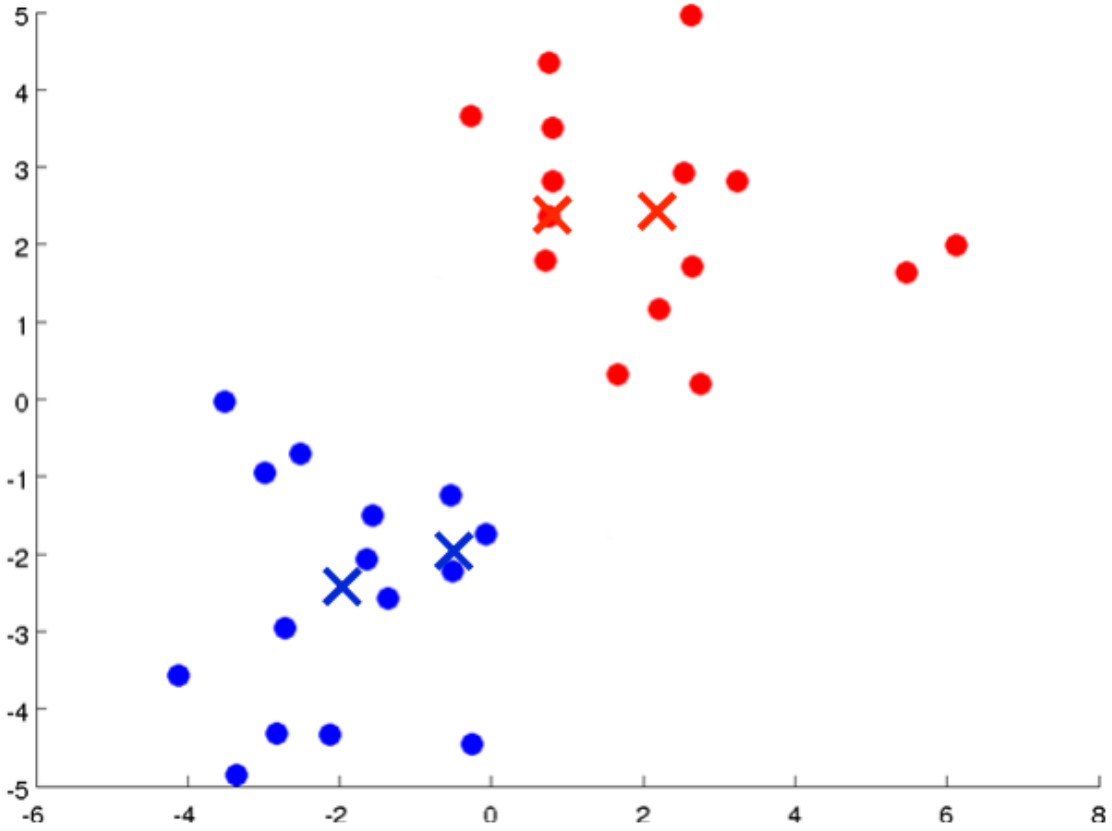
Procedure illustration:

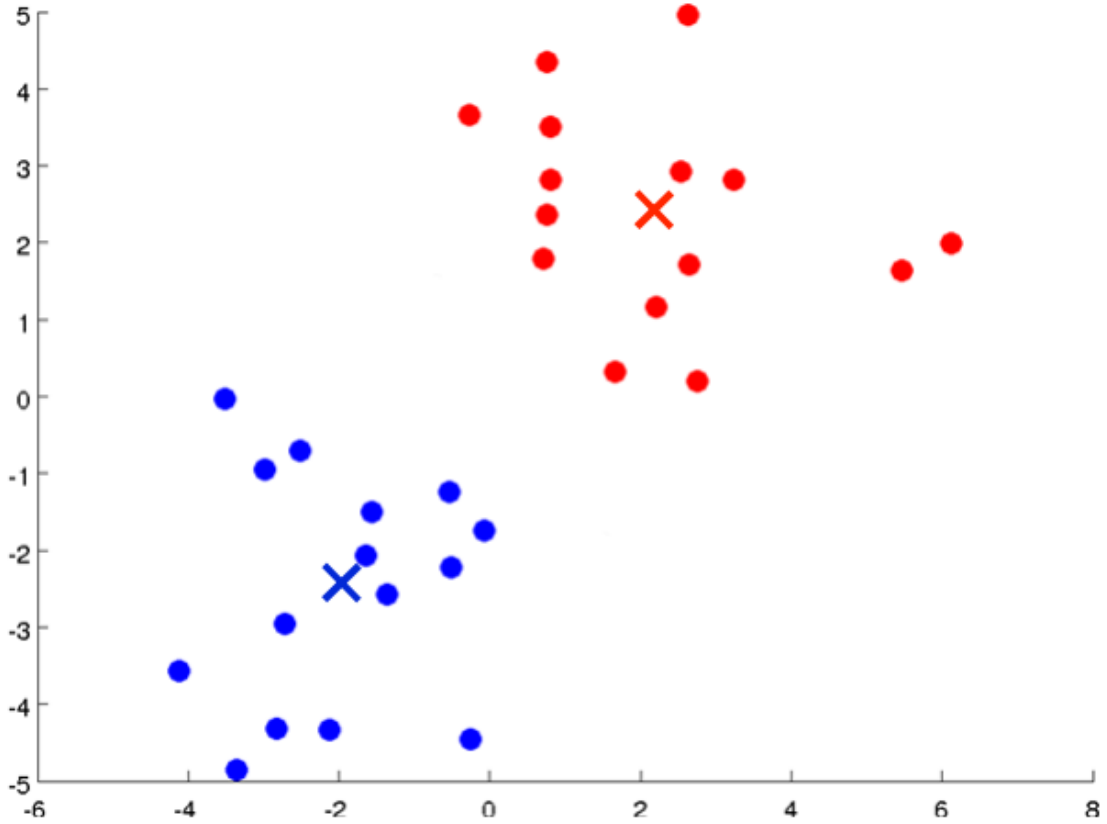




3. K-means algorithm

Procedure illustration:





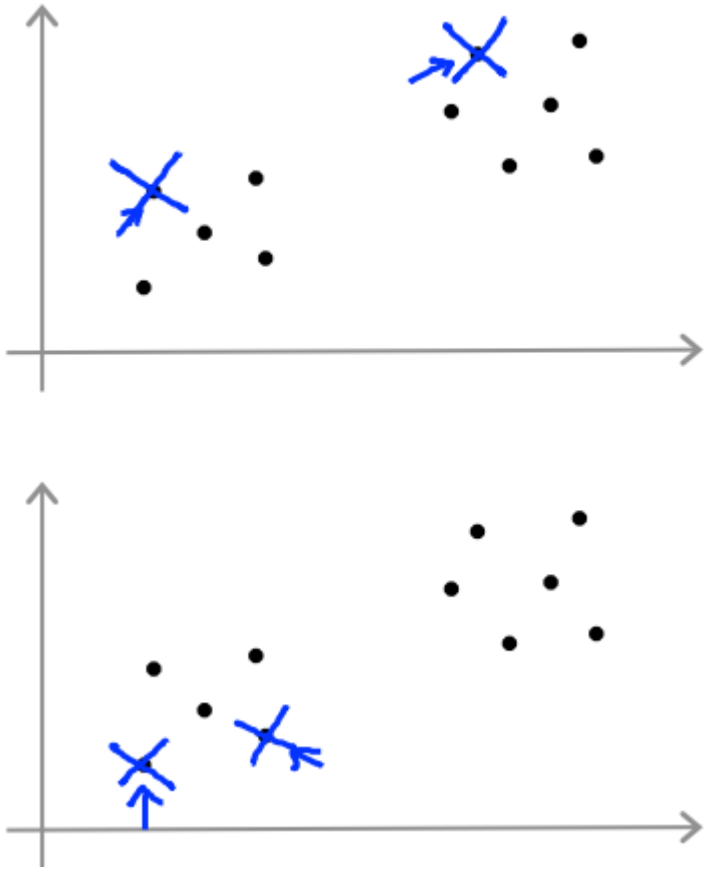
3. K-means algorithm

Random initialization

Should have $K < m$

Randomly pick K training examples.

Set $\mu_1, \ldots, \mu_K$ equal to these K examples.



3. K-means algorithm

How to choose *k*?

For i = 1 to 100 { Could repeat several times to get the best solution

Randomly initialize K-means.

Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$.

Compute cost function (distortion)

$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

}

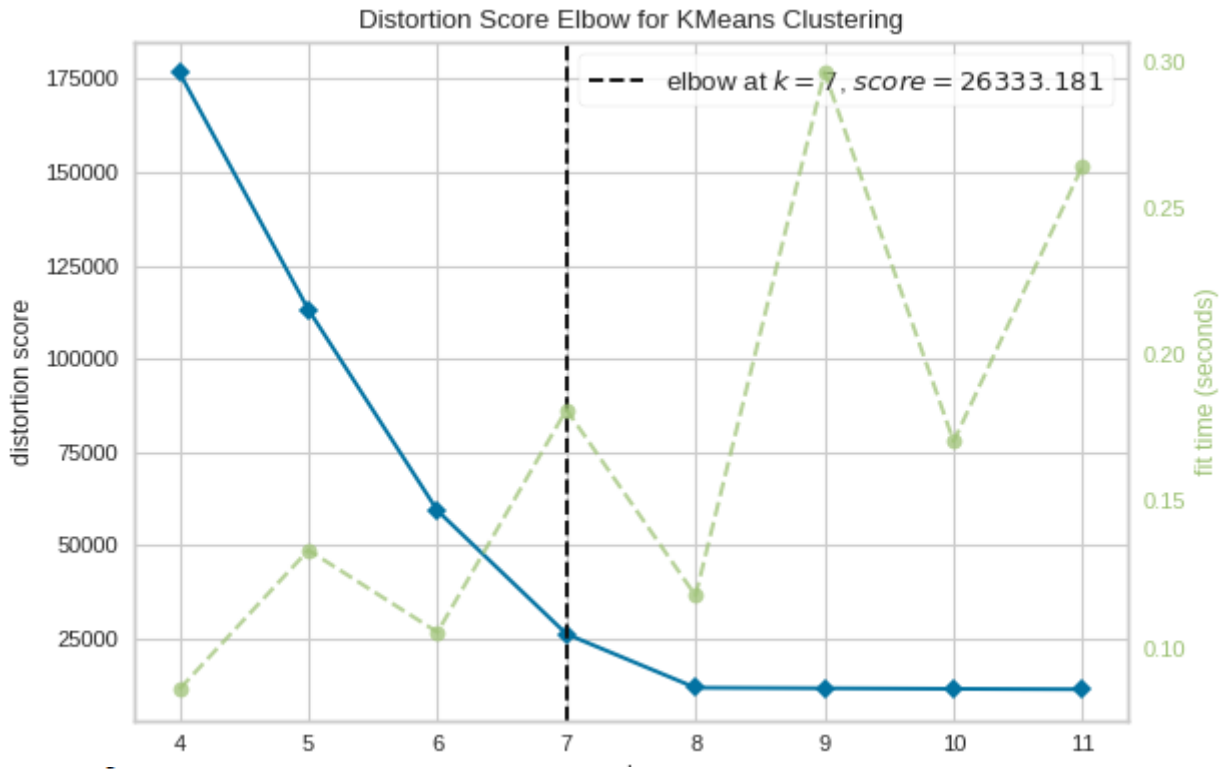
Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

3. K-means algorithm

How to choose *k*? "Elbow" method

- As *k* is large (smaller clusters), *J* is smaller however the model is easy overfitting
- *k* should be chosen at the "elbow" where *J* is not significantly reduced



J: distortion score

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

5. Summary

- K-means is a parametric method, where the parameters are the prototypes.
- Inflexible; the decision boundary is linear.
- Fast! The update steps can be parallelized.
- There are several variations on the basic K-means algorithm
  1. K-means++ gives a more specific way to initialize clusters
  2. K-medoids chooses the centermost datapoint in the cluster as the prototype instead of the centroid. (The centroid may not correspond to a datapoint.)