WEEK # 8
Unsupervised Learning

1. CLUSTERING

1.1. Unsupervised Learning : Introduction:

Algorithm finds some structure in the data for us.

ex/ market segmentatn
social network analysis
organize computing clusters
Astronomical data analysis

1.2. K-means Algorithm:

In clustering problem we are given an unlabeled dataset.

K-means algorithm:
Input: - K (# of clusters)
- Training set

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots \mu_K$
Repeat {
for $i=1$ to m $\qquad c^{(i)} = \min_k \|x^{(i)} - \mu_k\|^2$
$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$
for $k=1$ to K
μ_k := average(mean) of pnts assigned to cluster
}

1.3. Optimizatn Objective:

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

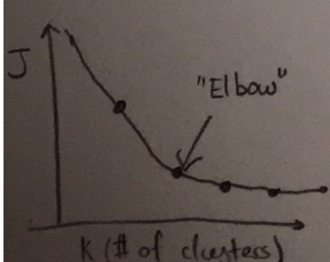
distortn cost fxn.

1.4. Random Initializatn:

Depending on the random initializatn K-means can end up at different solutns.
Initialize K-means lots of time, ⇒ Pick clustering that gave lowest cost $J(c^{(1)}, \ldots c^{(m)}, \mu_1, \ldots \mu_K)$

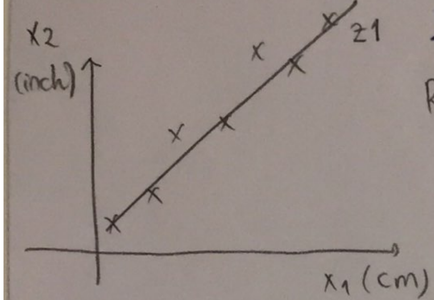
valid for $K = 2-10$; if $K \geq 100$ the 1st initializatn gives similar (optimal) solutn.

1.5. Choosing # of Clusters:

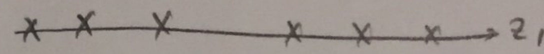


2.) MOTIVATION.

2.1. Data Compression:



Reduce data from 2D to 1D.



2.2. Visualizatn:

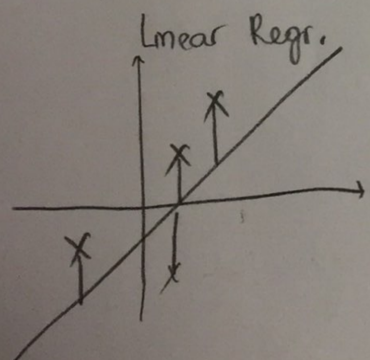
3.) PCA - Principal Component Analysis.

3.1. PCA Problem formulatn:

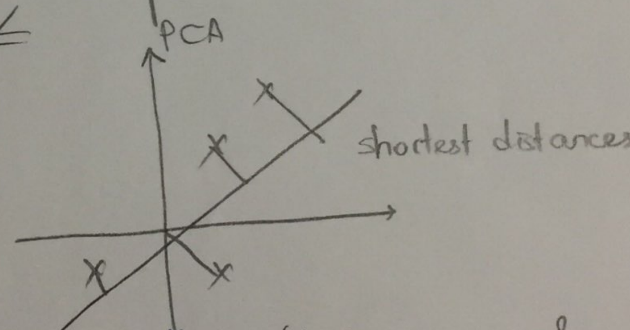
PCA tries to find lower dimensional surface onto which to project data so that sum of squares of distances btw the pnts and surface are minimized. (projectn error).

→ mean normalizatn feature scaling.
→ Apply PCA

project onto linear subspace spanned by this set of k vectors ($u^{(1)}, u^{(2)} \dots u^{(k)}$)



$\neq$



3.2. PCA Algorithm: (find $u_1, u_2 \dots$ & $z_1, z_2 \dots$)

Data preprocessing (feature scaling / mean normalizatn).

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)} \longrightarrow x_j^{(i)} \longleftarrow \frac{x_j^{(i)} - \mu_j}{s_j}$$

Reduce data from n-dimensions to k-dimensions:

1) Compute Covariance mtx. $\Sigma = \frac{1}{m} \sum_{i=1}^{n} (x^{(i)}) \cdot (x^{(i)})^T$

2) Compute "Eigenvectors" of mtx Σ. [U,S,V] = svd (Sigma);
Ureduce = U(:, 1:k);
z = Ureduce' * x;

4. APPLYING PCA.

4.1. Reconstructn from Compressed Representatn:

$$\underbrace{X_{approx}}_{n\times 1} = \underbrace{U_{reduce}}_{n\times k} \cdot \underbrace{z}_{k\times 1}$$

4.2. Choosing the # of Principal Components:

Average squared projectn error: $\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} - X_{approx}^{(i)} \|^2 \longrightarrow A$

Total variatn in the data: $\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} \|^2 \longrightarrow B$

$\frac{A}{B} \leq 0.01$ (1%) "99% of variance is retained".

$\phantom{\frac{A}{B}}$ 0.05 (5%) "95% " " "

Try PCA with $k=1$

Compute $U_{reduce}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x^{(1)}_{appr}, \dots, x^{(m)}_{appr}$

Check if $\frac{A}{B} \leq 0.01$

else

$[U,S,V] = svd(Sigma)$

$$S = \begin{bmatrix} S_{11} & & & 0 \\ & S_{22} & & \\ & & \ddots & \\ 0 & & & S_{nn} \end{bmatrix}$$

For given k: $\boxed{1 - \frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \leq 0.01}$

iterate until satisfies.

4.3. Advice for Applying PCA:

PCA is used to speed up Supervised Learning.

$x^{(1)}, x^{(2)}, \dots, x^{(m)} \in \mathbb{R}^{10000}$

$\downarrow$ PCA

$z^{(1)}, z^{(2)}, \dots, z^{(m)} \in \mathbb{R}^{1000}$

Mapping $x^{(i)} \rightarrow z^{(i)}$ should be defined by running PCA only on the training data. This mapping can be applied to $x_{cv}^{(i)}$ & $x_{test}^{(i)}$.

Applicatn of PCA:

- Compression: 1) Reduce memory/disk needed to store data
  2) Speed up learning algorithm
- Visualizatn: