Clustering / Unsupervised learning

training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(m)}\}$

K-Means Algorithm

Cluster centroids (2 random points for 2 clusters)

Cluster assignment, mean and move

Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

$x^{(i)} \in \mathbb{R}^n$ (drop $x_0 = 1$ convention)

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

cluster assign: for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

move centroid: for $k = 1$ to K

μ_k := average (mean) of points assigned to cluster k

$\min_k \|x^{(i)} - \mu_k\|^2 \rightarrow c^{(i)}$

eliminate a cluster centroid with no points (can just randomly assign 1)

}

K-means for non-separated clusters

Optimization Objective (Cost Fcn)

$c^{(i)}$ = index of cluster $(1, 2, \dots, K)$ to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)

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

Optimization objective:

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

$$\min_{\substack{c^{(1)}, \dots, c^{(m)} \\ \mu_1, \dots, \mu_K}} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$$

Distortion Cost Function

cluster assignment = minimize $J(\dots)$ wrt $c^{(1)}, c^{(2)}, \dots, c^{(m)}$ ←

(holding $\mu_1, \dots, \mu_K$ fixed)

move centroid = minimize $J(\dots)$ wrt $\mu_1, \dots, \mu_K$

Random Initialization

$K < m$

Local optima

$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$ try multiple rand initializations

For $i = 1$ to 100 {

Randomly initialize K-means

Run k-means get $c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K$

Compute cost function (Distortion)

$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

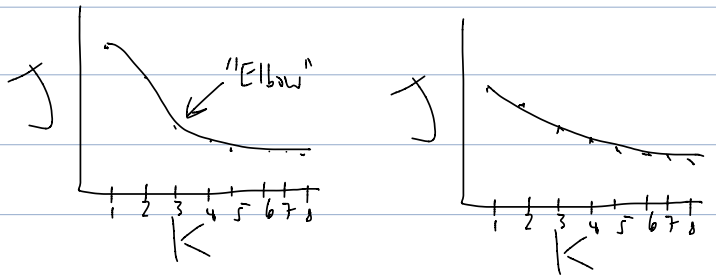
}

Pick clustering that gave lower cost fcn

$K = 2$-6 ← worth it ≃

Choosing the Number of Clusters

Elbow method: worth a shot



metric-based (future purpose)

Dimensionality Reduction

Data Compression

$X1) \rightarrow (X-1)1)$

Visualization

$K=2$ to 3 $\quad K \leq N$

Principal Component Analysis Problem Formulation

Reduce from 2D to 1D: Find a direction (vector $u^{(1)} \in \mathbb{R}^n$) onto which to

project the data so as to minimize the projection error

Reduce from N-D to K-D: Find k vectors $u^{(1)}, \dots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Find lower dimension w/ proj. to minimize the closest projections

---

PCA Algorithm

Data preprocessing

Training set: $x^{(1)}, \dots, x^{(m)}$

Feature scaling/mean normalization/preprocessing:

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

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

If different features on different scales (size of house, # bedrooms) scale features to have comparable range of values

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{s_j}$$

Reduce data from N-D to K-D

Compute "Covariance matrix"

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} (x^{(i)})(x^{(i)})^T$$

Compute "eigenvectors" of matrix Σ:

$$[U, S, V] = \text{svd}(\text{Sigma});$$

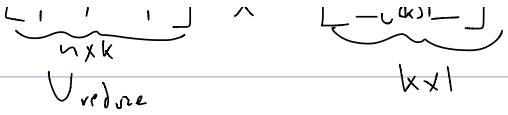
$\Sigma \in \mathbb{R}^{n \times n}$ ← nxn matrix

$$U = \begin{bmatrix} | & | & | & & | \\ u^{(1)} & u^{(2)} & u^{(3)} & \cdots & u^{(n)} \\ | & | & | & & | \end{bmatrix}$$

$u^{(1)} \cdots u^{(k)}$ ← k

$$z = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \cdots & u^{(k)} \\ | & | & & | \end{bmatrix}^T x = \begin{bmatrix} — u^{(1)T} — \\ \vdots \end{bmatrix} x$$

k x n ; n x 1



ENSURE EVERY FEATURE HAS ZERO MEAN

Reconstruction from Compressed Representation

$z = U_{reduce}^T x$

$z \in \mathbb{R} \rightarrow x \in \mathbb{R}^2$

$x_{approx}^{(n)} = U_{reduce} \cdot z$

$x_{approx} \in \mathbb{R}^n$; U_{reduce}: nxk ; z: kx1 ; result: nx1

Choosing the Number of Principal Components

Average squared projection error: $\frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - x_{approx}^{(i)}\|^2$

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

Typically, choose k to be smallest value so that

$$\frac{\frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m}\sum_{i=1}^{m} \|x^{(i)}\|^2} \leq 0.01 \quad (1\%)$$

0.05

"99% of variance is retained"

99-95 common

Algorithm:

Try PCA with k=1

Compute $U_{reduce}, z^{(1)}, \ldots, z^{(m)}, x_{approx}^{(1)}, \ldots, x_{approx}^{(m)}$

Check if

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}-x_{approx}^{(i)}\|^2}{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}\|^2} \leq 0.01 \ ?$$

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

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

For given k

$$1 - \frac{\sum_{i=1}^{k} s_{ii}}{\sum_{i=1}^{n} s_{ii}} \leftarrow \text{CV} \leq 0.01 \qquad \leftarrow \text{ALL}$$

Pick smallest value of k for which

$$\frac{\sum_{i=1}^{k} s_{ii}}{\sum_{i=1}^{m} s_{ii}} \geq 0.99 \quad (99\% \text{ variance retained})$$

Advice for applying PCA

Ex: Supervised learning speedup

$(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)}) \qquad x^{(i)} \in \mathbb{R}^{10,000}$ (100 × 100 image)

Extract inputs:

Unlabeled: $x^{(1)}, \ldots, x^{(m)} \in \mathbb{R}^{10000}$ new x

↓ PCA (mapping $x \to z$) (defines)

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

New training set:

$(z^{(1)}, y^{(1)}), \ldots (z^{(m)}, y^{(m)}) \qquad h_\theta(z) = \frac{1}{1+e^{-\theta^T z}}$

Note: should be done on training set. Could apply to CV or test

for defining mapping $x \to z$ mapping

Bad use of PCA: to prevent overfitting BAD

Might work ok, but not good way to address. Use regularization