Week 08

- Unsupervised Learning
    - Clustering
    - Dimensionality Reduction

I. Clustering

1. Unsupervised learning

- Clustering

2. K-Means Algorithm

Init clustering with 2 points or clutser centroids to group data around them.

Algorithm:

- Input:
    - K number of clusters
    - Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}; x_0 = 1$
- Init K cluster centroids $\mu_1, \mu_2, \ldots \mu_K$
- Repeat : {
    - for i = 1 to m
        - $c^{(i)} := index(\text{ from 1 to K }) \text{ of centroid closest to } x^{(i)} \Rightarrow min_k ||x^{(i)} - \mu_k||$
    - for k: 1 to K
        - $\mu_k := \text{average of points assigned to cluster k}$
- }

3. Optimization

First loop is the optimization objective $J(..)$

$c^{(i)}$ = index of cluster (1, .. K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

$J(c^{(1)}, .., c^{(m)}, \mu_1, .., \mu_k) = 1/m \sum_{i=1}^{m} ||x^{(i)} - u_{c^{(i)}}||^2 = min_{c^{(1)},..,c^{(m)},\mu_1,..,\mu_k} J(c^{(1)}, .., c^{(m)}, \mu_1, .., \mu_k)$

4. Random initialization

$K < m$ as K: number of clusters ; m: number of training examples

Pick random K training examples

Set $\mu_1, ..; \mu_K$ equal to these examples

Cluster and re cluster the clustered clusters to find local optima

Compute: cost function (distortion) $J(c^{(1)}, c^{(2)}, .., \mu_1, \mu_K)$

Pick clustering that gave lowest cost J

5. Choosing the Number of Clusters

- Rule for the right K
  - Elbow method: Get a chart of J |__ K
  - If the line chart resembles an arm __, then the "elbow" (the point of inflection on the curve) is a good indication that the underlying model fits best at that point.
- Can choose by hand and see

II. Motivation

1. Data compression

- Dimensionality reduction = Data compression
  - Reduce data from 2D to 1D
    - $x1, x2$ -> $z1 =$ single line with projection of points.
  - Reduce from 3D to 2D
    - $x1, x2, x3$ -> $z1, z2$

2. Visualization

Dimensionality reduction to plot and visualize data

III. Principal component analysis: PCA

1. Problem formulation

- Pass from 2D to 1D
- Find a direction (vector) to project data on to minimize projection error

  =>
- Pass from nD to kD
- Find k directions (vectors $u^{(1)}, ... u^{(k)}$) to project data on to minimize projection error

PCA is not linear regression: in linear regression, we plot lines vertically on the fitting line. In PCA, we plot lines perpendicularly on the fitting line.

2. Algorithm

- training set $x^{(1)} ... x^{(m)}$
- Data preprocessing
  - Feature scaling// normalization

    $\mu_j = 1/m \sum_{i=1}^{m} x_j^{(i)}$

replace each $x_j^{(i)}$ with $x_j - \mu_j$

- PCA:
  - 3Dto2D;
    - $u1$ and $u2$
    - z = $z1$ and $z2$
  - 2Dto1D
    - $u1$ as a vector
    - $z1$ straight line

  Compute eigenvectors

```
[u,s,v] = svd(sigma)
```

3. Applying PCA