Unsupervised Learning

Part1: Clustering

Ioannis Patras

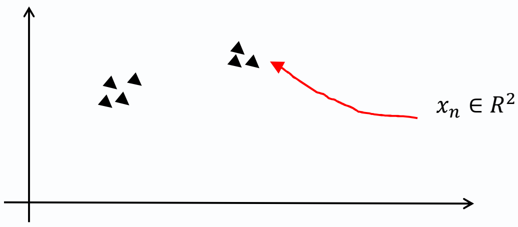
ECS708 Machine Learning

Unsupervised Learning

By contrast to supervised learning the training set consists of only data points (no 'target data').

Goal: Find some structure in the data

$$X = \{x_n \mid n = 1 \ldots m\}$$



Slide no: 0-2

Clustering

Given the hue and size (features) of fruits, group them into clusters

Possible uses: Find hidden structure in the data

- news clustering, genes clustering, image clustering, audio clustering.

Slide no: 0-3

K - means clustering

Given: $X = \{x_n \mid i = 1 \ldots m\}$ $x_n \in R^N$ and the number of clusters K

Find: $\{c_n | n = 1 \ldots m\}, c_n \in [1 \ldots K]$ and $\{\mu_k | k = 1 \ldots \mathrm{K}\}, \mu_k \in R^N$

The K-Means clustering algorithm works as follows. First initialize K centres $\boldsymbol{\mu}_k$ (for example by picking K input samples randomly) then iteratively repeat the following steps.

Step - 1: classify x_n to the cluster k with the nearest centre μ_k

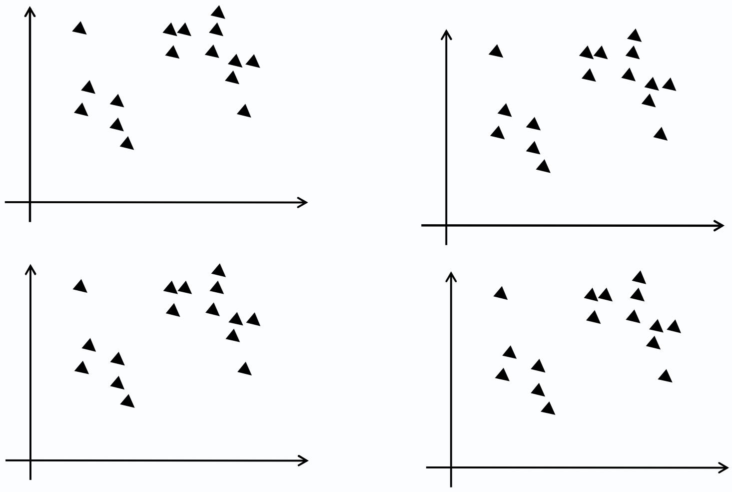
$c_n = k$(or $x_n \in C_k$) if $|x_n - \mu_k|_2 < |x_n - \mu_j|_2$ for all j $\neq k$ $\neq c_n$

Step - 2: re-compute μ_k to be the mean of the class C_k

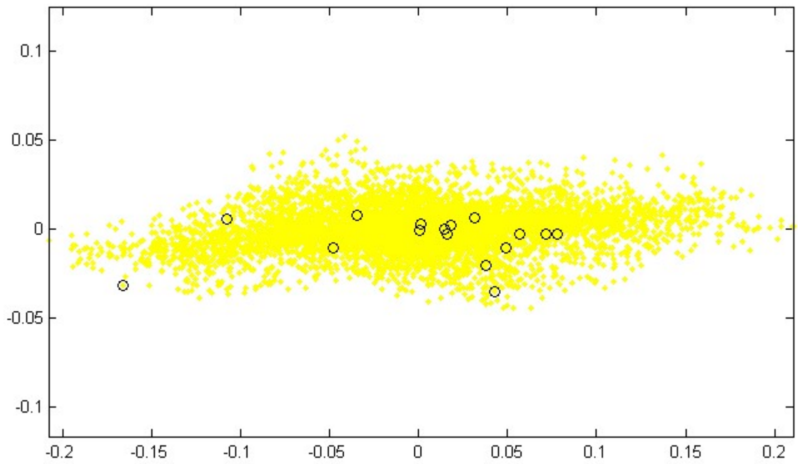
$$\mu_k = \sum_{x_n \in C_k} x_n \Big/ \sum_{x_n \in C_k} 1 = \frac{1}{|C_k|} \sum_{x_n \in C_k} x_n$$

Until the solution stabilizes (i.e. until classes of x_n stop changing in Step 1).

K - means clustering (example)

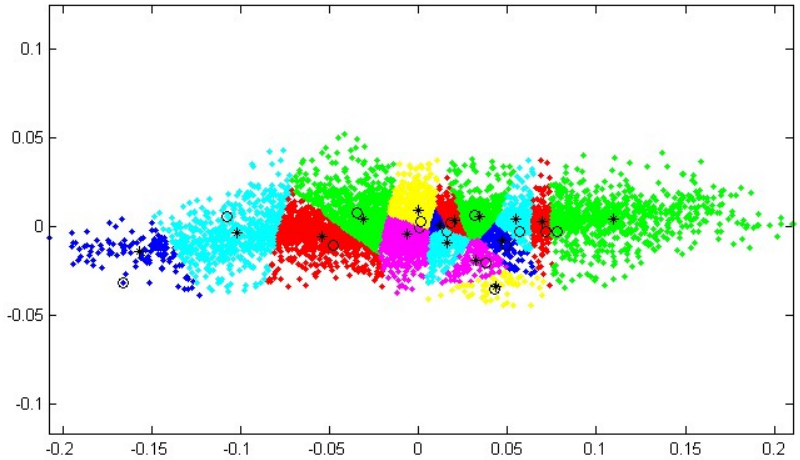


K - means: initialisation

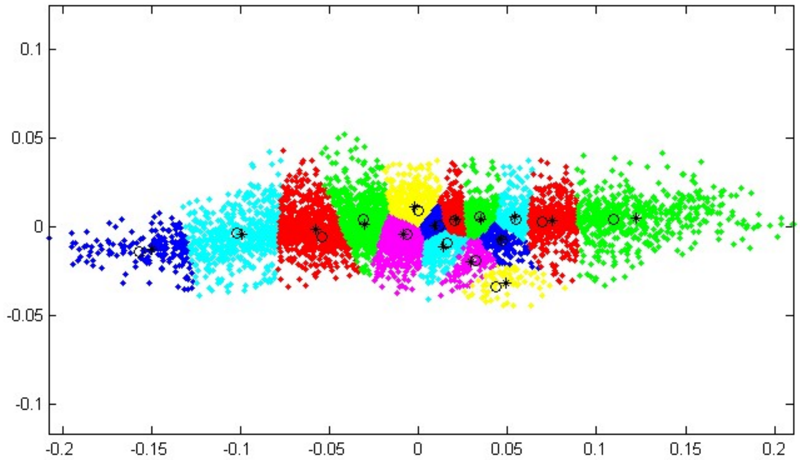


We have used random initialisation – more sophisticated schemes exist

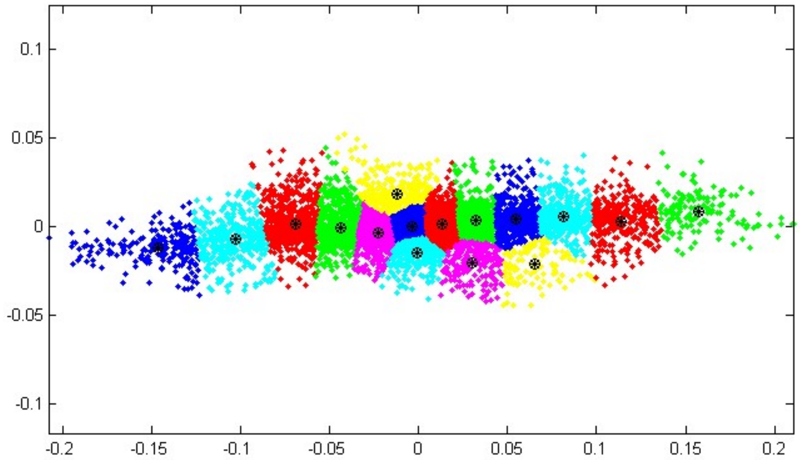
K - means: 1st iteration



K - means: 2nd iteration



K - means: converged (20th iteration)



Convergence of K - means

We can now show that the two steps of K-means optimise a cost function

$$J(c_1,\ldots,c_n,\mu_1,\ldots,\mu_K) = \sum_n \| x_n - \mu_{c_n} \|^2$$

Where $c_n \in [1 \ldots K]$ indicates the cluster to which $\mathbf{x_n}$ belongs.

Non convex function, may have local minima, difficult to optimize.

K-means follows an optimisation scheme called co-ordinate descent (or ascent if the objective needs to be maximised)

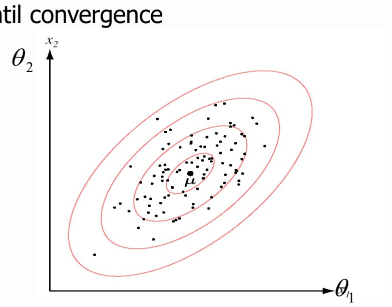
Coordinate descent

Minimize $J(\theta_1, \theta_2)$ by iterating between two steps:

Step 1: Keep θ_2 fixed and optimise w.r.t. θ_1

Step 2: Keep θ_1 fixed and optimise w.r.t θ_2

Until convergence



Given some conditions, the procedure converges to a local minimum

"Optimality" of K - means (1)

Cost function $J(c_1,\dots,c_m,\mu_1,\dots,\mu_K) = \sum_n \| x_n - \mu_{c_n} \|^2$

Step 1: Optimise w.r.t $c_1,\dots,c_m$

$$c_n = \arg\min_k \sum_{n=1} \| x_n - \mu_k \|^2$$

$$= \arg\min_k \| x_n - \mu_k \|^2$$

Thus to minimise the cost with respect to classifications we choose $\mathbf{c}_n$ to be the k for which $||\mathbf{x}_n - \boldsymbol{\mu}_k||^2$ is smallest (i.e. we choose the closest centre).

$J\left(c_1, \ldots, c_m, \mu_1, \ldots, \mu_K\right) =$

"Optimality" of K - means (1)

Cost function $J(c_1, \ldots, c_m, \mu_1, \ldots, m_k) = \sum_{n=1}^{m} \left|\left|x_n - \mu_{c_n}\right|\right|^2$

Step 1: Optimise w.r.t μ_k by setting the derivative wrt it to zero.

$$\nabla_{\mu_k} J(\ldots) = \sum_{x_n \in C_k} -2\mu_k (x_n - \mu_k)$$

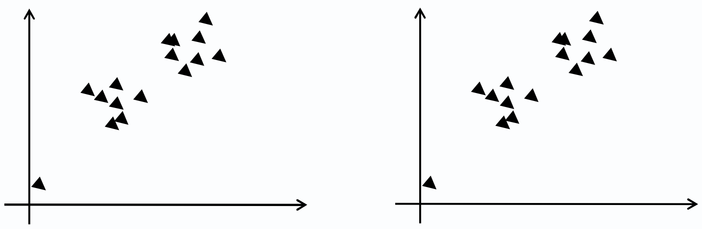
$$\nabla_{\mu_k} J(\ldots) = 0 \Rightarrow \mu_k = \frac{\sum_{x_n \in C_k} x_n}{\sum_{x_n \in C_k} 1} = \frac{1}{|C_k|} \sum_{x_n \in C_k} x_n$$

Where $|C_k|$ are the number of points assigned to cluster k

That is, the μ_k that minimizes the cost function is the average of the points that belong to cluster k.

Local minima

K-means converges only to a local minimum.



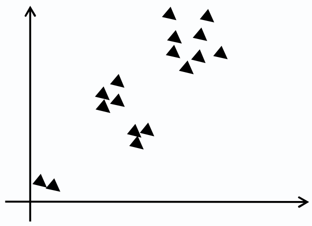
Usually K-means is run several times with different initialisations.

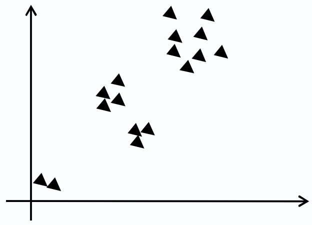
The quality of each solution is determined by evaluating the cost function

The solution leading to the lowest cost is chosen.

How many clusters?

Ambiguous! Not well posed problem.

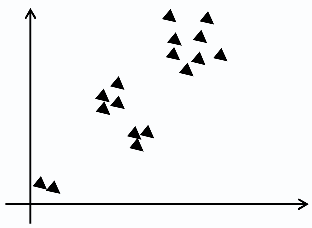


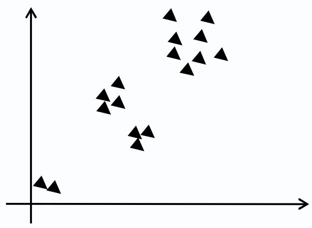


Cost function is monotonic wrt K. Larger K leads to smaller J

How many clusters?

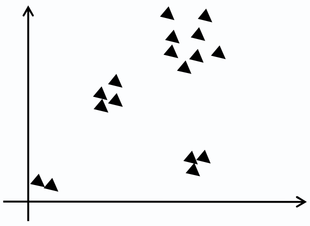
Ambiguous! Not well posed problem.

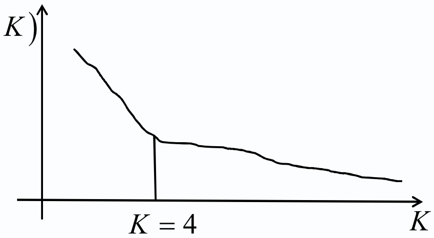




How many clusters?

1. Plot cost as function of K. Cost function is monotonic wrt K. Larger K leads to smaller J. In some cases, the shape may give a hint for the "correct" value of K.





How many clusters?

2. Evaluate on the application. Obtain different clusters for different values of K. Calculate a measure $J'(K)$ that depends on the application

Example a: News clustering:

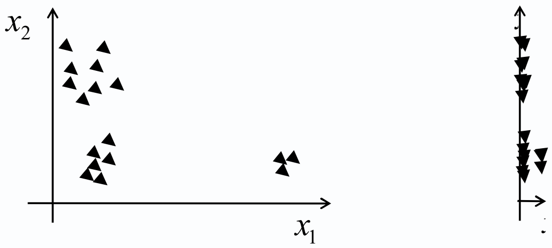
Evaluate: Present to actual users clustering results with different number of clusters. Asses whether they prefer fewer clusters (i.e. Broader topics) or more clusters (i.e. More specialized topics)

Example b: A recommendation system cluster users into groups and gives recommendations according to which cluster it think that a user belongs.

Data scaling

Is sensitive to data scaling.

$x_1 \leftarrow \alpha x_1$ can move the cluster of the points in the right arbitrarily to the left or right.



Normalisation (stretching) $$x_1 \leftarrow \frac{x_1 - min_n\left\{x_1^{(n)}\right\}}{\max_n\left\{x_1^{(n)}\right\} - min_n\left\{x_1^{(n)}\right\}}$$

Normalisation (by variance) $x_1 \leftarrow \sigma^{-1} x_1$

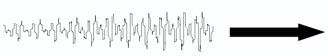
K - means for VQ

MoG interpretation is only one way to view K-means.

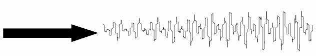
For one particular problem K-means turns out to be optimal:

> ***Problem:*** *What is the best (MMSE) way to quantize vectors of i.i.d. data using N bits per sample?*

This is called Vector Quantization (VQ). It is the basis for state-of-the-art speech codecs.



...001110010101001
1001000100001011
10010010101001010...



Vector Quantization involves two stages:

1. Dividing vector space into $K=2^N$ discrete regions (i.e. classification!)
2. Identifying a reconstruction value for each region

Summary

- Introduced the simple K-means clustering algorithm
  - Limitations
  - Convergence