EE3-23: Machine Learning

Deniz Gündüz and Krystian Mikolajczyk

Department of Electrical and Electronic Engineering
Imperial College London

Fall 2019

Today

- Nearest neighbour classification
- Unsupervised learning (Clustering)

Nearest Neighbour Classification

- One of the earliest and simplest learning algorithms
- Memorize the training set
- Predict label of any new instance based on the labels of its "closest" neighbours in the training set (instance-based)
- Assumption: Labels of nearby points in the feature space are the same
- Does not identify a predictor function from a class of specified functions based on the training dataset (non-parametric)

Nearest Neighbour Classification

Labelled training set: $(x_1, y_1), \ldots, (x_m, y_m), \quad x_i \in \mathbb{R}^d, \quad y_i \in \mathcal{C}$

Nearest neighbour: For new instance x, define $nn(x) \in [m] \triangleq \{1, \ldots, m\}$:

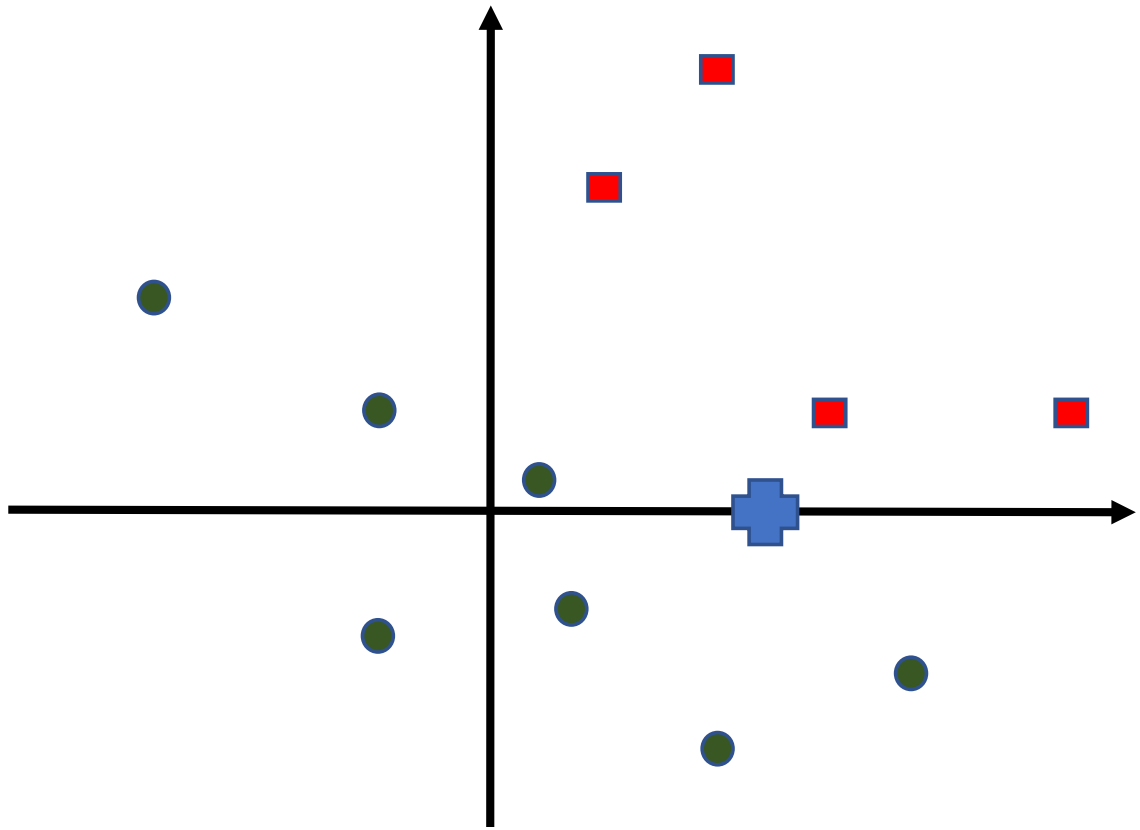
$$nn(x) = \operatorname*{argmin}_{i \in [m]} \|x - x_i\|^2$$

returns the index of the training example nearest to x

Classification rule:

$$y = f(x) = y_{nn(x)}$$

Example



Will be classified as red!

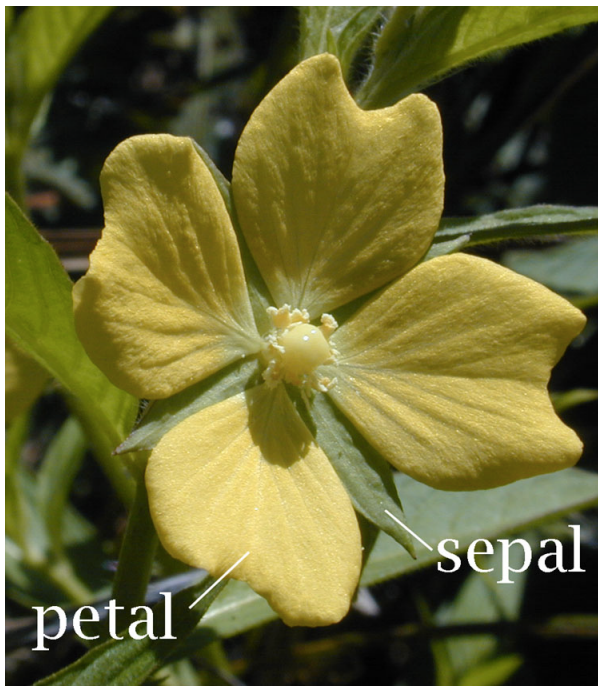
Minimal training, but expensive testing

Famous example: Iris classification

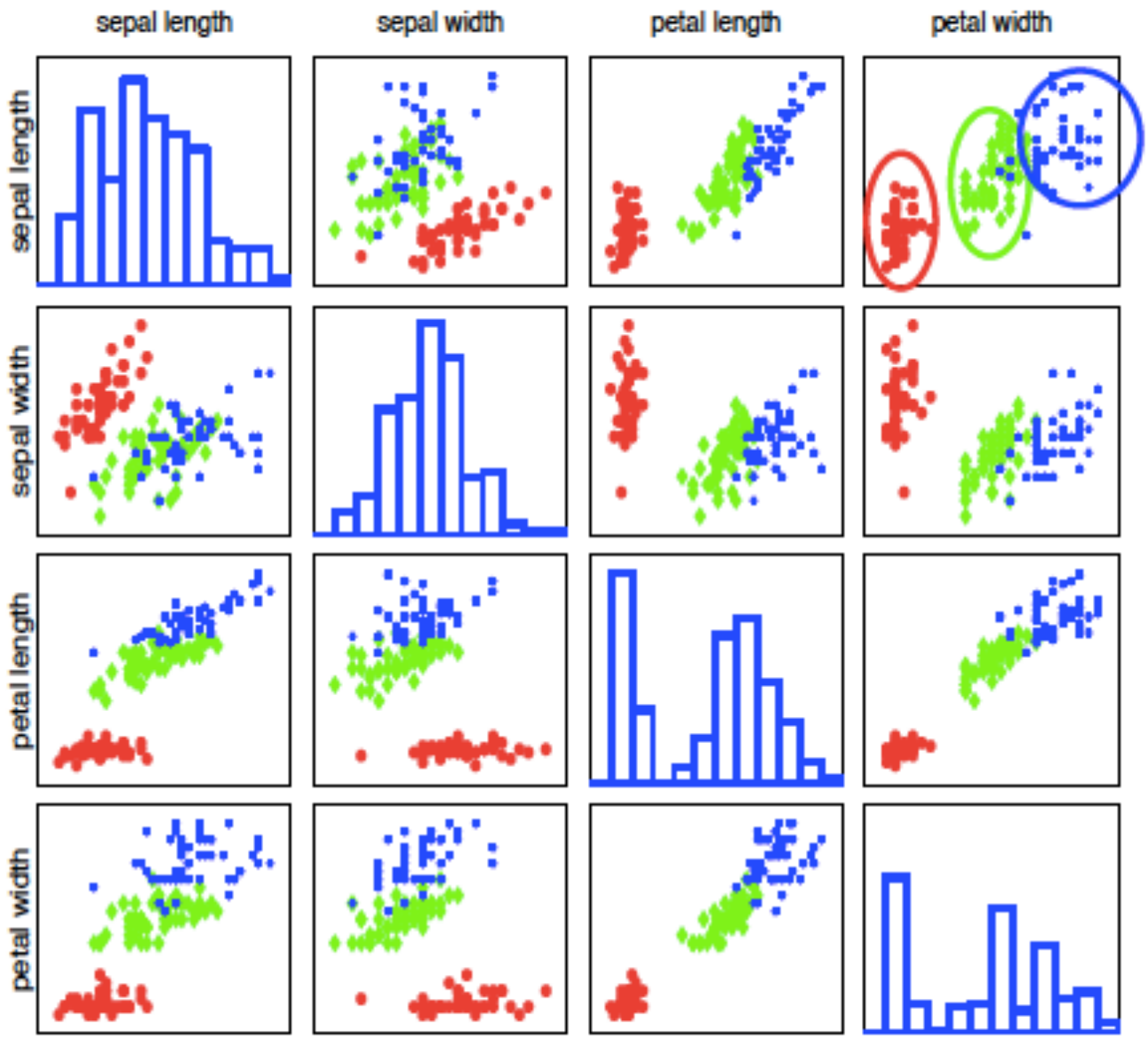


- Three types:

  setosa, versicolor, and virginica
- Dataset introduced originally by statistician Ronald Fisher in 1936
- 50 observations from each three species
- Four features measured: length and width of sepals and petals



Iris classification



Example: Iris classification with two features

Training data

ID (i)	petal width (x_i)	sepal length (x_2)	type
1	0.2	5.1	setosa
2	1.4	7.0	versicolor
3	2.5	6.7	virginica

New flower:

petal width $= 1.8$, sepal length $= 6.4$

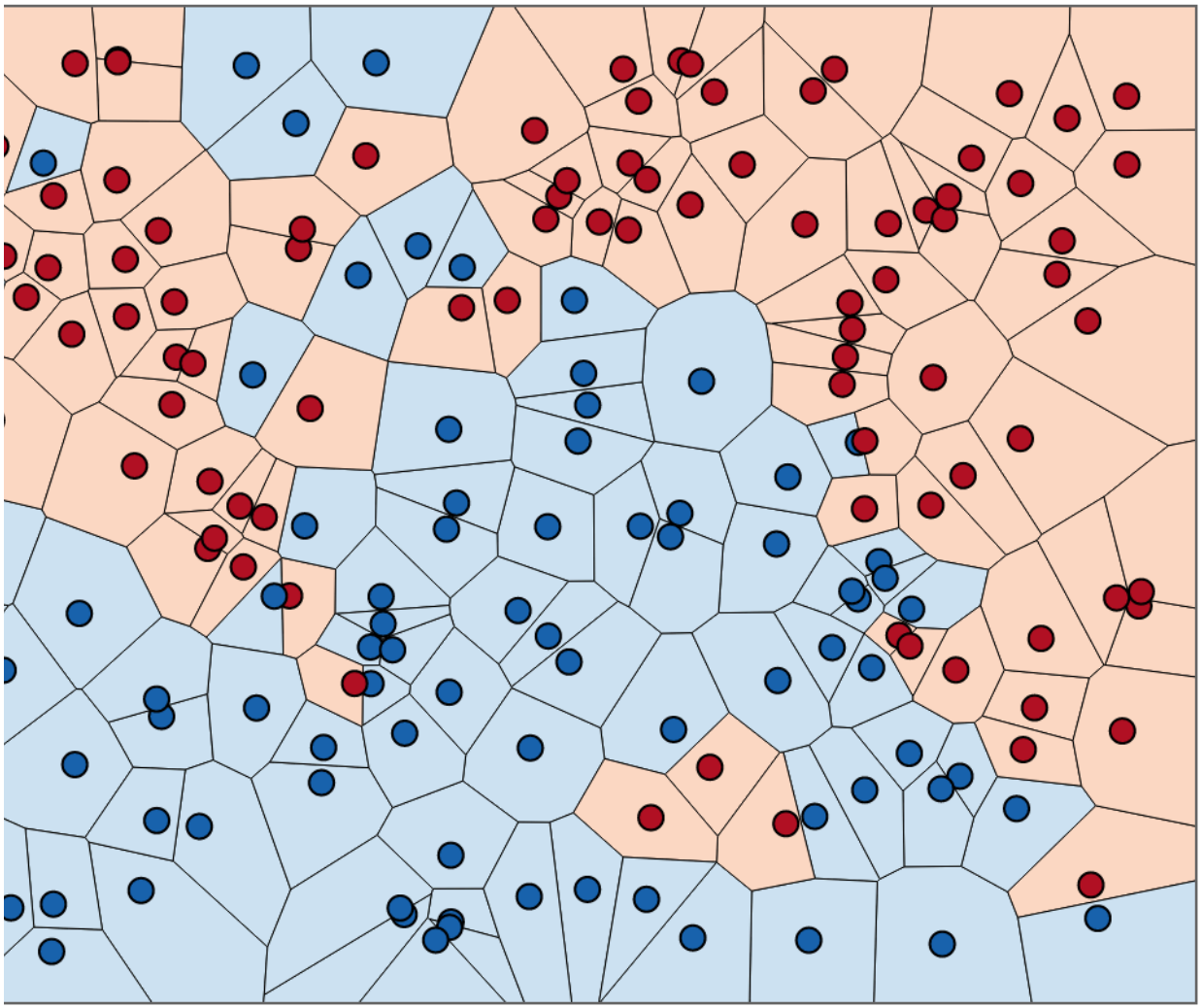
ID (i)	Distance
1	1.75
2	0.72
3	0.76

Predicted category: versicolor

Decision boundary

We can determine the label of any point in the space.

This leads to decision boundaries



K-nearest neighbour (KNN) classification

Consider K nearest neighbours:

- Nearest neighbour: $nn_1(x) = \text{argmin}_{i \in [m]} \|x - x_i\|^2$
- 2nd nearest neighbour: $nn_2(x) = \text{argmin}_{i \in [m] \setminus \{nn_1(x)\}} \|x - x_i\|^2$
- 3rd nearest neighbour: $nn_3(x) = \text{argmin}_{i \in [m] \setminus \{nn_1(x), nn_2(x)\}} \|x - x_i\|^2$

Set of K-nearest neighbours:

$$knn(x) = \{nn_1(x), \ldots, nn_K(x)\}$$

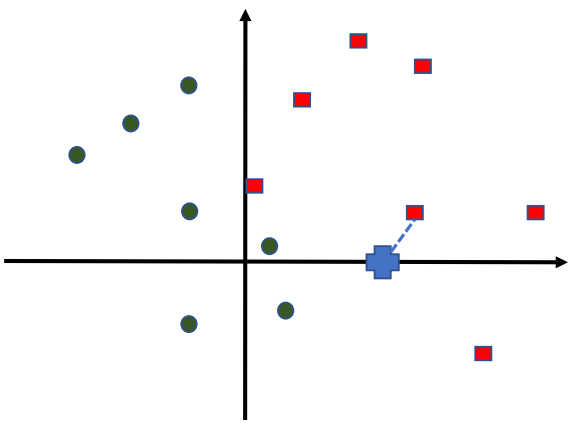
Classification rule:Majority vote among KNNs:

$$v_c = \sum_{i \in knn(x)} \mathbb{I}(y_i = c), \quad \forall c \in \mathcal{C}$$

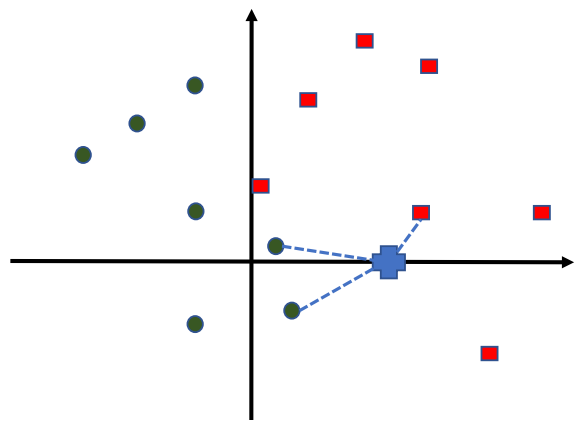
$$y = f(x) = \underset{c \in \mathcal{C}}{\text{argmax}}\, v_c$$

Example

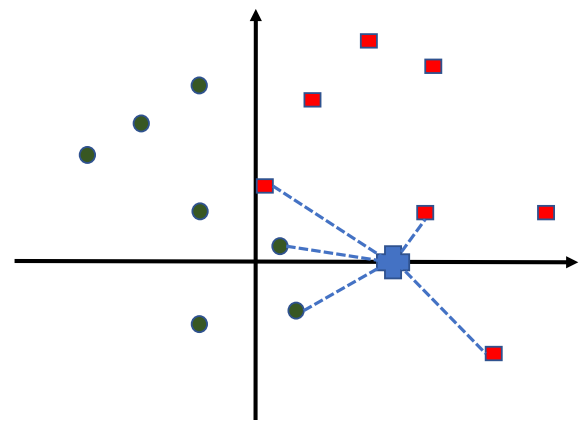
$K = 1$, Label: Red



$K = 3$, Label: Green

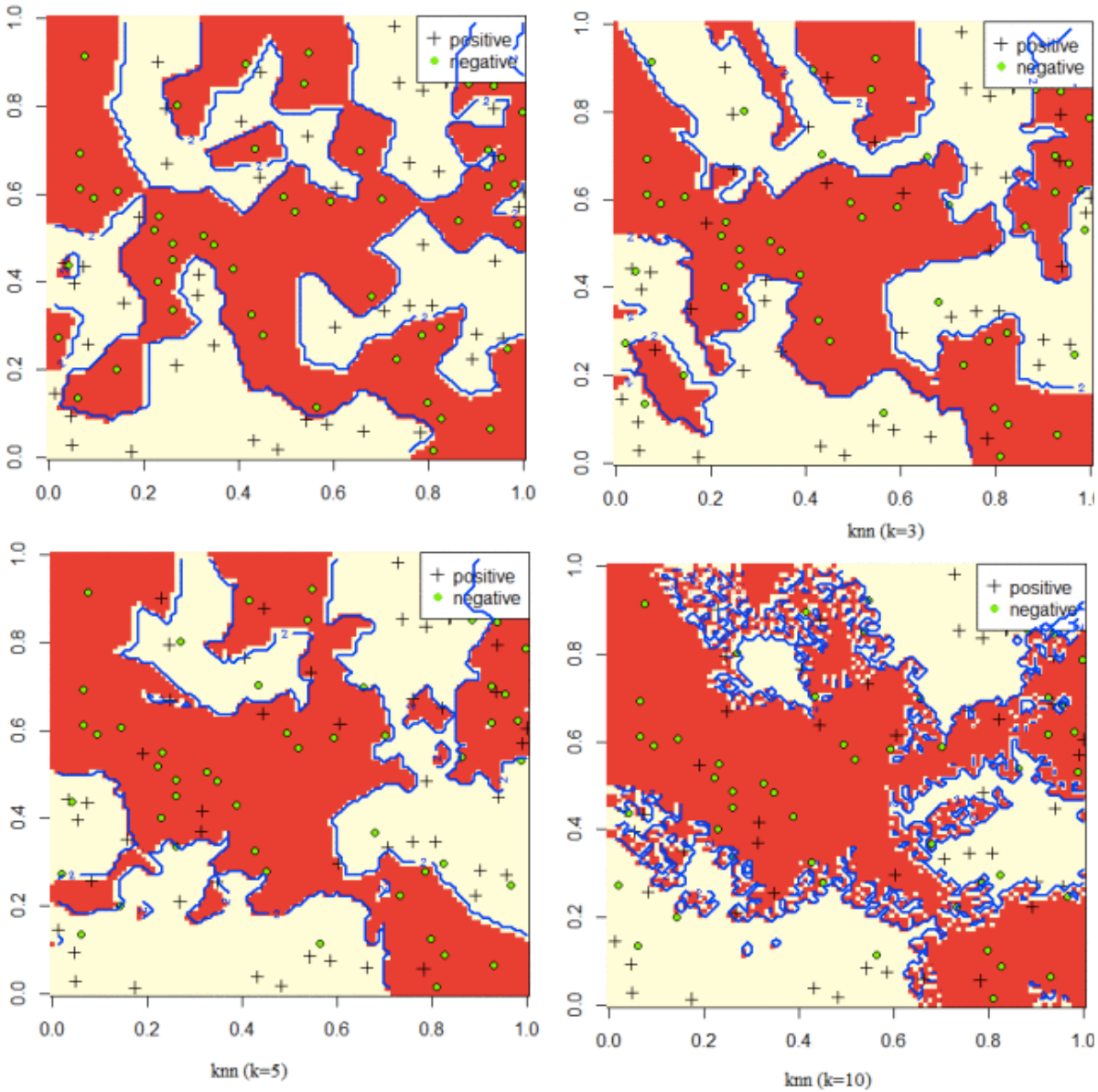


$K = 5$, Label: Red



Increasing K

As K increases, decision boundaries become smoother



Distance measure

Different distance measures can be used, i.e.,

$$\|x - x'\|_p = \left(\sum_{j=1}^{d} |x_j - x'_j|^p \right)^{1/p}$$

Pros and cons of KNN classification

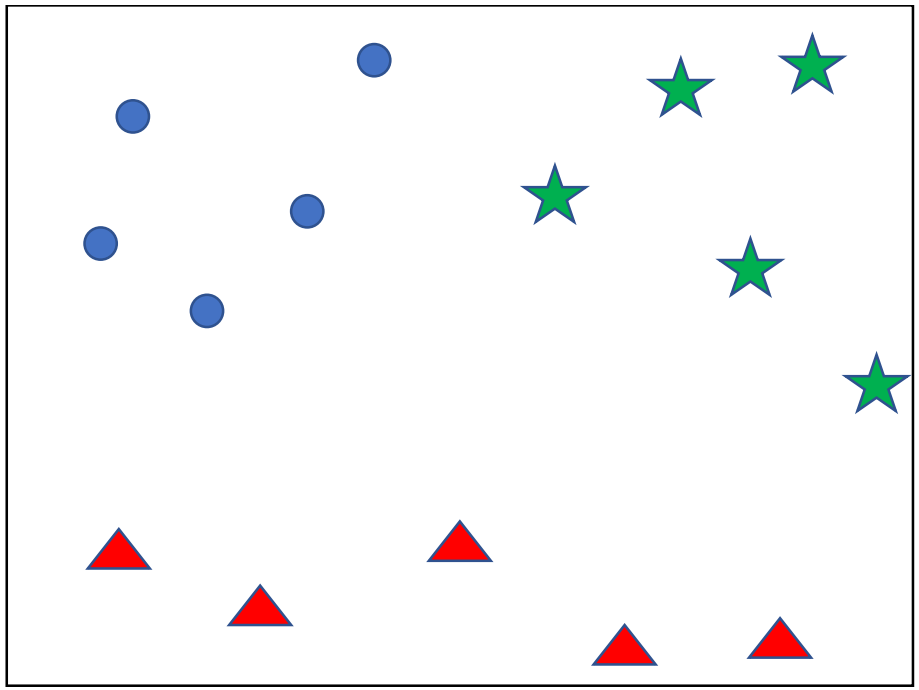
- Advantages
  - Easy to implement: just compute distances
  - Has strong theoretical guarantees
- Disadvantages
  - Computationally intensive for large-scale problems: $O(m \cdot d)$ for labelling a data point (can be reduced by approximate search methods)
  - We need to keep all the training data (non-parametric)
  - Choosing the right distance measure and K can be involved.

Machine Learning

- Supervised learning: Given data samples with labels (X, y), we want to learn a function $y = f(X)$ to predict labels of new samples
  - Classification: y is discrete
  - Regression: y is continuous
- Unsupervised learning: We are given only samples of data X, we want to compute a function $y = f(X)$ that provides a simpler representation
  - y is discrete: Clustering
  - y is continuous: Matrix factorization, autoencoders, Kalman filtering

Clustering

- Goal is to group 'similar' items into clusters



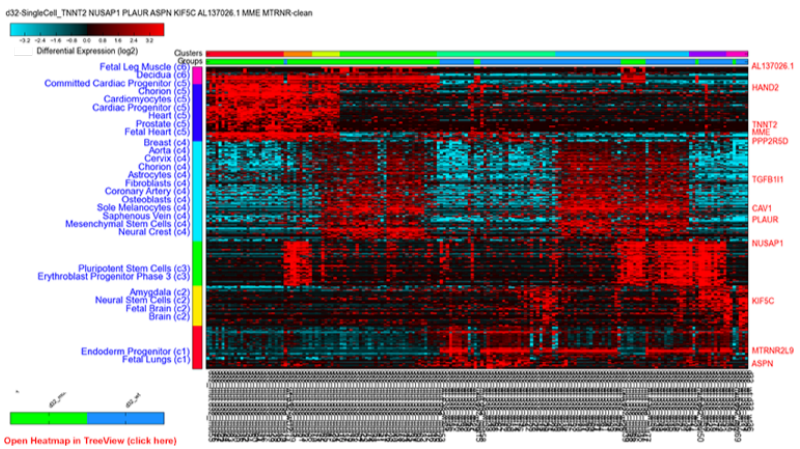
- We want
  - the items in the same cluster to be similar
  - items in different cluster to be dissimilar

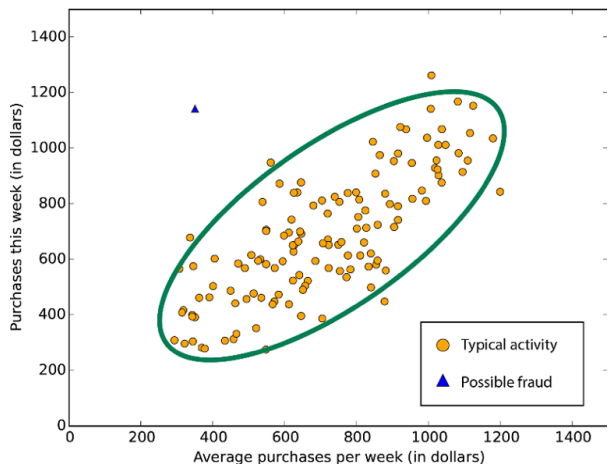
Early Application

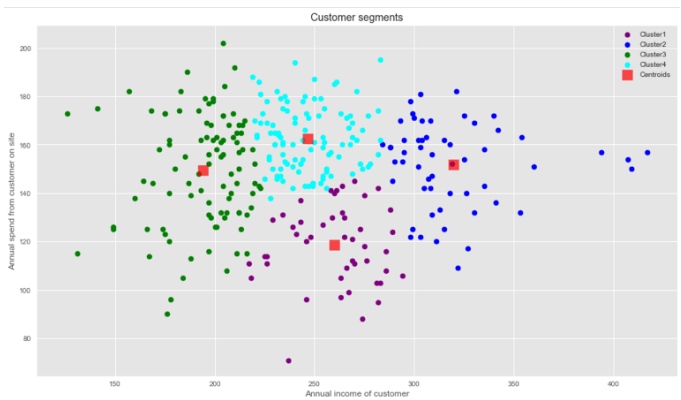
- John Snow (1813-1858), a London physician, created a map showing the deaths caused by a cholera outbreak in Soho and the locations of water pumps in the area.
- Observed that deaths were clustered around certain pumps
- Removing the handles stopped the deaths

Applications of Clustering

- Anomaly detection (identifying fake news, spam detection, etc.)
- Marketing (cluster customers, products, etc.)
- Gene clustering



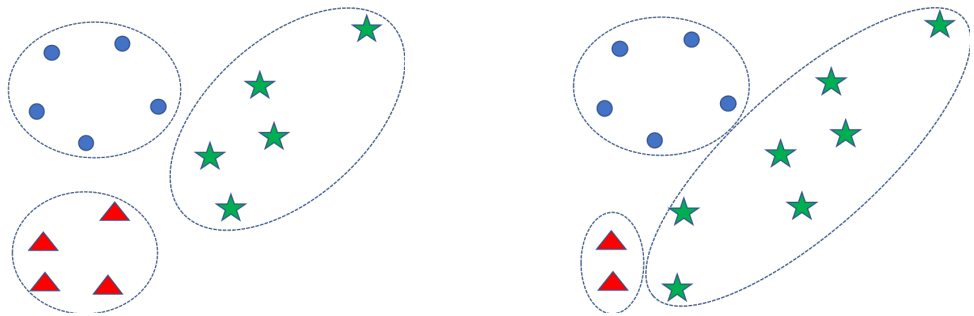




How to measure 'similarity' ?

- We want a function that assigns a real number to every pair of two samples from the space,
- Function value should increase with dissimilarity of the objects
- For example:
  - Eucledian distance: $d(x, y) = \sqrt{\sum_{i=1}(x_i - y_i)^2}$
  - Correlation coefficient: $d(x, y) = \frac{\sum_{i=1}(x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$

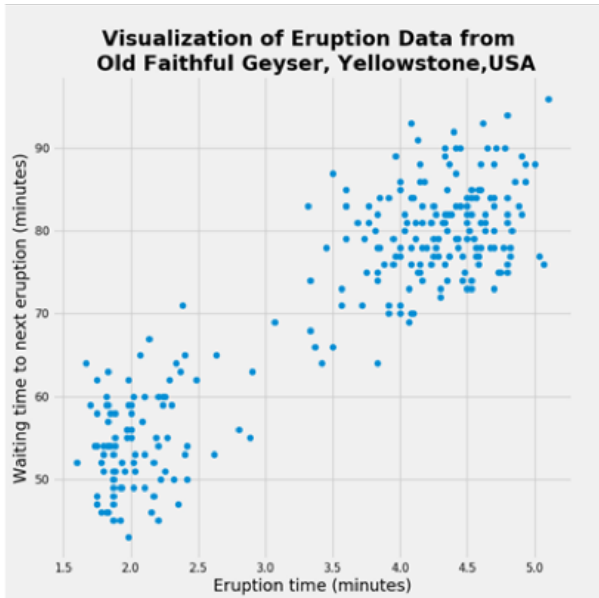
How to evaluate clusters?



(a) good cluster (b) bad cluster

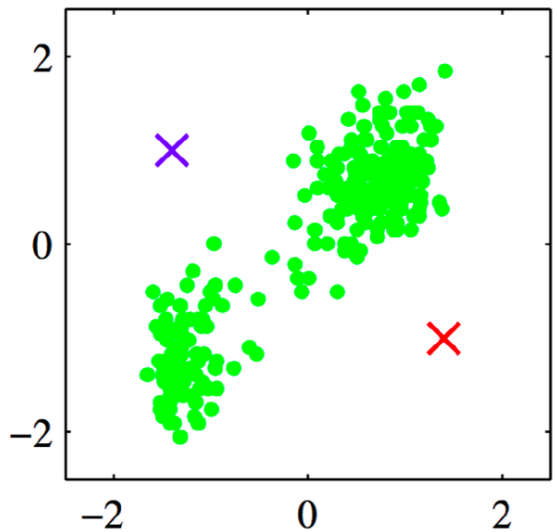
- Intra-cluster cohesion (compactness)
  - How close the samples in a cluster to the cluster center
- Inter-cluster separation (isolation):
  - How far (dissimilar) different cluster centroids from one another.
- In most case we depend on expert judgement

k-Means Clustering



k-Means Clustering

- Standardise data
- Choose two cluster centers

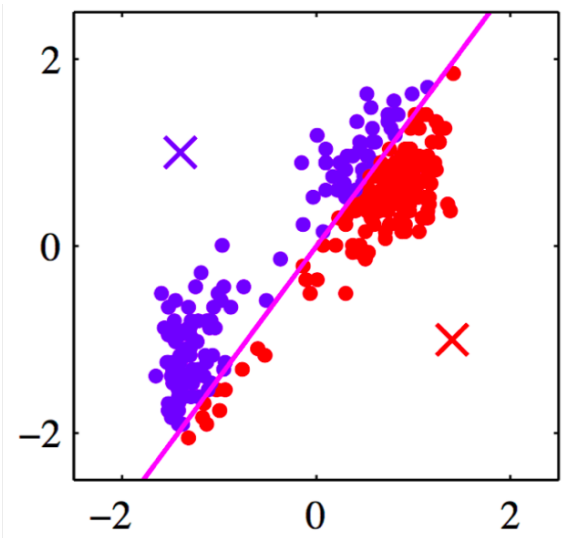


1

[1] C. Bishop, Pattern Recognition and Machine Learning, Figure 9.1.

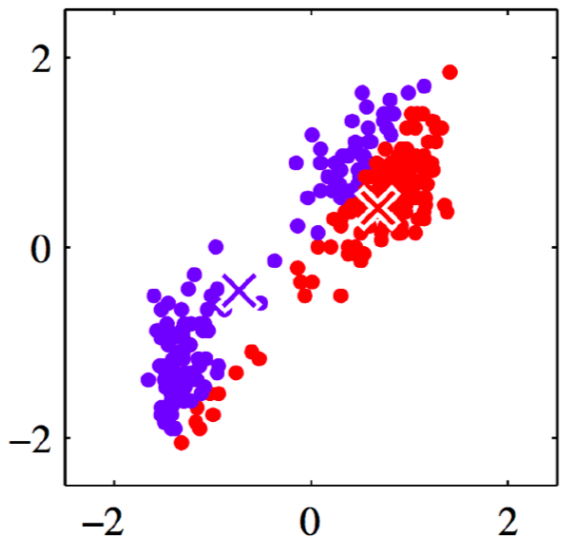
k-Means Clustering

- Assign each data point to one of the two clusters
- Equivalent to classification of data samples with the perpendicular bisector of the two cluster centers



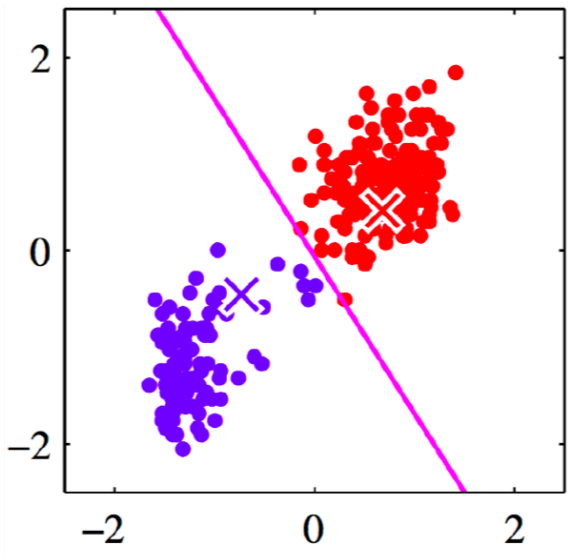
k-Means Clustering

- Recompute cluster centers



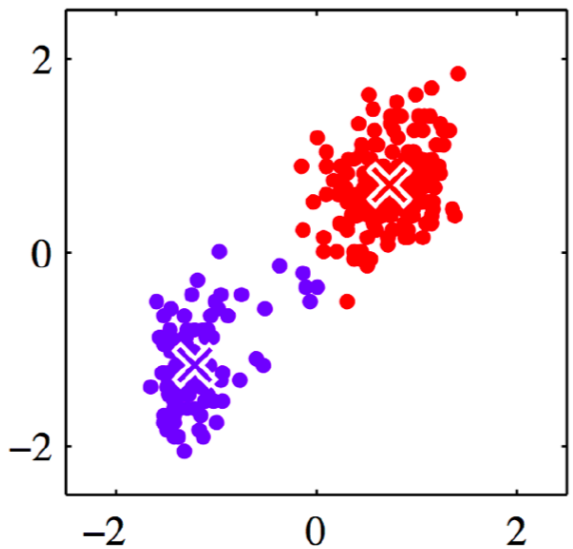
k-Means Clustering

- Assign points to the closest center



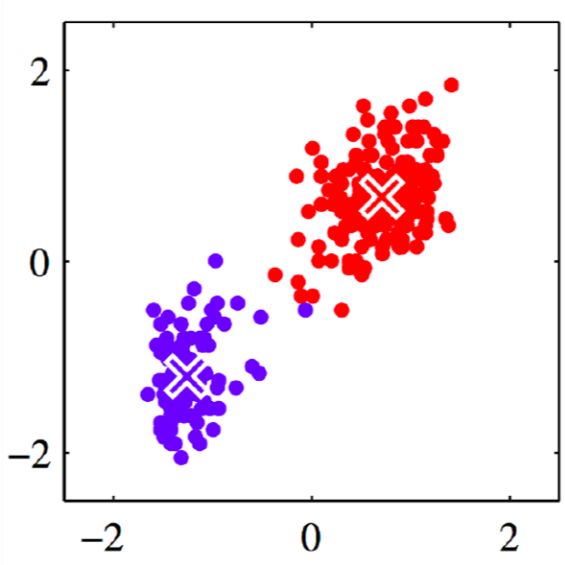
k-Means Clustering

- Compute cluster centers



k-Means Clustering

- Follow the same steps until convergence



k-Means Algorithm: Formal Description

- Let $\mathcal{X}$ be a space with some distance metric d.
  (e.g., $\mathcal{X} = \mathcal{R}^d$ and $d(x, y) = \|x - y\|$
- Dataset: $\mathcal{D} = \{x_1, \ldots, x_n\},\ x_i \in \mathcal{X}$
- We want to create k clusters $C_1, \ldots, C_k$
- The cluster center of C_i is given by

$$\mu_i = \mu(C_i) = \operatorname*{argmin}_{\mu \in \mathcal{X}} \sum_{x \in C_i} d(x, \mu)$$

- For Euclidean distance, μ_i is simply the average of the samples in C_i

k-Means Algorithm: Formal Description

Consider $\mathcal{X} = \mathbb{R}^d$ and $d(x, y) = \|x - y\|^2$

1. Initialize cluster centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ randomly
2. Repeat until convergence
   - For $i = 1, \ldots, n$, set

$$c_i = \operatorname*{argmin}_j d(x_i, \mu_j)$$

   - For $j = 1, \ldots, k$, set

$$\mu_j = \frac{\sum_{i=1}^n \mathbb{1}\{c_i = j\} x_i}{\sum_{i=1}^n \mathbb{1}\{c_i = j\}}$$

Objective of k-Means Algorithm

Define the following objective function:

$$J(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i))^2 \qquad = \sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2$$

- Clustering can also be considered as lossy data compression
- All points in a cluster are represented by the cluster center, introducing distortion.
- Above objective can be considered as the reconstruction error
- We need $\log_2(k)$ bits to represent all the clusters

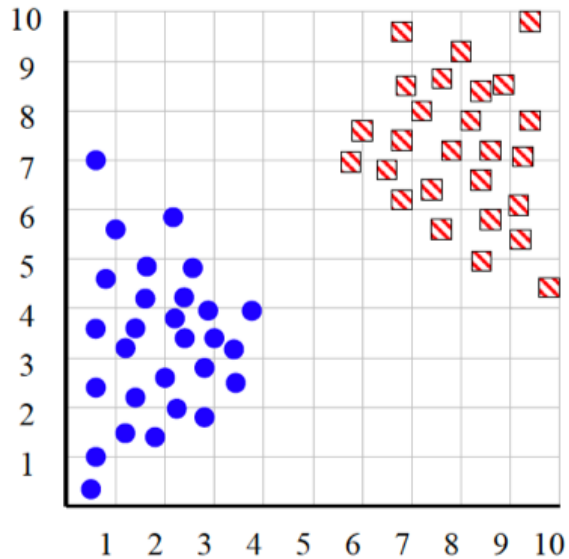
Convergence of k-Means Algorithm

$$J(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i))^2 \qquad = \sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2$$

- In each iteration, we keep μ_i's fixed and minimize J with respect to c_i's, then fix c_i's and minimize J with respect to μ_i's
- J monotonically decreases, hence must converge
- In theory it can oscillate between two or more clusterings (with the same J value (almost never happens in practice)
- J is a non-convex function, so k-means algorithm is not guaranteed to converge to a global minimum
- Recommendation: run several times with different initialization, and pick the result with the smallest objective function

How to Choose the Right Number of Clusters

- In general, we don't know the answer
- Consider the following dataset

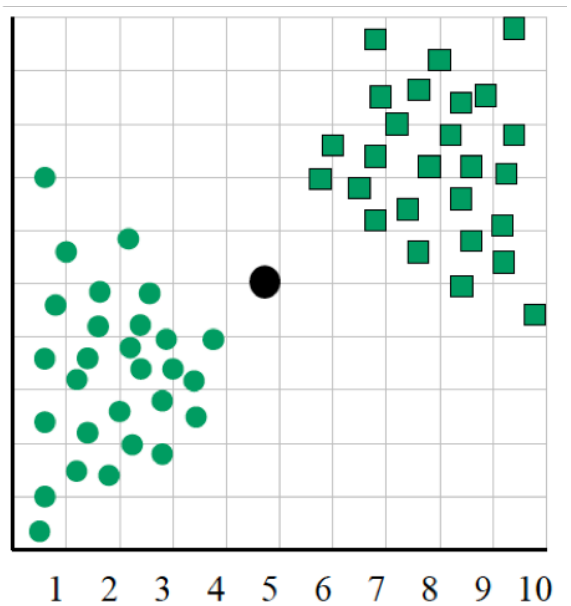


2

[2] Example from T. Mitchell, 10601, Fall 2012.

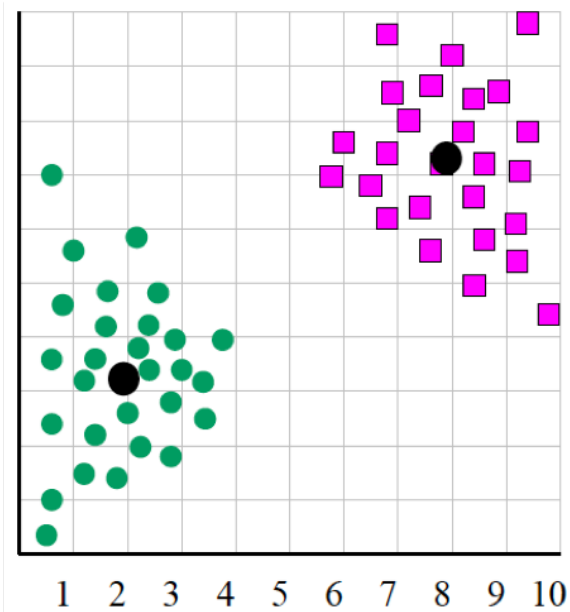
How to Choose the Right Number of Clusters

- For $k = 1$, the objective function is 873



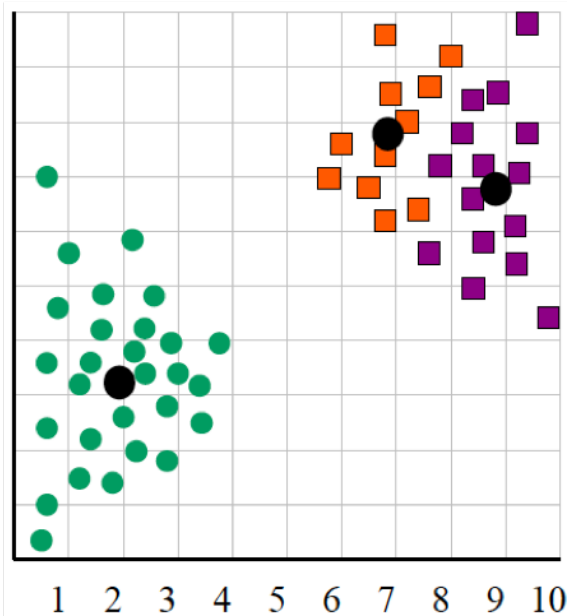
How to Choose the Right Number of Clusters

- For $k = 2$, the objective function is 173.1



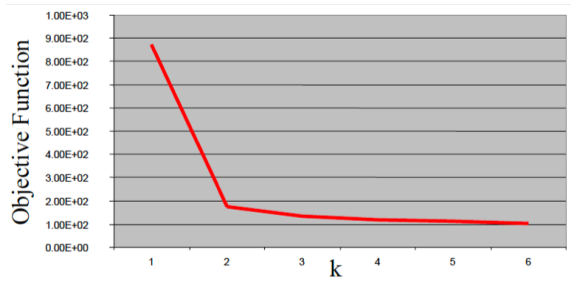
How to Choose the Right Number of Clusters

- For $k = 3$, the objective function is 133.6



How to Choose the Right Number of Clusters

- If we plot the objective function for $k = 1, 2, \ldots, 6$



- Abrupt change at $k = 2$ suggests presence of two clusters in the data
- This technique for determining the number of clusters is known as "knee finding" or "elbow finding"
- Not always as clear as this example!