Lecture 20
Clustering (1)

Outline

- Unsupervised learning (Competitive Learning) and Clustering
- K-Means Clustering Algorithm

Unsupervised Learning

- Data Mining
  - Understand internal/hidden structure of data distribution
- Labeling (Target value, teaching input) Cost is High
  - Large amount of feature vectors
  - Sampling may involve costly experiments
  - Data label may not be available at all
- Pre-processing for classification
  - features within the same cluster are similar, and
  - often belong to the same class

Competitive Learning

- A form of unsupervised learning.
- Neurons compete against each other with their activation values. The winner(s) reserve the privilege to update their weights. The losers may even be punished by updating their weights in opposite direction.
- Competitive and Cooperative Learning:

  Competitive: Only one neuron's activation can be reinforced.

  Cooperative: Several neurons' activation can be reinforced.

Competitive Learning Rule

- A neuron WINS the competition if its output is largest among all neurons for the same input x(n).
- The weights of the winning neuron (k-th) is adjusted:

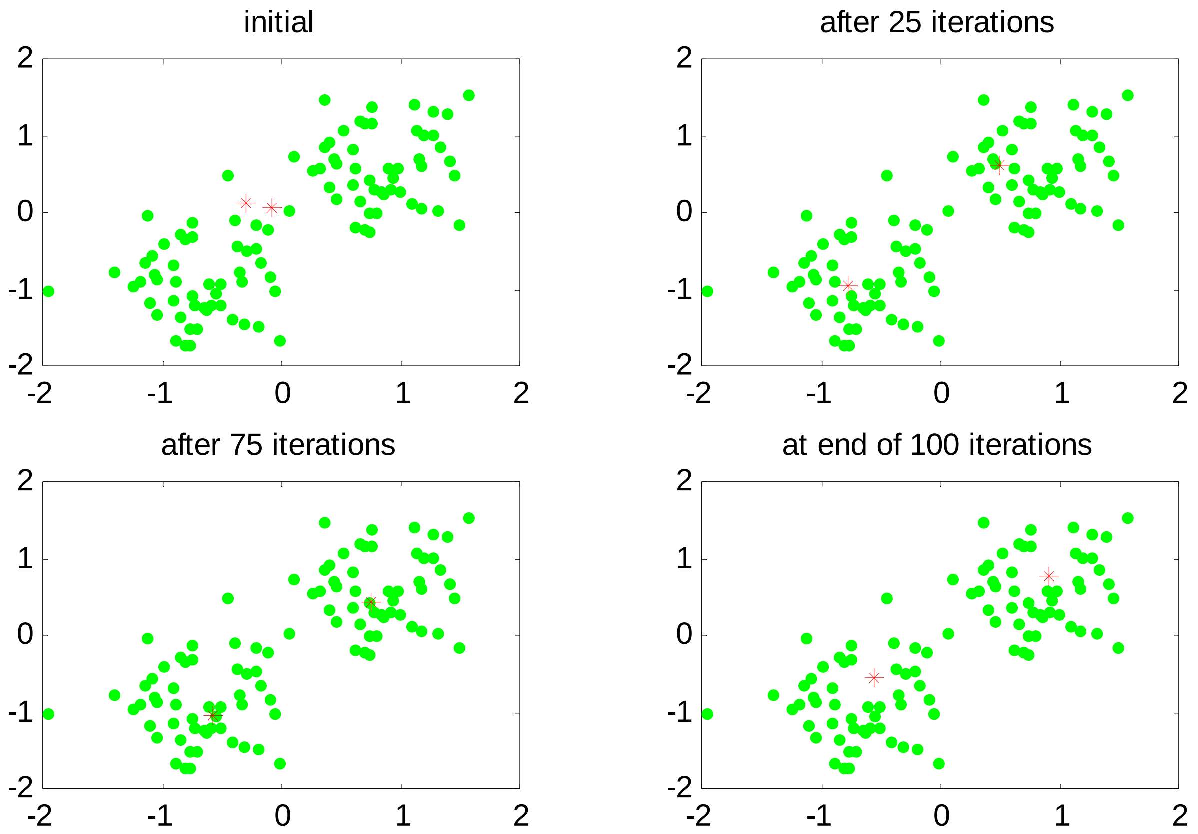
$$\Delta w_k(n) = [x(n) - w_k(n)]$$

The positions of losing neurons remain unchanged.

- If the weights of a neuron represents its POSITION. If the output of a neuron is inversely proportional to the distance between x(n) and $w_k(n)$, then

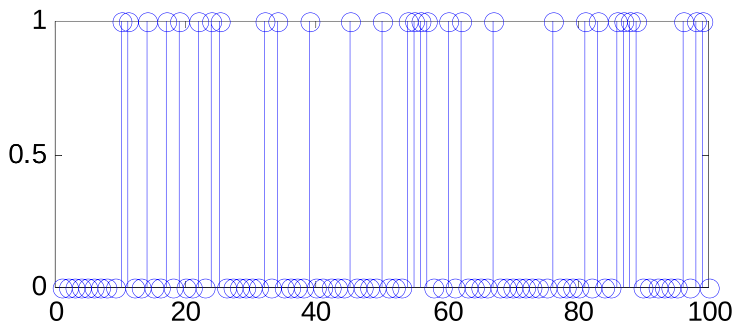
Competitive Learning = CLUSTERING!

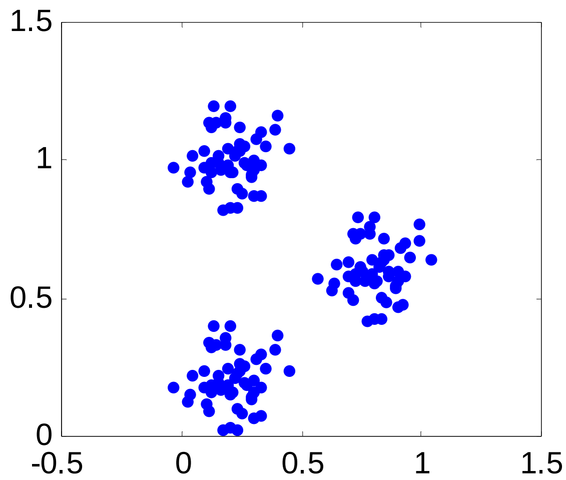
Competitive Learning Example



learncl1.m

What is "Clustering"?





What can we learn from these "unlabeled" data samples?

- Structures: Some samples are closer to each other than other samples
- The closeness between samples are determined using a "similarity measure"
- The number of samples per unit volume is related to the concept of "density" or "distribution"

Clustering Problem Statement

- Given a set of vectors $\{x_k; 1 \le k \le K\}$, find a set of M clustering centers $\{\mathbf{w}(i); 1 \le i \le c\}$ such that each x_k is assigned to a cluster, say, $\mathbf{w}(i^*)$, according to a distance (distortion, similarity) measure $d(x_k, \mathbf{w}(i))$ such that the average distortion

$$D = \frac{1}{K}\sum_{i=1}^{c}\sum_{k=1}^{K} I(x_k, i) d(x_k, W(i))$$

is minimized.

- $\mathbf{I}(x_k,i) = 1$ if $d(x_k, \mathbf{w}(i)) < d(x_k, \mathbf{w}(j))$, $j \ne i$; and $= 0$ otherwise -- indicator function.

k-means Clustering Algorithm

Initialization: Initial cluster center **w**(i); $1 \le i \le c$, D(–1)= 0, **I**(x_k,i) = 0, $1 \le i \le c$, $1 \le k \le K$;

Repeat

(A) Assign cluster membership (Expectation step)

Evaluate d(x_k, **w**(i)); $1 \le i \le c$, $1 \le k \le K$

I(x_k,i) = 1 if d(x_k, **w**(i)) < d(x_k, **w**(j)), $j \neq i$;

= 0; otherwise. $1 \le k \le K$

(B) Evaluate distortion D: D(Iter) = **I**(x_k,i)d(x_k, **w**(i)) $1 \le k \le K$

(C) Update code words according to new assignment (Maximization)

$$W(i) = \sum_{k=1}^{N} I(x_k, i) x_k, \qquad N_i = \sum_{k=1}^{N} I(x_k, i), \qquad 1 \le i \le c$$

(D) Check for convergence

if 1–D(Iter–1)/D(Iter) < ε , then convergent = TRUE,

A Numerical Example

x = {-1, -2,0,2,3,4},

W={0.1, 0.3}

1. Assign membership

   0.1: {-1, -2, 0}

   0.3: {2, 3, 4}

2. Distortion

   $D = (-1-0.1)^2 + (-2-0.1)^2 + (0-0.1)^2 + (2-0.3)^2 + (3-0.3)^2 + (4-0.3)^2$

3. Update W to minimize distortion

   $W_1 = (-1-2+0)/3 = -1$

   $W_2 = (2+3+4)/3 = 3$

4. Reassign membership

   -1: {-1, -2, 0}

   3: {2, 3, 4}

5. Converged!

Kmeans Algorithm Demonstration