Machine Learning

Lecture 7

Data Clustering

Motivating Problems

- A true colour image – 24bits/pixel, R – 8 bits, G – 8 bits, B – 8 bits

1677216 possible colours

- A gif image - 8bits/pixel

256 possible colours

Motivating Problems

(r, g, b)

For each pixel in the original image

Find the closest colour in the Colour Table

r_0	g_0	b_0
r_1	g_1	b_1
⋮	⋮	⋮
r_{255}	g_{255}	b_{255}

Colour Table

Record the index of that colour (for storage or transmission)

To reconstruct the image, place the indexed colour from the Colour Table at the corresponding spatial location

Motivating Problems

(r, g, b)

For each pixel in the original image

Find the closest colour in the Colour Table

r_0	g_0	b_0
r_1	g_1	b_1
⋮		
r_{255}	g_{255}	b_2

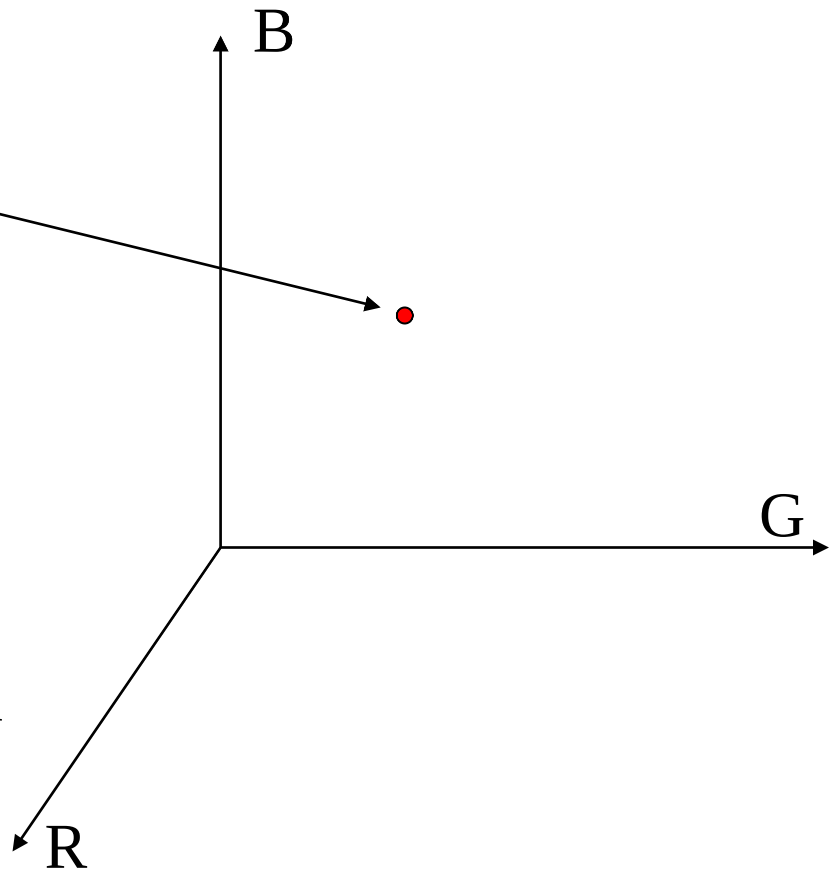
Colour Table

How to choose the colours in the table?

the
indexed colour from the Colour Table at the corresponding spatial location

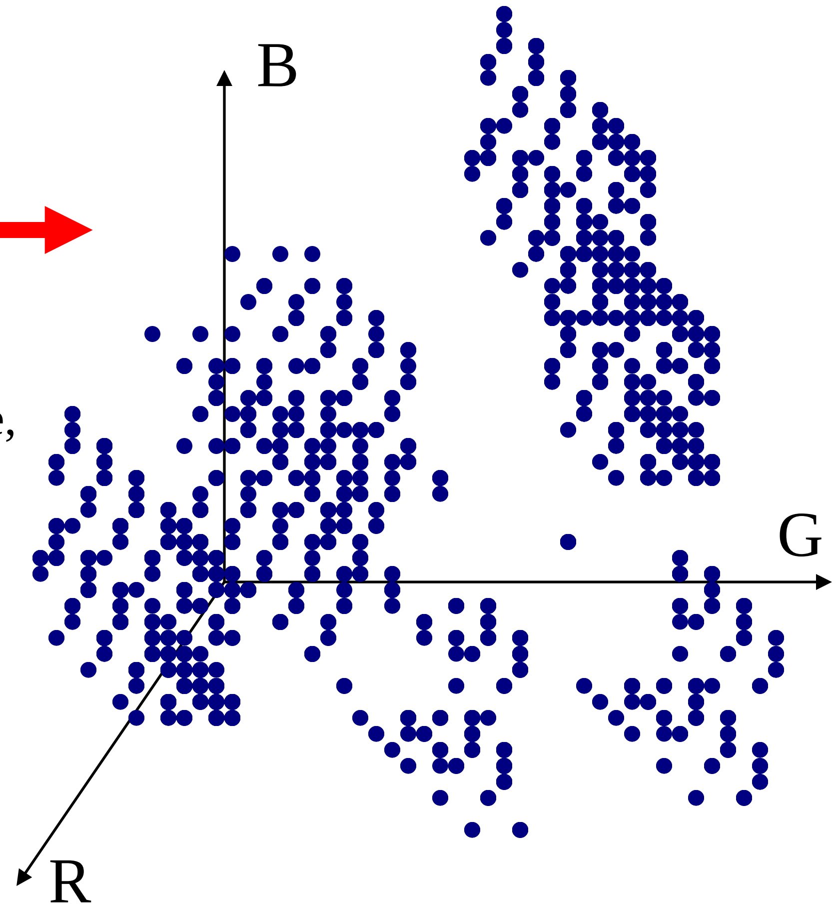
Motivating Problems

(r, g, b)



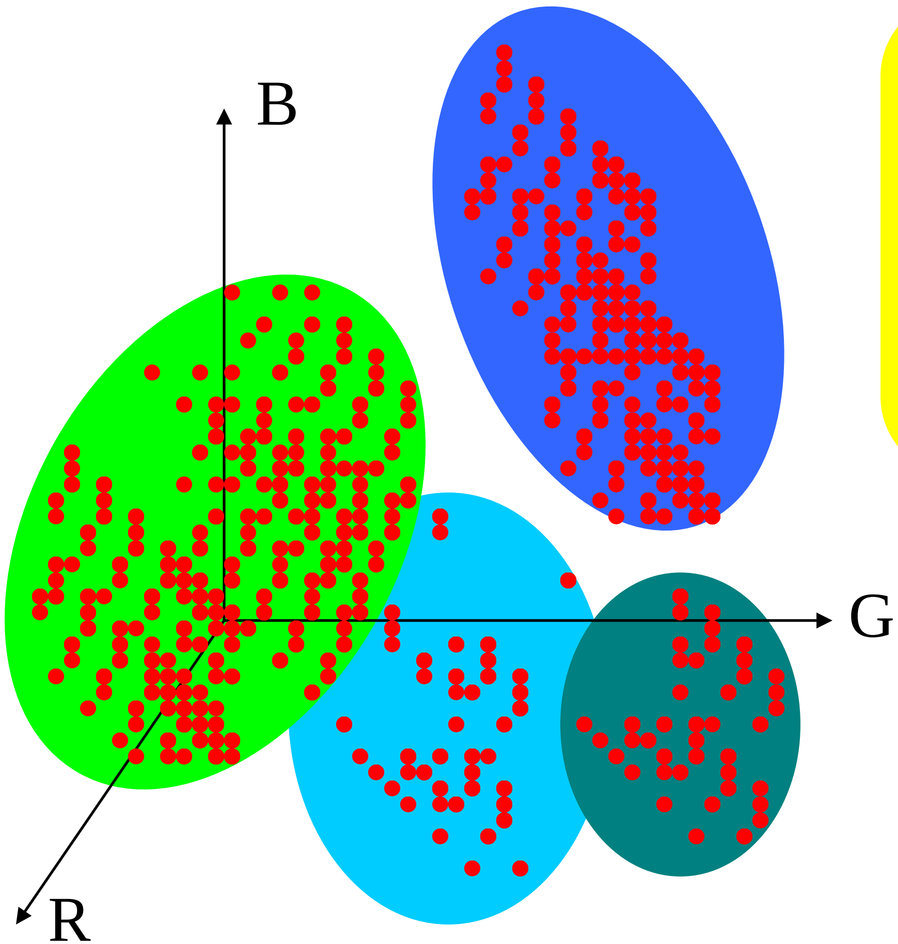
A pixel corresponding to
A point in the 3 dimensional
R, G, B space

Motivating Problems



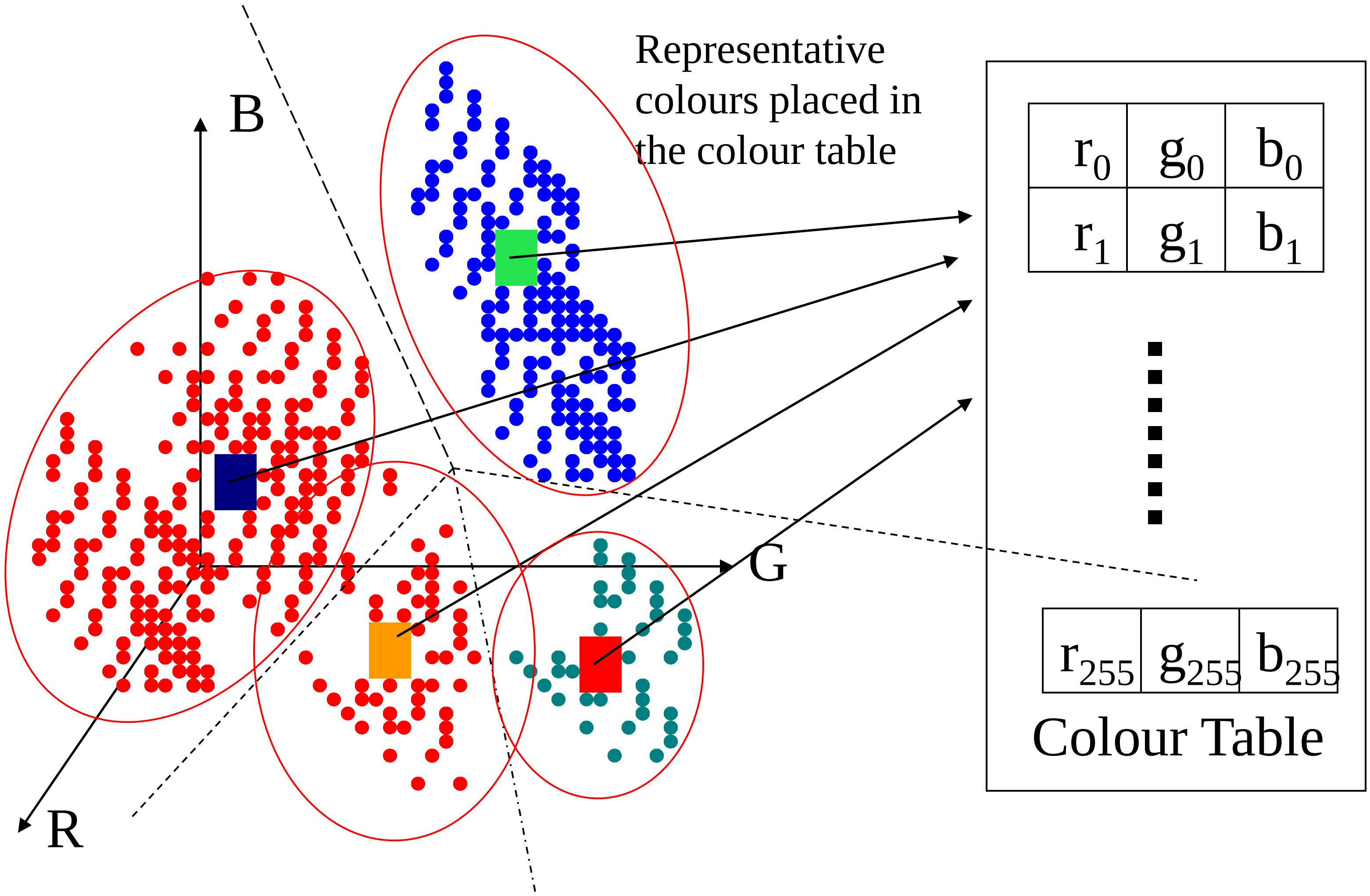
Map all pixels into the R, G, B space, "clouds" of pixels are formed

Motivating Problems



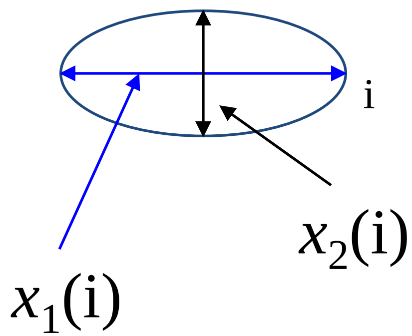
Group pixels that are close to each other, and replace them by one single colour

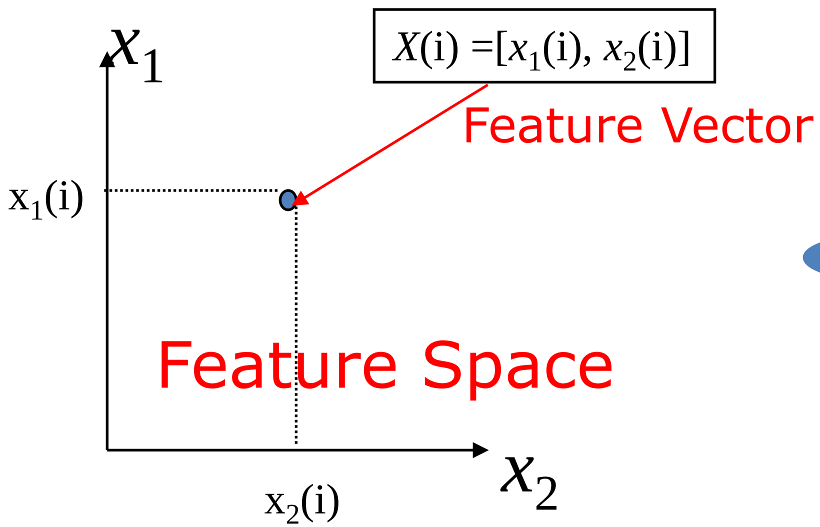
Motivating Problems

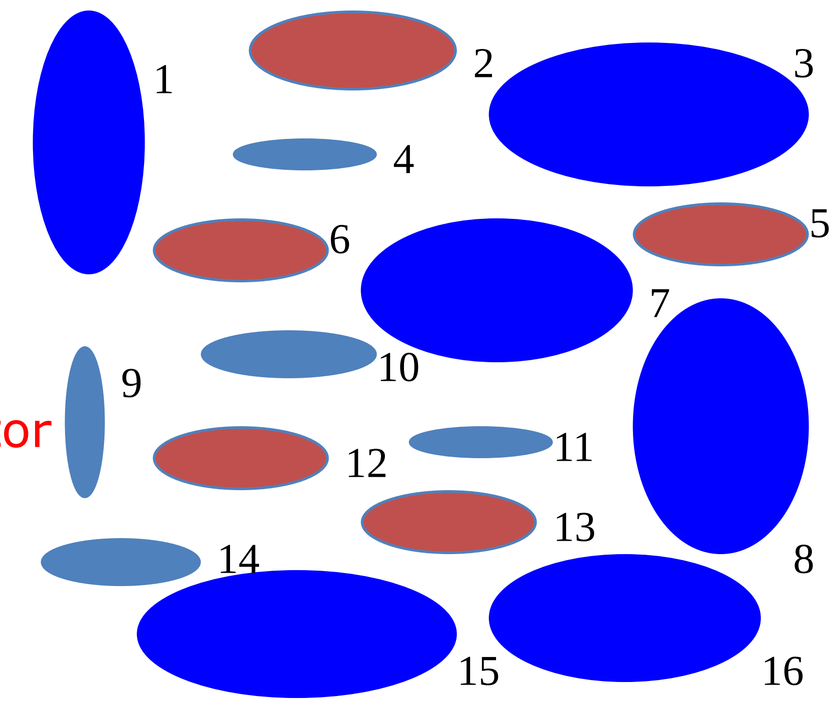


Motivating Example

- Classify objects (Oranges, Potatoes) into large, middle, small sizes



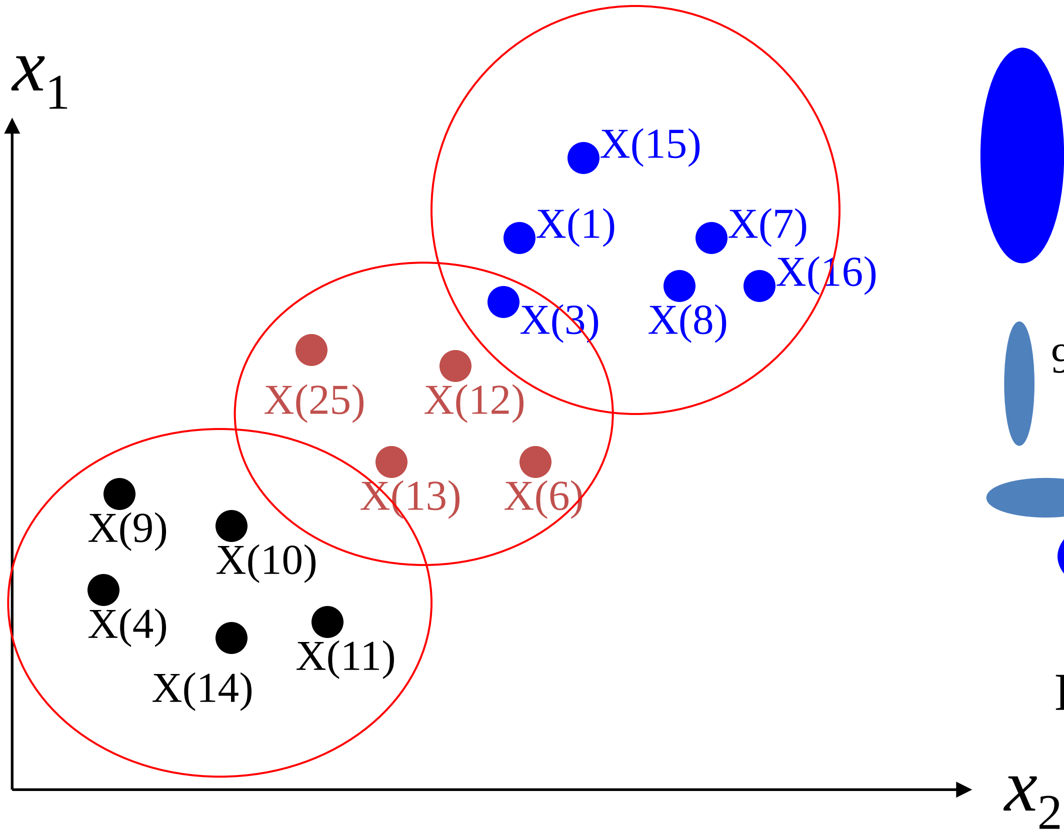


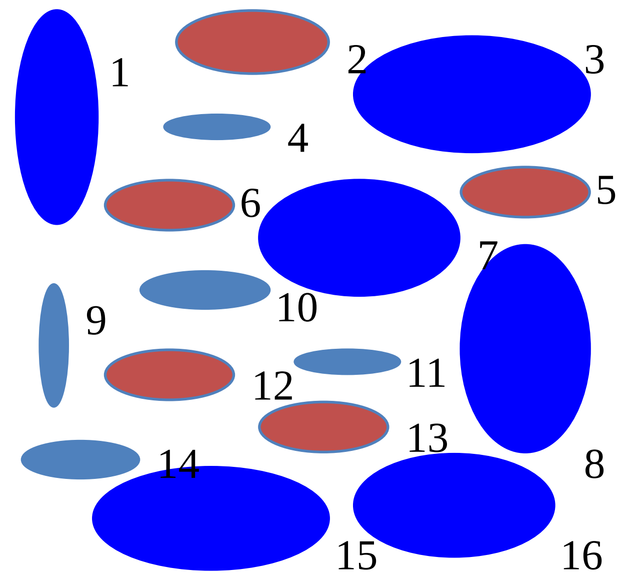


Elliptical blobs (objects)

Motivating Example

- From Objects to Feature Vectors to Points in the Feature Space





Elliptical blobs (objects)

K-Means

- An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing N_j data points

  - Define, $X(i) = [x_1(i), x_2(i), \ldots, x_n(i)]$, $i = 1, 2, \ldots N$, as N data points

  - We want to cluster these N points into K subsets, or K clusters, where K is pre-set

  - For each cluster, we define $M(j) = [m_1(j), m_2(j), \ldots, m_n(j)]$, $j=1, 2, \ldots K$, as its prototype or cluster centroids

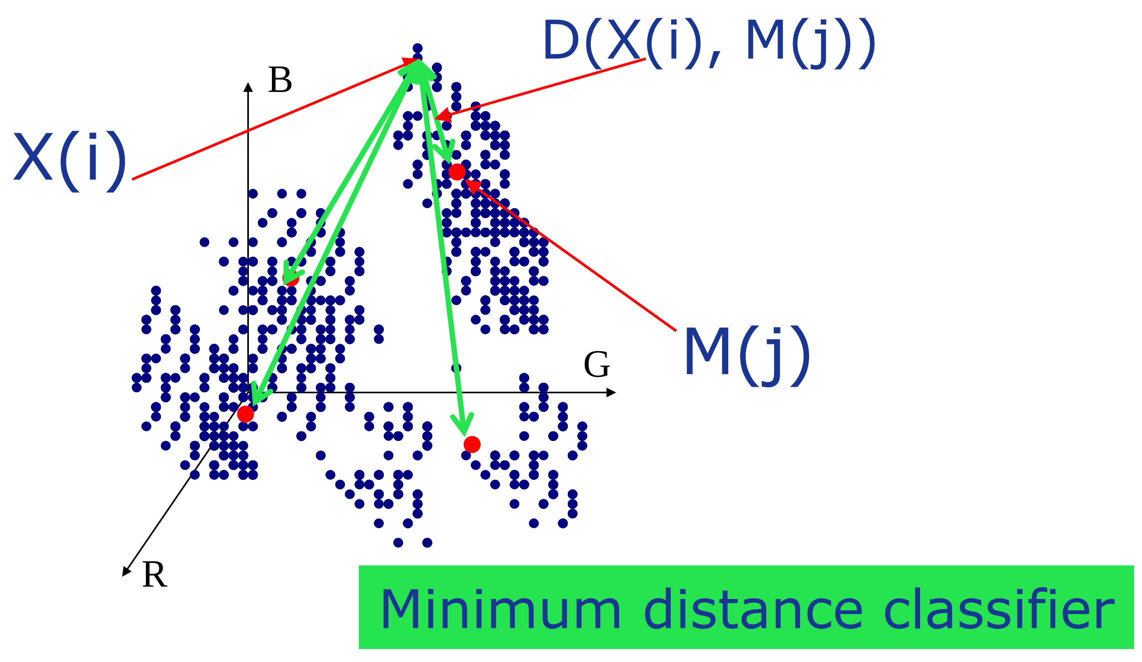
  - Define the distance between data point X(i) and cluster prototype M(j) as

$$D\big(X(i), M(j)\big) = \|X(i) - M(j)\|^2 = \sum_{l=1}^{n} \big(x_l(i) - m_l(j)\big)^2$$

K-Means

- A data point X(i) is assigned to the jth cluster, C(j), X(i) $\in$ C(j), if following condition holds

$$D\big(X(i), M(j)\big) \le D\big(X(i), M(l)\big) \quad \text{for all } l = 1, 2, \ldots, k$$



K-Means Algorithm

Step 1

- Arbitrarily choose from the given sample set k initial cluster centres,

$M^{(0)}(j) = [m^{(0)}_1(j), m^{(0)}_2(j), ,..., m^{(0)}_n(j)]$ $j = 1, 2, ..., K,$

e.g., the first K samples of the sample set
or can also be generated randomly

Set $t = 0$ (t is the iteration index)

K-Means Algorithm

Step 2

- Assign each of the samples $X(i) = [x_1(i), x_2(i), \ldots, x_n(i)]$, $i = 1, 2, \ldots N$, to one of the clusters according to the distance between the sample and the centre of the cluster:

$$
\begin{aligned}
& X(i) \in C^{(t)}(j) \\
& if \;\; D\left(X(i), M^{(t)}(j)\right) \leq D\left(X(i), M^{(t)}(l)\right) \\
& \text{for all } l \; = \; 1,\; 2,\; \ldots,\; k
\end{aligned}
$$

K-Means Algorithm

Step 3

Update the cluster centres to get

$M^{(t+1)}(j) = [m^{(t+1)}_1(j), m^{(t+1)}_2(j), ,\ldots, m^{(t+1)}_n(j)]$; $j = 1, 2, \ldots, K$

according to

$$M^{(t+1)}(j) = \frac{1}{N_j^{(t)}} \sum_{X(i) \in C^{(t)}(j)} X(i)$$

$N^{(t)}_j$ is the number of samples in $C^{(t)}_j$

K-Means Algorithm

Step 4

– Calculate the error of approximation

$$E(t) = \frac{1}{2} \sum_{j=1}^{K} \sum_{X(i) \in C^{(t)}(j)} \left\| X(i) - M^{(t)}(j) \right\|^2$$

K-Means Algorithm

Step 5

– If the terminating criterion is met, then stop, otherwise

Set $t = t+1$

Go to Step 2.

K-Means Algorithm

Stopping criterions

– The K-means algorithm can be stopped based on following criterions

1. The errors do not change significantly in two consecutive epochs

   $|E(t)-E(t-1)|<\varepsilon$, where ε is some preset small value

2. No further change in the assignment of the data points to clusters in two consecutive epochs.

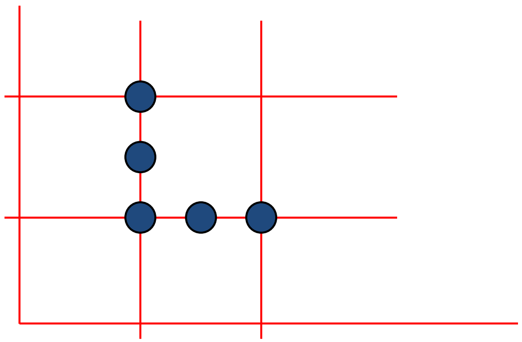
3. It can also stop after a fixed number of epochs regardless of the error

K-Means Algorithm

A worked example to see how it works exactly

Five 2-dimensional data points

(1, 1); (1.5, 1), (2, 1), (1.5, 1), (2, 1)



Cluster them into two clusters and find the cluster centres

K-Means Algorithm

- What is the algorithm doing exactly?
  - It tries to find the centre vectors M(j)'s that optimize the following cost function

$$E = \frac{1}{2}\sum_{j=1}^{K}\sum_{X(i)\in C(j)}\left\|X(i)-M(j)\right\|^2$$

K-Means Algorithm

- What is the algorithm doing exactly?

$$\frac{\partial E}{\partial m_l(j)} = \frac{\partial}{\partial m_l(j)}\left(\frac{1}{2}\sum_{j=1}^{K}\sum_{X(i)\in C(j)}\sum_{l=1}^{n}\left(x_l(i)-m_l(j)\right)^2\right)$$

$$= \frac{\partial}{\partial m_l(j)}\left(\frac{1}{2}\sum_{X(i)\in C(j)}\sum_{l=1}^{n}\left(x_l(i)-m_l(j)\right)^2\right)$$

$$= \sum_{X(i)\in C(j)}\left(x_l(i)-m_l(j)\right)\frac{\partial\left(x_l(i)-m_l(j)\right)}{\partial m_l(j)}$$

$$= -\sum_{X(i)\in C(j)}\left(x_l(i)-m_l(j)\right)$$

K-Means Algorithm

- What is the algorithm doing exactly?

$$\frac{\partial E}{\partial m_l(j)} = 0 \longrightarrow -\sum_{X(i)\in C(j)} \left(x_l(i) - m_l(j)\right) = 0 \longrightarrow \sum_{X(i)\in C(j)} x_l(i) = N_j m_l(j)$$

$$\longrightarrow m_l(j) = \frac{1}{N_j} \sum_{X(i)\in C(j)} x_l(i) \longrightarrow M(j) = \frac{1}{N_j} \sum_{X(i)\in C(j)} X(i)$$

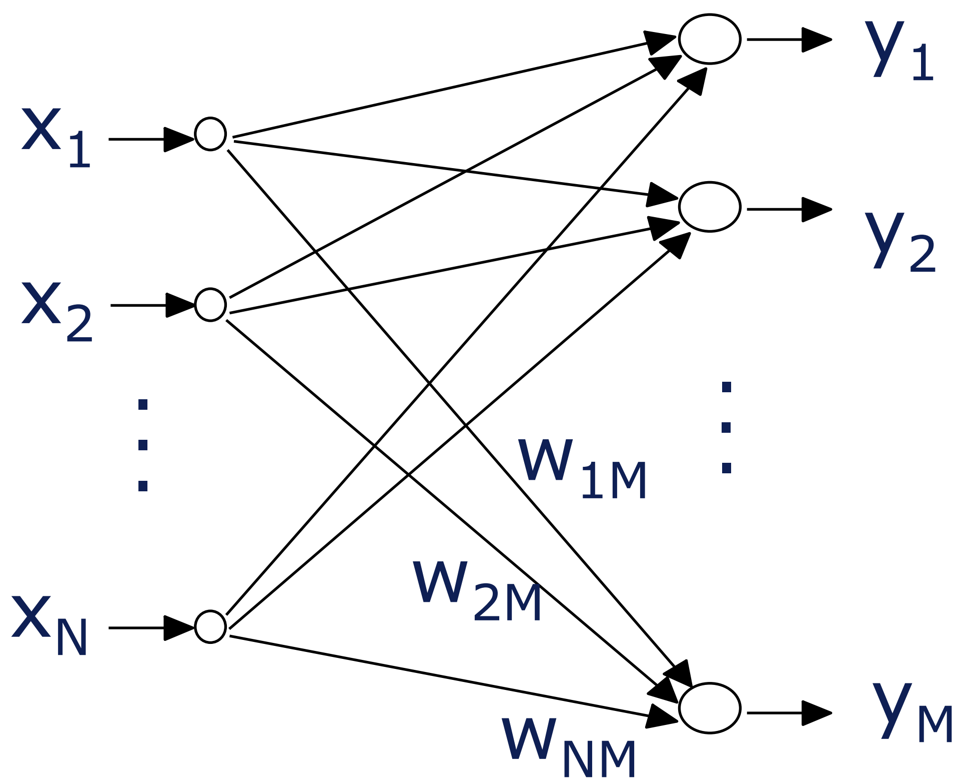
K-means cluster centre updating rule (Step 3)

K-Means Algorithm

- Some remarks
  - Is a gradient descent algorithm, trying to minimize a cost function *E*
  - In general, the algorithm does not achieve a global minimum of E over the assignments.
  - Sensitive to initial choice of cluster centers. Different starting cluster centroids may lead to different solution
  - Is a popular method, many more advanced methods derived from this simple algorithm.

competitive learning neural networks

- This type of network usually has one layer of fan-out units and one layer of processing units:



competitive learning neural networks

- The processing layer consists of M processing units, each receiving N input signals from the fan-out units. The x_i input to processing unit j has a weight w_{ij} assigned to it.

- The output of the processing units compete on the basis of which of them has its weight vector, $W_j = [w_{1j}, w_{2j}, \dots w_{Nj}]$, for all j, closest to the input vector X (as measured by a distance function D).

- The winner unit generates an output signal of 1; all the others units having outputs of 0

competitive learning neural networks

- Each processing unit calculates the distance between the input vector and the weight vector connecting the input to it, the activation of the *i*th processing units is

$$a_i = D(W_i, X) \qquad i = 1,2,\cdots M$$

- $D(W_i, X)$ is a distance measurement function, the most common choice for this is the Euclidean distance.

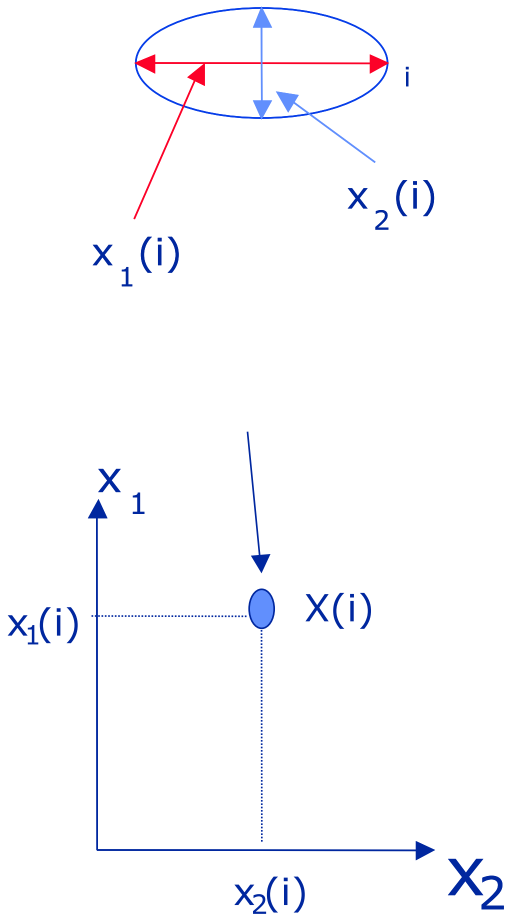
- Once each processing unit has calculated its activation, a competition takes place to see which output unit has the smallest activation value

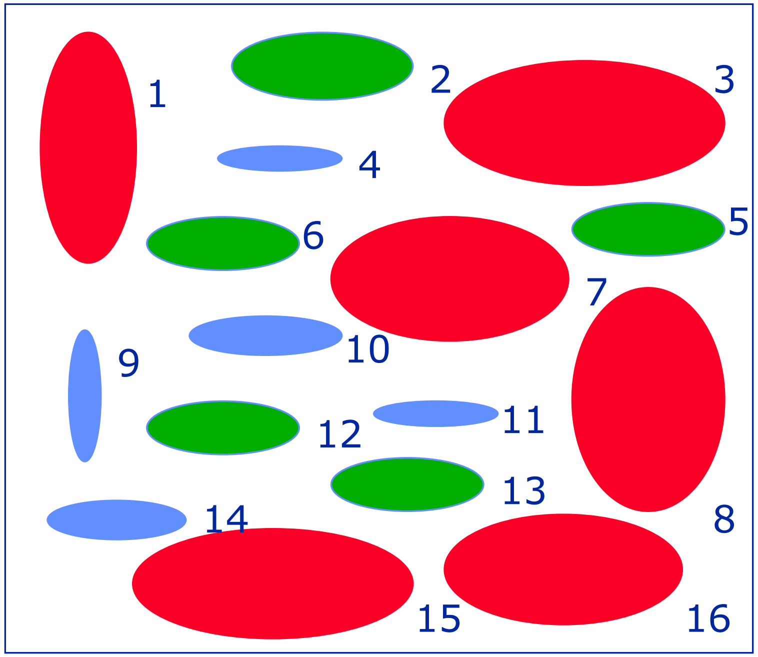
- This implies finding the unit which has its associated weight vector closest to the input vector *X*. The unit with the smallest activation value is declared as winner, all others are losers.

competitive learning neural networks

- The aim of such network is to cluster the input data
- Similar inputs should be classified as being in the same cluster
- There is no known desired outputs
- The outputs are found by the network itself from the correlation of the input data
- Such a network is also called **self-organising or unsupervising** neural network
- (Unsupervised) Competitive Learning

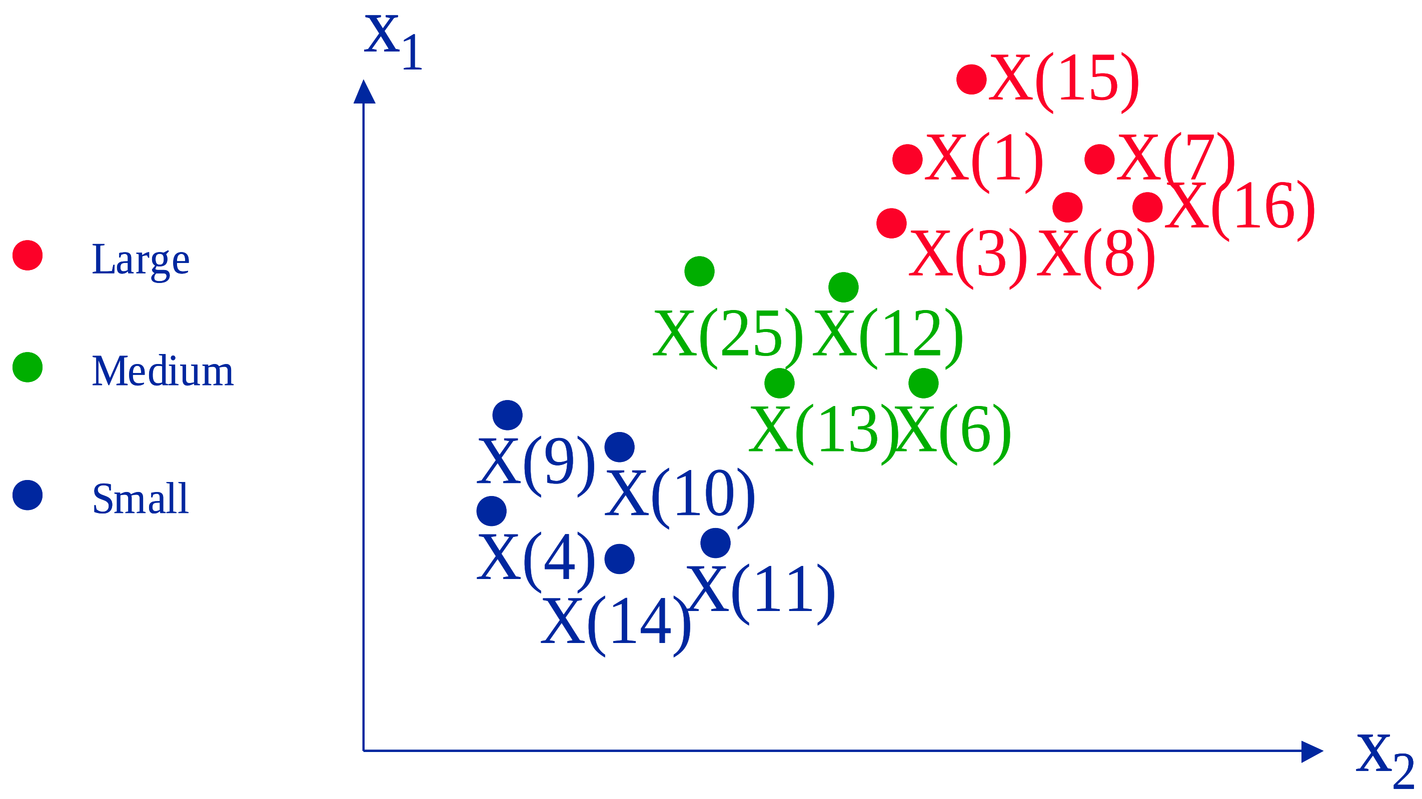
Competitive learning algorithms



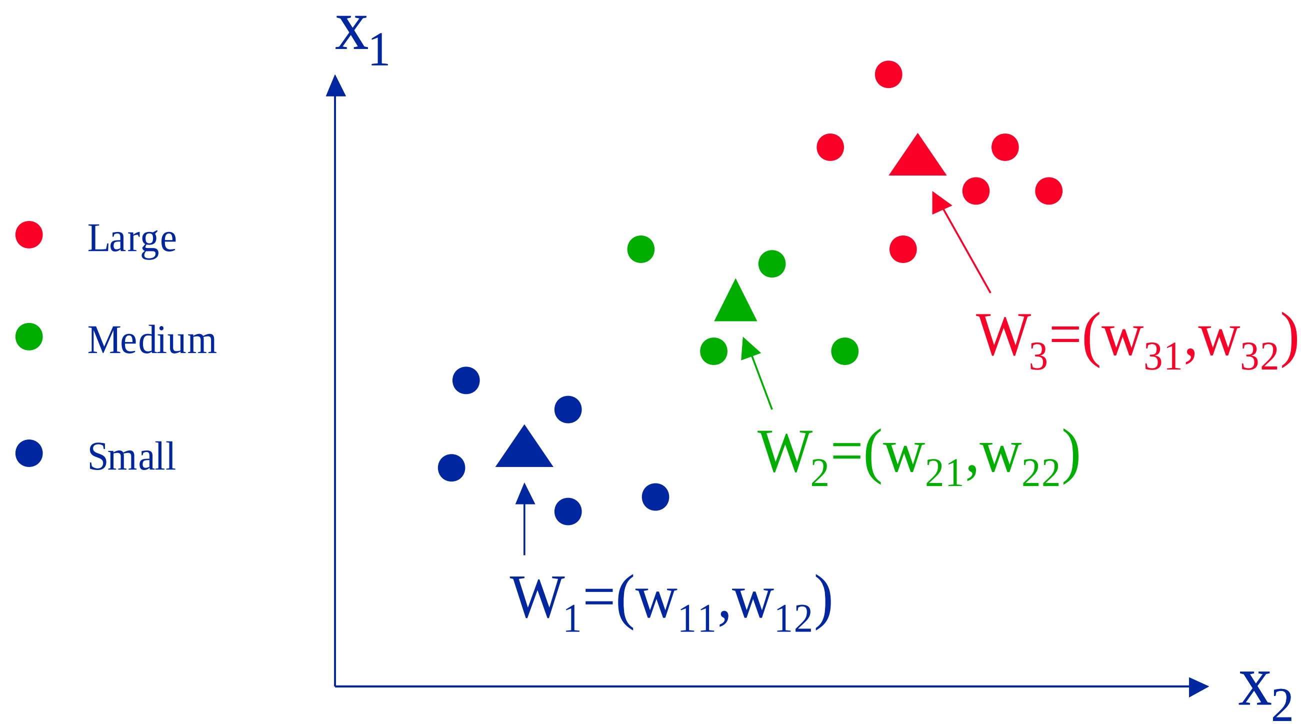


Task: Classify the elliptical shape blobs into three sizes: Large, Medium and Small

Competitive learning algorithms



Competitive learning algorithms

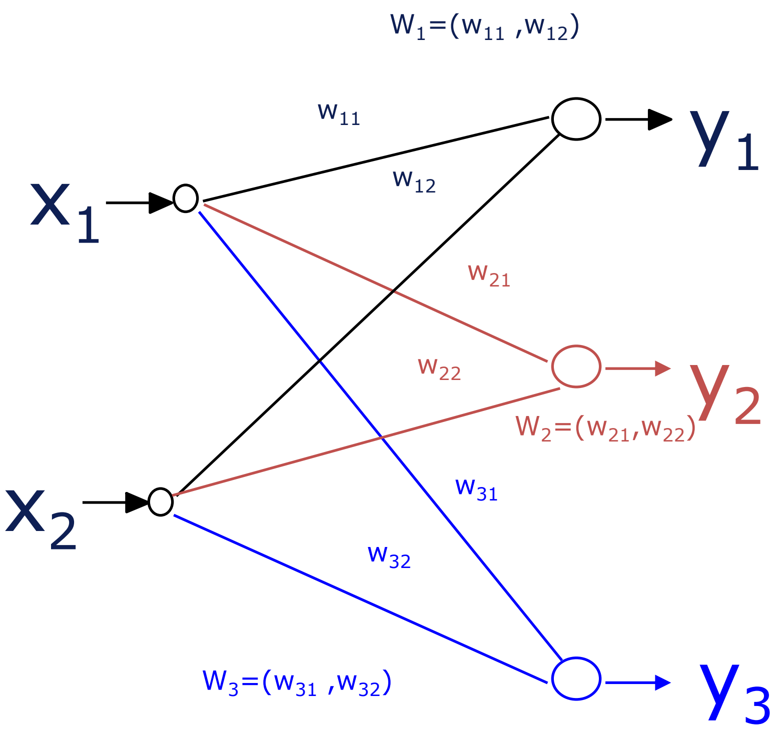


W_3 is the prototype (mean) vector of the large size blobs

W_2 is the prototype (mean) vector of the medium blobs

W_1 is the prototype (mean) vector of the small size blobs

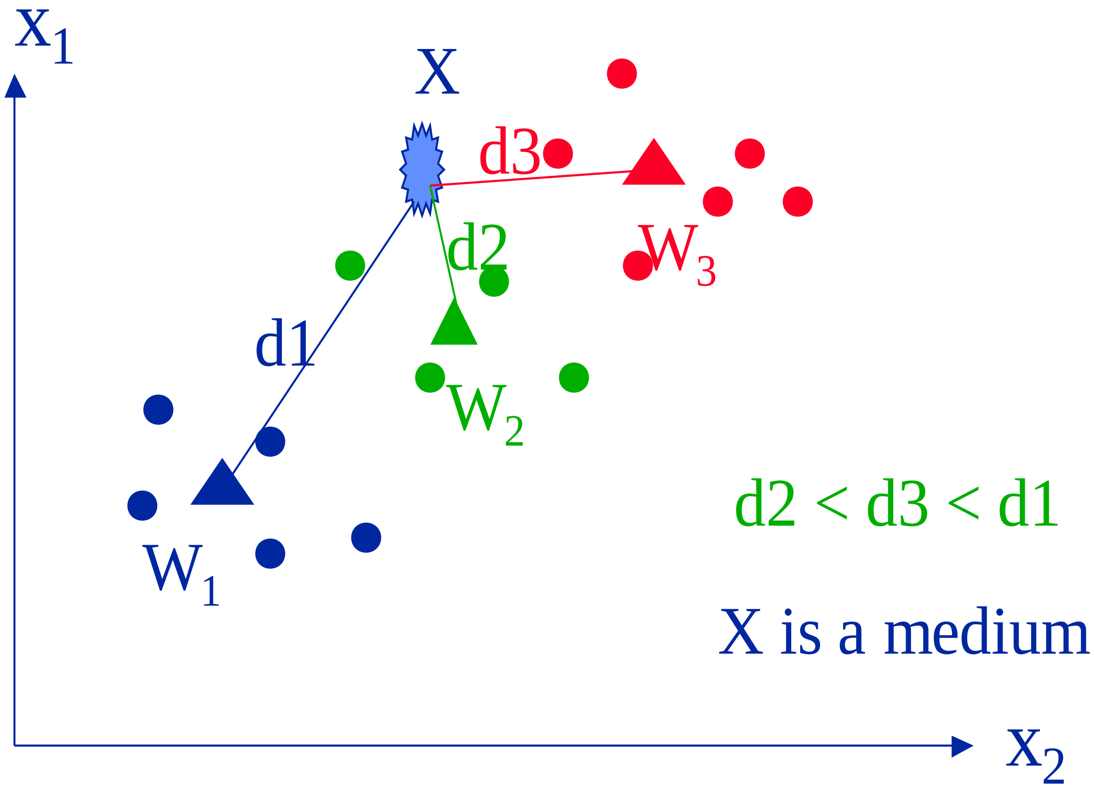
Competitive learning algorithms



$$y_i = \begin{cases} 1 & if \quad \|X - W_i\| \le \|X - W_j\|, j = 1,2,\ 3 \\ 0 & \text{otherwise} \end{cases}$$

Competitive learning algorithms

Minimum Distance Classifier



d2 < d3 < d1

X is a medium size blob

Competitive learning algorithms

Competitive Learning

- Competition

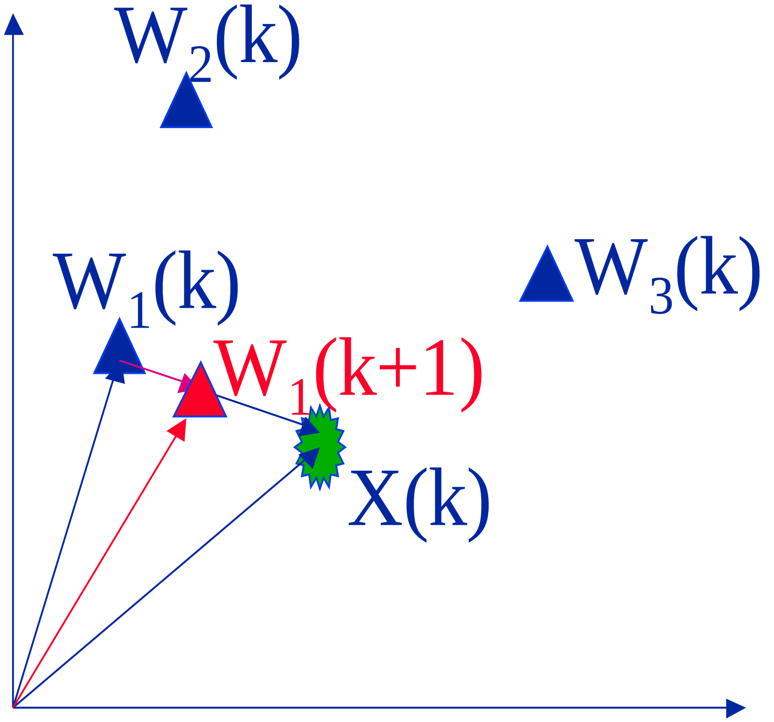
$$y_i(k) = \begin{cases} 1 & if \quad \left\| X(k) - W_i(k) \right\| \le \left\| X(k) - W_j(k) \right\|, \; j = 1, 2, \cdots, M \\ 0 & \text{otherwise} \end{cases}$$

- Learning

$$W_i(k+1) = W_i(k) + \alpha y_i(k)\left(X(k) - W_i(k) \right)$$

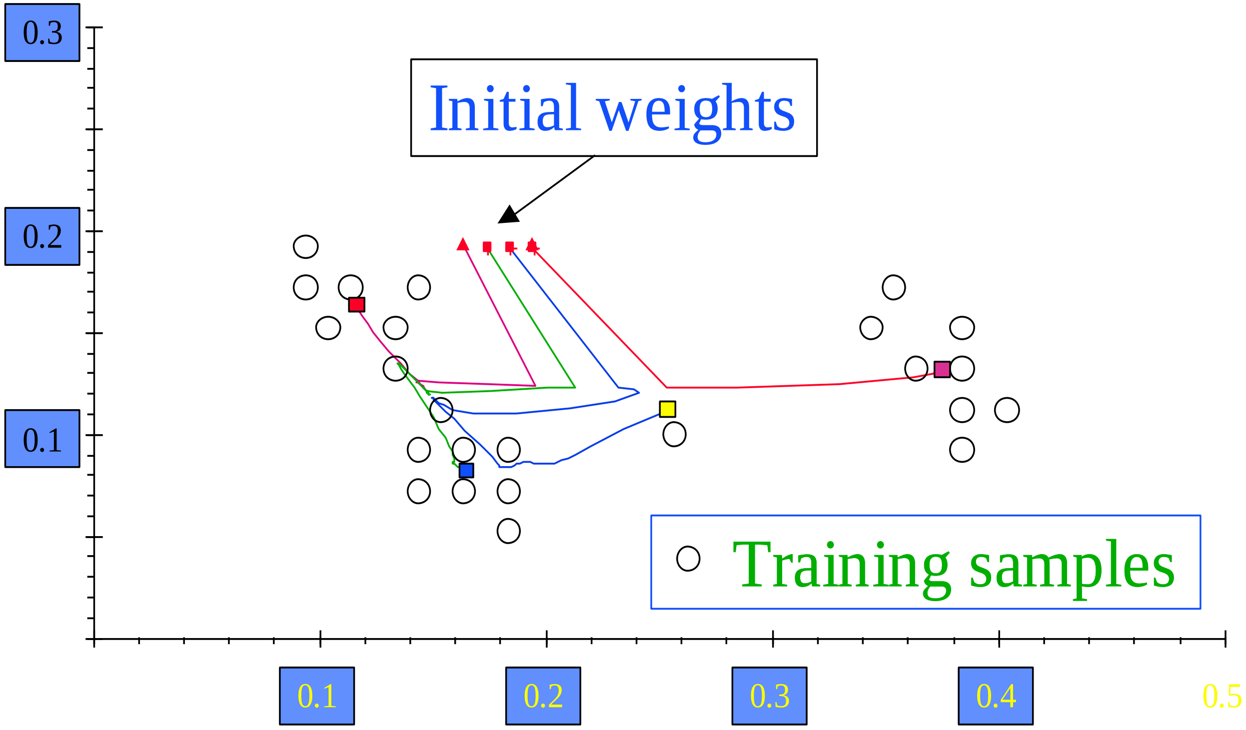
Competitive learning algorithms

Competitive Learning



Competitive learning algorithms

A Learning Example



Cost Function of Competitive Network

$$E\left(X(i)\right) = \sum_{j=1}^{M} y_j\left(i\right)\left\|X(i) - W_j\left(i\right)\right\|^2 = \sum_{j=1}^{M} y_j\left(i\right)\sum_{l=1}^{n}\left(x_l(i) - w_{jl}\left(i\right)\right)^2$$

- Derive the competitive learning algorithm using gradient descent:

$$\frac{\partial E}{\partial w_{jl}} = -2y_j\left(i\right)\left(x_l(i) - w_{jl}\left(i\right)\right)$$

Relation to k-means Algorithm

- Batch
- Online

Further Readings

- Many tutorial material on the Internet and relevant textbooks

Tutorial/Exercise Questions

1. An application of the k-means algorithm to a 2 dimensional feature space has produced following three cluster prototypes: M1 = (1, 2), M2 = (2, 1) and M3 = (2, 2). Determine which cluster will each of the following feature vectors be classified into.

   (i) X1 = (1, 1)
   (ii) X2 = (2, 3)

2. The k-means algorithm has been shown to minimize the following cost function. Derive an "online version" of the k-means algorithm following similar ideas as the delta rule. In this case, the prototype will be updated every time a training sample is presented for training, this is in contrast to k-means algorithm where only after all samples are presented the prototypes will be updated.

$$E = \frac{1}{2}\sum_{j=1}^{K}\sum_{X(i)\in C(j)}\left\|X(i)-M(j)\right\|^2$$