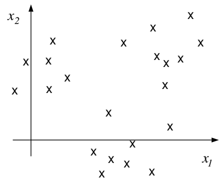
CLUSTERING

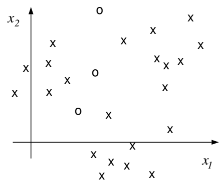
- What do you do when you have a *very large* data set with *very high dimensionality* and *lots of complex structure*?
- The safest thing to do to get started is to *cluster* the data
- We'll start by describing elementary clustering ideas that are widely used, and then generalize in a number of useful directions

The K-means algorithm

Consider a set of data described by a feature vector x:



The K-means algorithm starts with K randomly chosen means (let's assume that we use $K = 3$):

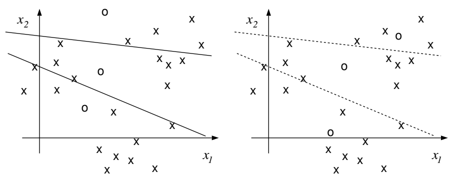


and iterates between two steps:

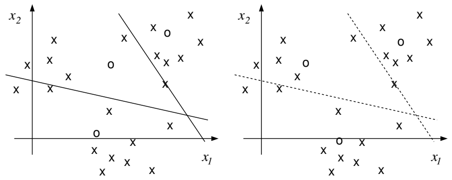
- partition the data according to the current values of the means
- move the means to the centroid of the data in the current partition

The K-means algorithm (cont.)

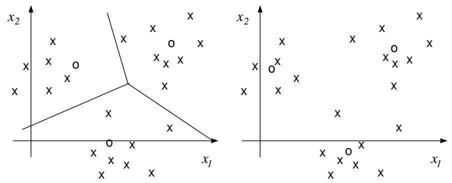
Iteration 1:



Iteration 2:

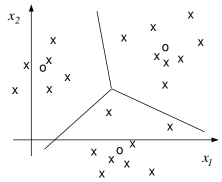


Iteration 3:



The K-means algorithm (cont.)

Iteration 4:



The fact that the partitions remained the same between *Iteration 3* and *Iteration 4* means that the algorithm has converged

Virtues of K-means:

- simplicity
- generality

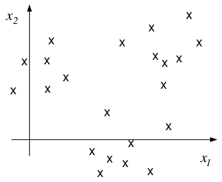
Drawbacks of K-means:

- can perform poorly with overlapping clusters
- lack of robustness to outliers
- 0/1 output is not very informative

Gaussian mixture models

- Gaussian mixture models are probabilistic ("soft") versions of K-means
- Slightly more complex to implement but tend to perform better

Again assume that we have some data to cluster:



We again assume that we have three means, but these means are now viewed as means of Gaussian distributions

Each Gaussian also has a covariance matrix

- we'll assume covariances of the form $\sigma^2 I$ for simplicity of presentation

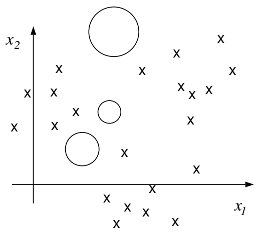
And we also allow "prior" weights π_i

Gaussian mixture models (cont.)

Thus we have the following assumption for the *class-conditional probabilities*, where q is the class label vector:

$$P(x|q^i = 1) = \frac{1}{(2\pi\sigma_i)^d} e^{-\frac{1}{2\sigma_i^2}\|x-\mu_i\|^2}$$

E.g., here we have three classes:



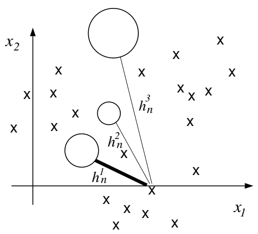
The "EM" algorithm for fitting a Gaussian mixture iterates between two steps:

- Compute the "responsibility" of each Gaussian for each data point (gives us a "soft partition")
- Move the means of the Gaussians to the centroid of the data, weighted by the responsibilities

Gaussian mixture models (cont.)

These two steps are called the "E step" and "M step", respectively (for "estimation" and "maximization")

E step:



where the "responsibility" h_n^i for data point x_n is computed as follows:

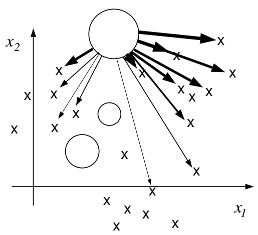
$$h_n^i = \frac{\frac{\pi_i}{\sigma_i^d} e^{-\frac{1}{2\sigma_i^2}\|x_n-\mu_i\|^2}}{\sum_j \frac{\pi_j}{\sigma_j^d} e^{-\frac{1}{2\sigma_j^2}\|x_n-\mu_j\|^2}}$$

which is a simple consequence of Bayes formula:

$$h_n^i \equiv P(q_n^i = 1|x_n) = \frac{P(q_n^i = 1)P(x_n|q_n^i = 1)}{\sum_j P(q_n^j = 1)P(x_n|q_n^j = 1)}$$

Gaussian mixture models (cont.)

M step:



In the M step, each mean is moved to the centroid of all the data, *weighted by the responsibilities*:

$$\mu_i^{(t+1)} = \frac{\sum_n h_n^i x_n}{\sum_n h_n^i}$$

There are similar update rules for the covariances and for the priors

Two points:

- in the limit of very narrow Gaussians, which yield posterior probabilities h_n^i that are zeros and ones, EM reduces to the K-means algorithm
- K-means can be a good way to initialize EM

Maximum likelihood justification

The EM algorithm doesn't just come out of a hat; it's a method for maximum likelihood estimation

The probability of a data point x_n is given by the sum over all ways of obtaining the data point (a *mixture* distribution:

$$P(x_n) = \sum_i \pi_i P(x_n | q_n^i = 1)$$

The log likelihood of a data set is thus:

$$l = \sum_n \log \sum_i \pi_i P(x_n | q_n^i = 1)$$

and the gradient with respect to μ_i is:

$$\frac{\partial l}{\partial \mu_i} = \sum_n \left[\frac{\frac{\pi_i}{\sigma_i^d} e^{-\frac{1}{2\sigma_i^2}\|x_n - \mu_i)\|^2}}{\sum_j \frac{\pi_j}{\sigma_j^d} e^{-\frac{1}{2\sigma_j^2}\|x_n - \mu_j)\|^2}} \right] (x_n - \mu_i)$$

where we see the responsibilities emerging, and the weighted "pull" toward the data points

A full derivation of EM is a bit more work, but not much

Comments on clustering

- K-means is a simple, effective procedure
- Gaussian mixtures yield a "soft" version of K-means
- the Gaussian mixture is a building block
- the idea of a "clustering distance function" can be generalized quite a bit. By giving clustering a probabilistic foundation, we now see that we can replace "clustering distance function" by "likelihood under a model." And essentially *any* model can be used to form the basic clustering element. This allows us to construct "piecewise" or "local" versions of many other learning architectures.