Regression means ~~Regradation~~ or moving towards mean.
It

LOKENATH
Page No.
Date

~~Supervised~~
labeled data for which it lea

→ Linear regression

- It models the relationship b/w a single dependent variable (y) & one or more independent (x) variable
- ~~→ logistic regression~~
- ~~It produces a continuous valued output.~~
- It passes through mea-

Logistic:
classification algorithm or where y value is discrete.

Binary output either 0/1 yes/no

Linear regression:

Single variable (1 feature)

⇒ $h_\theta(x) = \theta_0 + \theta_1 x$.
hypothetical value (guessed from algorithm)
actual value y^i ~~from a~~ training set (x^i, y^i)

- $h_\theta(x)$ should be such that it more close y^i.

⇒ $J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\left(h_\theta x^{(i)}\right)-y^i\right)^2$
cost function

gradient descent: {repeat until convergence}

⇒ $\theta_j : \theta_j - \alpha \cdot \frac{\partial}{\partial\theta_j} J(\theta_0,\theta_1)$

$\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial\theta_0} J(\theta_0,\theta_1)$

$\rightarrow \theta_0 - \alpha \cdot \frac{1}{m}\sum_{i=1}^{m}(h_\theta x^i - y^i)$

$\theta_1 = \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta x^i - y^i)\cdot x^i$

~~unsupervised~~
~~cluster (group)~~

$h_\theta(x) = \theta_0 -$

$J(\theta_0 .. \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta x^i - y^i)^2$

$x_0 \quad x_1 \quad x_2 \ldots x_n$

Input →

Normal eqn formula for θ
$$(X^TX)^{-1}X^Ty$$
X is feature matrix of $m\times(n+1)$

Gradient descent

$\theta_j = \theta_j - \alpha \cdot \frac{1}{m}\sum_{i=1}^{m}(h_\theta x^i - y^i)x_j^i$

$\theta_j = \theta_j - \alpha \frac{\partial}{\partial\theta_j}$ cost function

multivariable (having n feature)

No of row (i) denotes no. of input & j denote feature in column.

⇒ $h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$

$x_1, x_2, \ldots x_n$ → features

x_j^i → i^{th} row & j^{th} column which is feature.

$\theta = [\theta_0 \ \theta_1 \ldots \theta_n]$

$X = \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix}$

let $x_0 = 1$

$h_\theta(x) = \theta^T x$

(2) pinv → fun to calc inverse in octave

Normal eqn for θ i.e $\theta = (X^T X)^{-1} X^T y$.

Can be use upto 1000 or 10,000 features.

[coz it is slow of ($O(n^3)$) order

Gradie	Nor
(1) choose α	→ x
(2) iterate till get $J(\theta)$ min.	→ x
$O(n^2)$	$O(n^3)$

when X is non invertible matrix

⇒ P.C.A mainly used for visualisation (so we compress dimension to 2D or 3D)

week 9 Movie recommendation :

1) n_m = no. of movies, n_u = no. of user, $r^{(i,j)} = 1$ if user j rated movies i
(i) denote, (j) denote

$y^{(i,j)} \rightarrow$ if ($r^{ij} = 1$) then y^{ij} = rating value

check by ($r .* M$)

$$Y^{ij} = \begin{array}{c|c} & user_1 \; user_2 \; \cdots \; user_{n_u} \\ \hline \text{movies } m_1, m_2, \ldots, m_m & \begin{bmatrix} \; \end{bmatrix}_{n_m \times n_u} \end{array}$$

rating matrix

$$r^{ij} = \begin{array}{c|c} & user \\ \hline movie & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{n_m \times n_u} \end{array}$$

parameter vector

$$\Theta = \begin{array}{c|c} & user_1 \; \cdots \; user_{n_u} \\ \hline movie & \begin{bmatrix} \theta_0 & \theta_0 \\ \theta_1 & \theta_1 \\ \theta_2 & \vdots \\ \vdots & \\ \theta_n & \theta_n \end{bmatrix}_{n \times n_u} \end{array}$$

$x^i \rightarrow$ ~~feature vector for~~ movie i

$$X = \begin{array}{c|c} & \\ 1^{st}\text{ movie} & \begin{bmatrix} x_0 & x_1 & x_2 & \cdots & x_n \\ \vdots & & & & \end{bmatrix}_{n_m \times n} \end{array}$$

feature matrix

$\theta^j \rightarrow$ parameter ve

(2) calculating cost function having X (feature) & minimizing

$$\min_{\theta^{(1)} \ldots \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \underbrace{\sum_{i:r(i,j)=1}}_{\text{all the movies rated by user } j} \left((\theta^j)^T x^i - y^{ij}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^j)^2$$

$$\theta_k^j = \theta_k^j - \alpha\left(\sum_{i:r(i,j)=1} \left((\theta^j)^T x^i - y^{ij}\right) x_k^i + \lambda \theta_k^j\right)$$

(3) calculating feature forced having prior (θ)

check from pdf

(4) Adding both (2) & (3) called Collaborative filtering

cost
$\Theta_j \rightarrow$
$x \rightarrow$

(5) mean normalisation (done because many user not rated any movie or given every movie 0)

recommend movie j related to i such that $\| x^i - x^j \|$ minimum

(1)

j = user
i = movie

n_m = no. of movies.

n_u = no of user

$r(i,j) \rightarrow$ if user j has rated movie i then $r^{ij} = 1$

$y(i,j) \;=\;$ if $(r^{i,j}) = 1$ then (y^{ij}) = rating value

movies	user$_1$	user$_2$	feature$_1$ romance	feature$_2$ action	feature$_3$
movie$_1$		$i=1$			
movie$_2$		$i=2$			
		$i=3$			

$\rightarrow$ feature matrix

$$\begin{array}{cccc} & x_0 & x_1 & x_2 \\ & \left[\begin{matrix} 1 & \cdot & - \\ 1 & & \\ 1 & & \\ 1 & & - \\ 1 & & \end{matrix}\right] \end{array} \qquad \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \leftarrow \text{Parameter matrix}$$

$\theta_0 + \theta_1 x - + \theta_2 x - = 5$

$\Rightarrow$ 1 user give n_m rating. and one movie has $(n+1)$ feature

$\Rightarrow$ x^i $\Rightarrow$ feature vector for movie i.
(selecting i^{th} row in the feature matrix)

$\Rightarrow$ θ^j $\Rightarrow$ Parameter vector for user j.

movies$_1$

user1	user2		x_0	x_1	x_2
		$m_1 \rightarrow$			
		$m_2 \rightarrow$	5	6	7
		$m_3 \rightarrow$			
		$m_m \rightarrow$			

Every user has its parameter vector

so.

user1 $\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$ user2 $\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$

for user j, movie i

predicted rating : $(\theta^j)^T x^i$

m^j = no. of movies rated by user j.

θ_0
$\theta_j \rightarrow$
$\theta_{n_u} \rightarrow$

$u(j) \rightarrow \theta_j =$ n_u (no. of user) $\times (n+1)$

$$\begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_n \\ \theta_0 & \theta & - & \theta \\ & & & \end{bmatrix}$$

θ_1 θ^1 $\Rightarrow$ 1^{st} row i.e parameter for user 1.

$$\begin{array}{cc} & 1 \; 2 \; 3 \; \cdots \; n_m \rightarrow \text{no of movies} \\ \begin{matrix} 1 \\ \text{no. of user} \\ n_u \end{matrix} & \begin{bmatrix} \\ 0 \; 5 \; 4 \; ? \; ? \; 2 \; 3 \\ \\ \end{bmatrix}_{n_u \times n_m} \end{array}$$

$\leftarrow$ rating matrix (y^{ij})

(2)

$h_\theta(x) \Rightarrow$ for user rating $\Rightarrow$ minim

$\left(h_\theta(x) - (\text{actual rating})\right)^2$

So, we have to minimize,

$$\Rightarrow \min_{(\theta^j)} \frac{1}{2m^{(j)}} \sum_{i:(r^{ij}=1)} \left((\theta^j)^T x^i - y^{ij}\right)^2 + \frac{\lambda}{2m^{j}} \sum_{k=1}^{n} (\theta_k^j)^2$$

$j \to$ for user j^{th}

For single user (j)

$$\min \theta_j = \frac{1}{2m} \sum_{i:r(i,j)=1} \left((\theta^j)^T x^i - y^{ij}\right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} (\theta_k^j)^2$$

~~for all user~~ For all user (calc θ matrix)

$$\min_{\theta^{(1)} \dots \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \underbrace{\sum_{i:r(i,j)=1}}_{\{\text{all the movie rated by user } j\}} \left((\theta^j)^T x^i - y^{ij}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^j)^2$$

Here k is k^{th} param/feature

Gradient descent:

parameter vector for j^{th} user

$$\theta_k^j = \theta_k^j - \alpha \left(\sum_{i:r(i,j)=1} \left((\theta^j)^T x^{(i)} - y^{(i,j)}\right) x_k^i + \lambda \theta_k^{(j)} \right)$$

(movie feature matrix for i^{th} row of movie)

$$p_i = \sum_{i=1}^{n} \text{Sum}(y^i)/\text{Sm}$$

③ Since most of the time feature value is not provided so, we will take feature from user & minimizing its cost.

$$\min_{x_i} \frac{1}{2} \sum_{j:r(i,j)=1} \left((\theta^j)^T x^{(i)} - y^{ij}\right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \underbrace{(x_k^i)^2}_{\text{gn } \theta_k^j \text{ gn parameter}}$$

For n_m movies

$$\min_{x^{(1)} \dots x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left((\theta^j)^T x^i - y^{i,j}\right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} (x_k^i)^2$$

$j:r(i,j)=1$ → all the user that has rated movie i

$\theta \to x \to \theta \to x$ - - - converge

more efficient : (Collaborative filtering)

combine both minimizing (feature & parameter)

$$J(x^{(1)} \dots x^{n_m}, \theta^1 \dots \theta^{n_u}) \text{ Cost Min} = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left((\theta^j)^T x^i - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{j=1}^{k} (x_k^i)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^j)^2$$

~~$\theta_0 = 0$~~
~~$x_0 = 1$~~ not needed here

$\theta = R^n$
$x = R^n$

Prior $\theta = R^{n+1}$
$x = R^{n+1}$

LOKENATH
Page No.
Date

Gradient - descent :

x is minimizing $\frac{\partial}{\partial x}$

$$x_k^i = x_k^i - \alpha \left(\sum_{j:r(i,j)=1} \left((\theta^j)^T x^i - y^{(i,j)}\right)\theta_k^j + \lambda x_k^{(i)} \right)$$

$$\theta_k^{(j)} := \theta_k^j - \alpha \left(\sum_{i:r(i,j)} \left((\theta^j)^T x^i - y^{(i,j)}\right) x_k^i + \lambda \theta_k^j \right)$$

filtering

(1) If x^i & θ^j are in column wise then use $\theta^T x$
x^i & θ^j both rowwise $x\theta^T$
so that this will give (1×1) matrix

recommending

$\Rightarrow$ $\| x^i - x^j \|$ should that small to recommend

How to find movie j related i

Mean normalization : (making querey of every matrix 0)

$\Rightarrow$ If any user has not rated any movie or has given every movie 0.

(1) calculate ~~mean~~ average of every movie.

rating $\rightarrow$ (4) $\left((\theta^j)^T x^i\right) + \mu^i$

If any user$^{(j)}$ has not rated any movies

the $\theta^j = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ $\quad y^{ij} = \begin{bmatrix} \mu_i \end{bmatrix}$

n

(1) take average of each movie
(2) subtract it from the rating matrix.
(3) Now do 2 matrix as real rating and do filtering