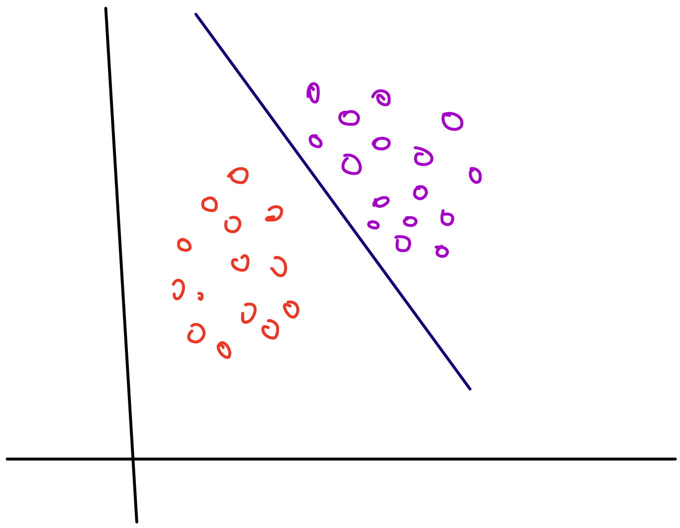
UNSUPERVISED LEARNING

$S = \{x^{(1)}, x^{(2)} \dots x^{(n)}\}$

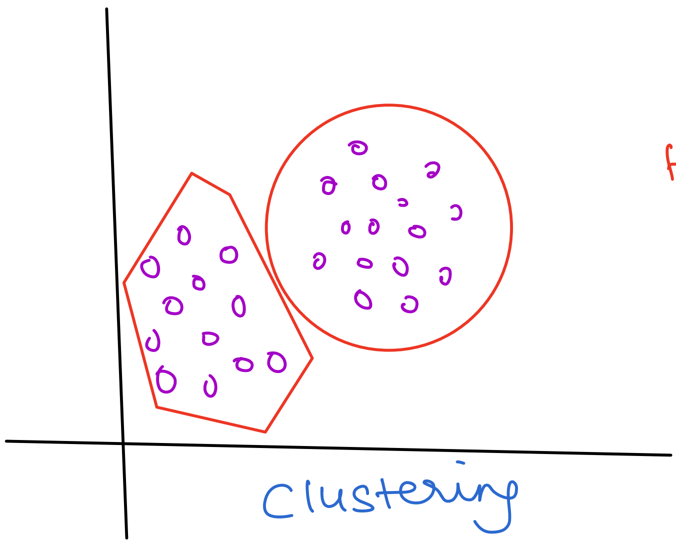
$x^{(i)} \in \mathbb{R}^d$



→ Logistic Regression

We had correct answers for each input for supervised

However in unsupervised setting



Clustering

Here you have to find the interesting structure.

K-Means Algorithm

$S = \{x^{(1)} \dots x^{(n)}\}$ $x^{(i)} \in \mathbb{R}^d$

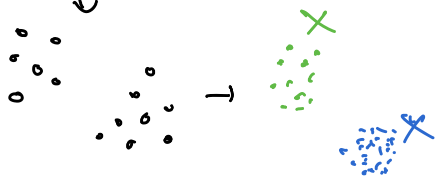
K-clusters

1. initialize cluster centroids
   $\mu_1, \mu_2, \mu_3 \dots \mu_k \in \mathbb{R}^d$ randomly

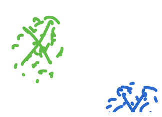
2. Repeat till convergence,
   For every i, set $c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|_2^2$
   For every j, set $\mu_j := \dfrac{\sum_{i=1}^{n} 1\{c^{(i)} = j\} \cdot x^{(i)}}{\sum_{i=1}^{n} 1\{c^{(i)} = j\}}$

→ Initialize centroids randomly
→ c is array of length n
→ set to identity of nearest means → arg min
→ μ_j means of all $x^{(i)}$'s for which $c^{(i)} = j$

can be thought as labelling clusters.



After labelling we find μ_i



and readjust it such that it is at center of clusters.

$$J(C, \mu) = \sum_{i=1}^{n} \|x^{(i)} - \mu_{c^{(i)}}\|_2^2$$

↓

Distortion Function

We use Co-ordinate Descent. It is a variant of gradient descent wherein we minimize loss wrt few variables holding others constant
One step optimize wrt C
Second step optimize wrt μ

$J \to$ non-convex, end up with different solns depending on initialization but we always get a soln i.e. it converges.

(Q) Why use labelled identities, when we have K-Means?

→ With K Means, you end up with diff solns i.e. different cluster identities depending on initialization

* K-value has to be chosen by us based on domain knowledge, we can use Elbow method

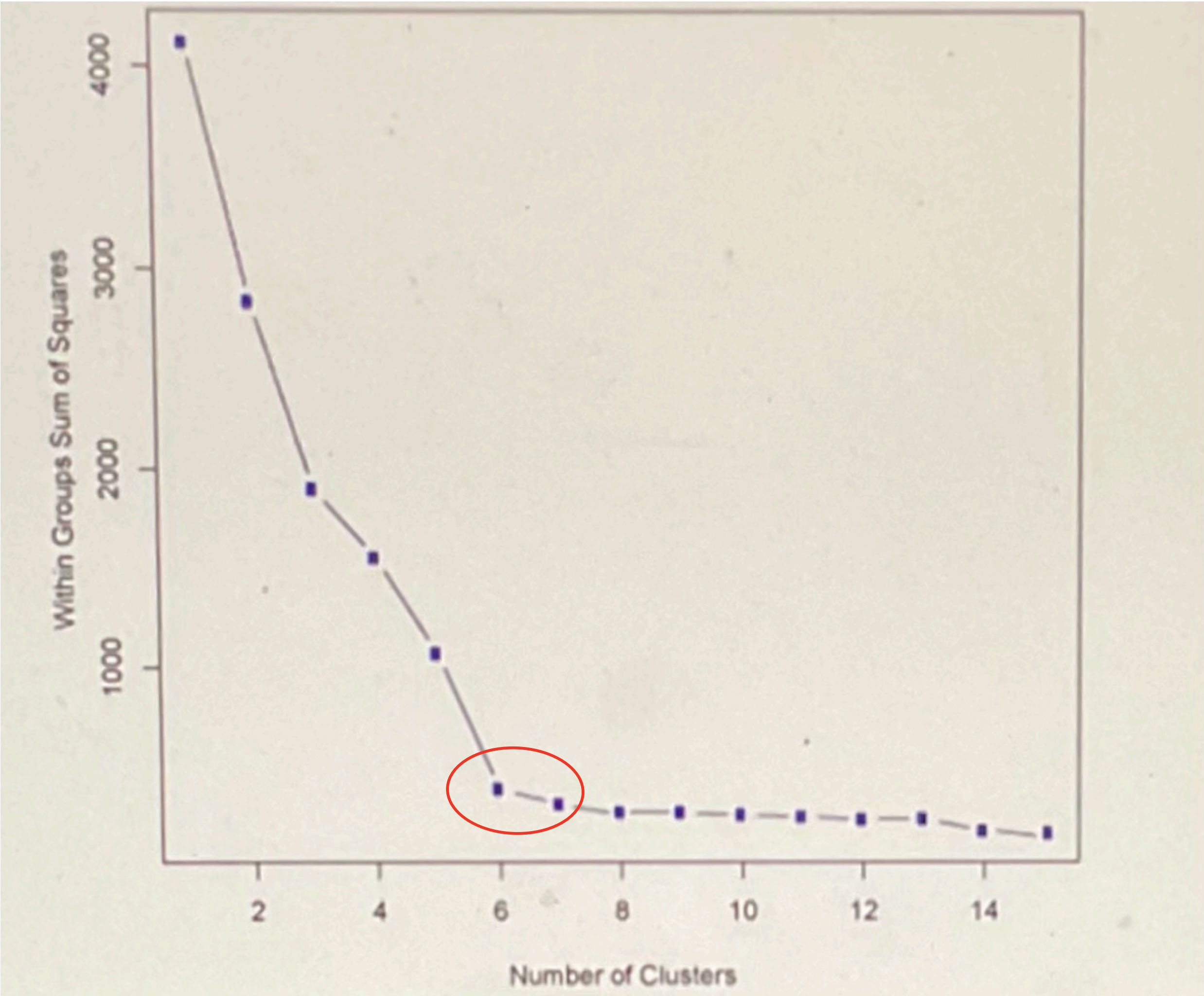
Elbow Method: Choose K at which SSE decreases abruptly.

We compute SSE i.e. sum of squared distance between each member of cluster and its centroid.

Do it for K=2, 4, 6....

If we plot K against SSE, you will see error ↓ as K gets larger as cluster increase in number & ↓ in size.

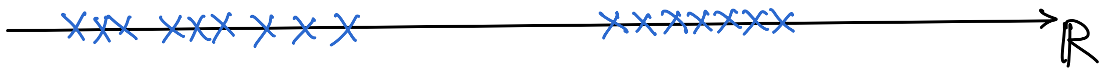
Choose a K at which SSE decreases abruptly



Choose k, where on ↑ k, you don't minimize within group S.S.E.

In this example, elbow occurs around 6-7

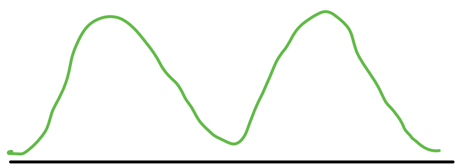
Density Estimation



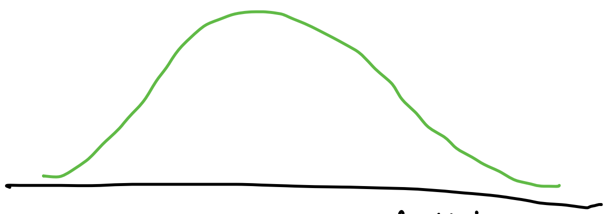
These points are sampled from a probability distribution

Since on $\mathbb{R}$ line, it is sampled from P.D.F.

We have to estimate the PDF (To fully fit we would have to use dirac delta)



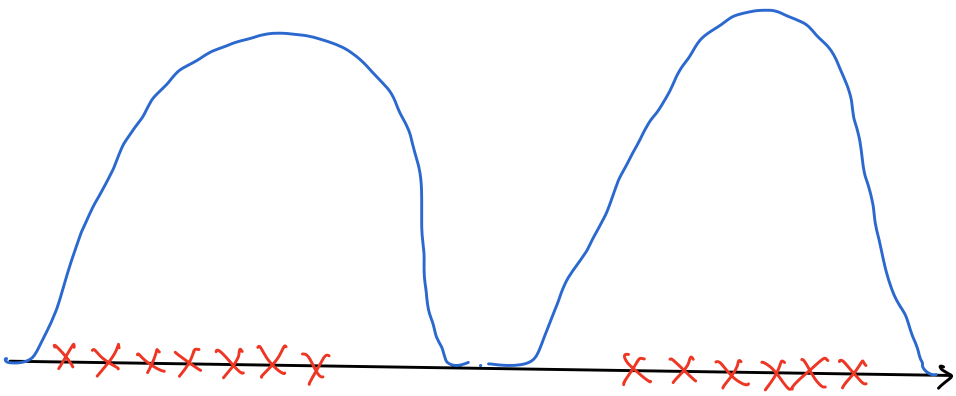
-OR-



. . . Many possibilities

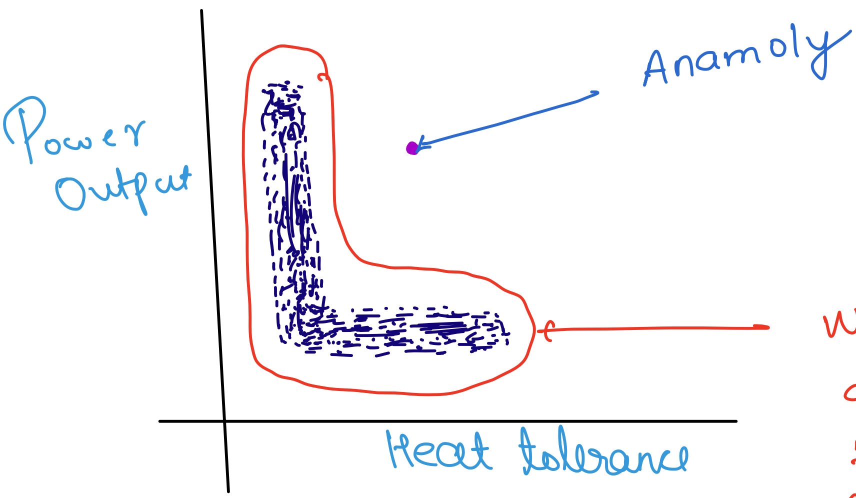
We have to estimate a smooth curve over a fixed number of observations.

Gaussian Mixture Models



We assume above points sampled from 2 Gaussians
This is in unsupervised setting.
In Supervised setting, we had G.D.A. Here, we don't have the y labels and covariance can be different as oppososed to GDA where it was same

Application: Anamoly Detection



we have p(x) which assigns high probability to anything in this region. Anything outside is ≈ 0

Choice of K is once again on user

GMM → Soft K-Means

$S = \{x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)} \dots x^{(n)}\}$

$z^{(i)} \sim \text{Multinomial}(\phi)$ ← Class identity

$\phi_j \geq 0 \quad \sum_{j=1}^{k} \phi_j = 1$ → Class prior tells how many examples of cluster j

which one of K

$\phi_j = p(z^{(i)} = j)$

$x^{(i)} | z^{(i)} = j \sim N(\mu_j, \Sigma_j)$

$z^{(i)}$ similar to $y^{(i)}$ in G.D.A

- latent variable as it is not observed

$$\log p(x; \phi, \mu, \Sigma) = \ell(\mu, \Sigma, \phi)$$

$$= \log \sum_{z} p(x, z; \phi, \mu, \Sigma)$$

$p(z)$ → Class prior

$p(X, Z)$ → model

$p(Z|X)$ → posterior

$p(X)$ → evidence

Z → latent variable

For each point assign a weight to cluster centroid i.e. posterior distribution of $p(z)$

[Inspired by K-Means]

Randomly initialize μ, ϕ, Σ

Repeat until convergence

E-Step : For each i, j set:

$$w_j^{(i)} = p(z^{(i)} = j \mid x^{(i)}; \phi, \mu, \Sigma)$$

Use Bayes Rule

M-Step: Update Parameters

$$\phi_j = \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{n} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{n} w_j^{(i)}}$$

every x contributes

$$\Sigma_j = \frac{\sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{n} w_j^{(i)}}$$

eg.

$K=3$

$p(z^{(i)}=j \mid x^{(i)})$

$$\begin{array}{cc} \begin{bmatrix} 0.1 \\ 0.7 \\ 0.2 \end{bmatrix} & \begin{array}{l} k=1 \\ k=2 \\ k=3 \end{array} \end{array} \qquad \overbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}^{\text{K-Means}}$$

Here we do a soft assignment instead of K-means where there is hard assignment.

$$p(z|x) = \frac{\overbrace{p(x|z)}^{\text{Gaussian}} \; \overbrace{p(z)}^{\text{Multinomial}}}{\sum_z p(x|z)\,p(z)}$$

↳ used in E-step

[You get softmax with quadratic features]

GDA → if covariances same, you get logistic
if covariances diff, you get curved.

Here diff covariances, so curved, hence softmax with quadratic features