K-MEANS CLUSTERING

Scheme to divide N data points $\{x_j\}$ into K clusters $\{c_i\}$

$r_{ji} = \begin{cases} 1, & \text{if } x_j \text{ is assigned to } c_i \\ 0, & \text{if } x_j \text{ is assigned to other clusters} \end{cases}$

Distortion Measure (Loss function): $J = \sum_{j=1}^{N} \sum_{i=1}^{K} r_{ji} \| \vec{x}_j - \vec{\mu}_i \|^2$

Find r_{ji} & $\vec{\mu}_i$ which minimize J.

Iterative Process

step 0 - choose $\vec{\mu}_i$ randomly from given data points.

step 1 - $r_{ji} = 1$ for (j, i) such that $\vec{\mu}_i$ is the closest mean vector for given point $\vec{x}_j$
else $r_{ji} = 0$

step 2 - for r_{ji} estimated in step 1, update $\vec{\mu}_i$

$$\nabla_{\mu_i} J = 0 \Rightarrow 2 \sum_{j=1}^{N} r_{ji} (\vec{x}_j - \vec{\mu}_i) = 0$$

$$\Rightarrow \vec{\mu}_i = \frac{\sum_j r_{ji} \vec{x}_j}{\sum_j r_{ji}}$$

Number of points in cluster c_i

Iterate till convergence.
(guaranteed)
could be local minima

Convergence is slow & several variations available.

Lossy Data Compression

For each of the N points, only store the identity of the cluster to which it belongs.
& values of the K cluster centers, $\vec{\mu}_i$
uses significantly less data if $K \ll N$.
also called vector quantization
& $\vec{\mu}_i$ code-book vectors

Example of data compression

Each image has N pixels of {R, G, B}
values stored with 8-bit precision.
So each image needs 24N bits

K-classes needs $\log_2 K$ bits

code-book vectors need 24K bits

∴ Total no. of bits = $24K + N\log_2 K$.

∴ Compression Ratio = $\frac{24K + N\log_2 K}{24N} \times 100\%$

$= \left(\frac{K}{N} + \frac{1}{24}\log_2 K\right) \times 100\%$

$= 13.94\%$ for $K = 10$ & $N = 100 \times 100$

K-Means	KNN
Unsupervised K = no. of clusters into which the given data needs to be classified.	Supervised K = no. of nearest neighbors to use for classification of a new data point

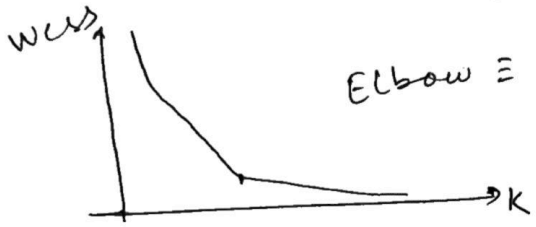
K-Mean & K-NN

K-Mean

- What will you do if number of clusters is not known?

WCSS [within Cluster Sum of Squares]

$$WCSS(K) = \sum_{i=1}^{K} \sum_{j=1}^{N} \left(x_j - x_{j:\mu_i}\right)^2$$



Elbow ≡ Sudden change of behaviour

- Can we do image clustering using this method?
  - Similar images may have different orientation
- How to handle very large dataset?
  - use minibatches.
- How to classify a new data point?

K-NN

small k - more noisy prediction
Large K - more stable prediction

- How to use K-NN for regression?

K-Mean	K-NN
- Unsupervised	- Supervised
- K- need to be determined by elbow method	- $K \sim \sqrt{N}$
- Does not work well with cluster of different sizes	- Size difference is not a problem
- Used for clustering	- Used for classification or regression
- Eager learner	- Lazy learner