size, $K=3$ Small, medium, large

t shirt example : m users, 2 features — height, weight

Page No. Date

features = height, weight = 2

K = no. of cluster

K mean algo

$$X = \begin{array}{c|c} & \text{height} \quad \text{weight} \rightarrow n \\ \begin{array}{c} x_1 \text{ user} \\ x_2 \text{ user} \\ \vdots \\ x^m \text{ user} \end{array} & \begin{bmatrix} - & - \\ - & - \\ - & - \end{bmatrix} \end{array}$$

$$\underset{R^n}{\mu} = \begin{array}{c|c} & \text{height} \quad \text{weight} \\ \begin{array}{c} \mu^1 \\ \vdots \\ \mu^k \end{array} & \begin{bmatrix} - & - \\ - & - \end{bmatrix} \end{array}$$

$C =$ c^1, c^2 ------ c^m [index of cluster]

$1 \le c^i \le K$

cluster of x_1, cluster of x_2, cluster of x_m.

Array with $c_1, c_2, \ldots, c_m$ | idx

① for ($i = 1$ to m)

c^i = index (from 1 to k) of cluster centroid closest to x^i

Calculated by $\|x^i - \mu^j\|$ $1 \le j \le K$ which give min will store in c^i.

② for ($k = 1$ to K)

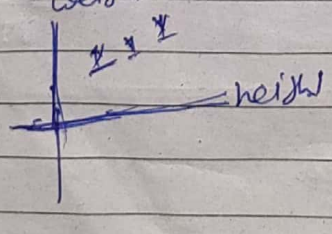
$\mu_k = \text{mean}(\quad)$

$k=1$ for cluster 1   or $c^i == 1 \; \forall i$

$A = (x^i)$ $(A; x^i)$

end.

mean(A) it give columnwise minimum.



~~If any cluster has no~~ points then either remove the cluster or randomly assign that cluster centroid.

Since c_i is index of cluster, if we want to calculate centroid i.e, μ_{c^i}

Optimization

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

for (i = 1 to 100) — is done b/w 50 & 1000 to get better soln.

K means {

randomly initialise $\mu_1 - \mu_K$

Calculate

Repeat {

(1) For (i = 1 to m)

(1(a)) $c^i =$ $k = 1$ to k

(2) $\mu_k =$ mean (—)

}

}

If K is b/w 2 & 10 doing random initialization b/w 50 & 1000 give better soln.

K means algorithm is a method to automatically cluster similar data example together.

$v = [\quad]_{1 \times -}$

$v(2:5) =$ It will give 2 to 5 elements

$c_i =$ idx $= [\quad]$

LOKENATH
Page No.
Date

Hell weight
~~feat~~ feature

$$idx = C^i = \begin{array}{c} c_1^i \\ c_2^i \\ \vdots \\ c_m^i \end{array} \begin{bmatrix} 1 \\ 2 \\ 1 \\ 3 \end{bmatrix} \xrightarrow{i}$$ nearest centroid of user.

user (from 1 to m) which has centroid = i

$c\_i = \cancel{0}$ $\quad$ $idx == i$ | if we are checking i == 1

$$c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

{idx == 1 will be return '1' other '0'}

H_c centroid

c_i_matrix = repmat (c_i, 1, n) ;

It will concatenate matrix c_i n times along columnwise. & 1 time (it 2 or 3 or m) then n time along row

~~(c_i, 2, n)~~

X .* c_i_matrix

will give H_{c^i} for c^i

centroid = (i, j)

with Gaussian Model.

- n is small, m is large -
create/add more features, then use logistic regression or SVM without a kernel.

Kmeans does two things: cluster assignment (putting data in cluster), move centroid

myf(n-on) ky(r-hon)

week 8

In unsupervised learning you are given unlabeled

K-means algorithm:

x is of n dimension or n feature

Input → K (no. of cluster)
→ training set ($x^1, x^2, \ldots x^m$)

$$x = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_2^1 & x_2^2 & x_2^3 \\ \cdot & & \end{bmatrix}$$

$x^i \in R^n$ (x^i belongs to R^n dimensional vector)

we have $x^1 \ldots x^m$ dataset, we divide it into K set or cluster. ($\mu_1 \ldots \mu_K$)

find centroid for every set also mean for every set.

then in c^i store centroid which is closest to x^i

$$C = \begin{array}{ccccc} 1 & 2 & \ldots & & m \\ \hline c_{\mu_1}^{x^1} & c_{\mu_2}^{x^2} & c_{\mu_2}^{x^3} & \ldots & \end{array}$$

$1 \times m$

μ_k is cluster centroid.

supervised
→ linear & logistic regression
having optimization.

unsupervised

k means algorithm

for (i = 1 to m)

c^i = index (from 1 to k) of cluster centroid closest to x^i.

at the value k

If any x^i is nearest to centroid k then it will be in cluster k

for (k = 1 to k)

μ_k = average of points assigned to cluster k.

is c1

For doing mean of cluster 1 or μ_1.

K means optimization:

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

cost function

element x^i

centroid of x_i cluster

Random initialisation

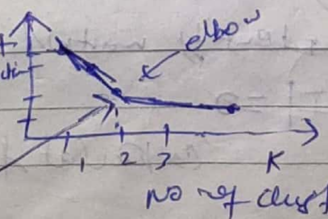
Since in starting we only know no. of cluster (i.e k) so, 1st pick k random element and initialize it as centroid of k cluster.

If we small k (no. of cluster) nearly less than 10 then random initialisation (50–1000 time) can give better sol^n.

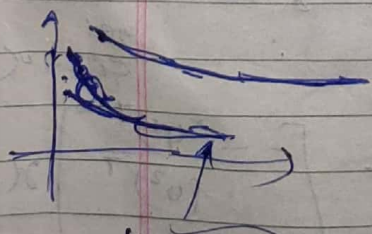
mean(x) → give column wise

choosing No of cluster: (value of k)

- Elbow method:



~~Not sure~~ Not always sure it work for a problem



pick that no of cluster where elbow is forming

Help

elbow not work good.