Recommender System

n_u = number of ~~movies~~ user

n_m = no of movies

$r(i,j) = 1$ if user j has rated movie i

$y(i,j)$ = rating given by user j to movie i (definetly only if $r(i,j) = 1$)

There are two types of recommendation System

- Content-based filtering
- Basic collaborative filtering

⇓

improve version

Collaborative filtering - A better approach:

Content based Recommandion:

In which we have given movies list with their features (romantic, comedy etc) and users list.

for example:-

Movies	user Alice(1)	user Bob(2)	user Carol(3)	user Dave(4)	(romantic) x_1	(a... x_2
Love at last	5	5	0	0	0.9	0
Romance Forever	5	?	?	0	1.0	0.0[illegible]
Cute Puppies	?	4	0	?	0.99	0
Nonstop car	0	0	5	4	0.1	1.0
Swords vs karate	0	0	5	?	0	0.9

In above table we have one movie list with two features (action and romantic) and have four user with rating.

We find missing rating and its base we decision can we recommend

4.95
-6.13

this Movie to user or not.

So we make every user have different table user rating with features and makes it linear regression.

For example | same as user 2, user 3, and so on..

x_0	x_1	x_2	Alice(1) user	$(\theta^j)^T x^i$	$-y^{(i,j)}$	square
m_1^i 1	0.9	0	5	1.85	-3.15	9.9225
m_2^i 1	1.0	0.01	5	2.02	-2.98	8.8804
m_3^i 1	0.99	0	?	?		
m_4^i 1	0.1	1.0	0	2.65	2.65	7.0225
m_5^i 1	0	0.9	0	2.3	2.3	5.29

we predict rating$^{(i)}$ of user j wit $(\theta^{(j)})^T x^i$ stars.

$\theta^{(j)}$ = parameter vector for user j

$(x^{(i)})$ = feature vector for movie i

So we tune thetas.

resume

$\theta_0 = 0.5$, $\theta_1 = 1.5$, $\theta_2 = 2$

0.1

$$\frac{dj}{d\theta_0} = \frac{1}{\cancel{x}} \Sigma \left((\theta^{j})^T \right) x^{(i)} - y^{(i,j)})^{\cancel{2}}$$

$$= \left(\Sigma ((\theta^{j})^T x^{(i)} - y^{(i,j)} \right) x_0$$

$$= (-1.18)(1)$$

$$\theta_{0\,new} = \theta_{0\,old} - \alpha \left(\frac{dj}{d\theta_0} \right)$$

$$= (0.5) - (0.1)(-1.18)$$

$$= 0.5 + 0.118$$

$$= \underline{0.618}$$

$$\frac{dj}{d\theta_1} = \frac{1}{\cancel{x}} \left(\Sigma \left(\theta^{(j)T} \right) x^{(i)} - y^{(i,j)} \right)^{\cancel{2}} + \lambda \theta_k^{(j)}$$

$$= \left(\Sigma \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) x_1 + \lambda \theta \right)$$

$$= (-5.55) + 0.5(1.5)$$

$$= -5.55 + 0.75$$

$$= -4.8$$

$$\theta_{1\,new} = \theta_{1\,old} - \alpha \left(\frac{dj}{d\theta_1} \right)$$

$$= 1.5 - 0.1(-4.8)$$

$$= 1.5 + 0.48$$

$$= \underline{1.98}$$

Scanned with CamScanner

$((\Theta^{(j)})^T - y^{(i,j)})\, x_1$

-2.835
-2.98
0.285
0
0
-0.0298
2.65
2.07

4.6902

$$\frac{dJ}{d\theta_1} = \frac{1}{x}\left(\sum (\Theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^{x} + \frac{\lambda \sum \Theta_k^{x}}{x}$$

(x_2)

$$= \left(\sum (\Theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^{x_2} + \lambda\, \Theta_k^{(j)}$$

$$= (4.6902) + (0.5)(2)$$

$$= 5.6902$$

$$\Theta_{2_{new}} = \Theta_{2_{old}} - \alpha \left(\frac{dJ}{d\Theta_2} \right)$$

$$= 2 - 0.1(5.6902)$$

$$= 1.43098$$

So its the first iteration of user 1 updated thetas once time. it is a big process So we compute for all users and updates its thetas and predict missing value.

we use:

To learn $\Theta^{(j)}$ (parameter for user j):

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

it's for one user.

to learn $\Theta^{(j=1,2,3 \ldots n)}$

$$\min_{\Theta^{(1)} \ldots \Theta^{(n)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\Theta^{j})^T (x^{(i)}) - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\Theta_k^{(j)})$$

it's for all users in one line.

* limitation of content based.
- It depends on availability of features; however, it doesn't work in absense of feature.
- if feature are not available then we have another approach called collaborative filtering.

$$h = \theta_0 + \theta_1 x_1 (x_2)^2$$

$\theta_0 = 2$; $\theta_1 = 1$; $\alpha = 0.01$, $\lambda = 10$

x_0	x_1	x_2	y	$(x_2)^2$	$y - h_\theta(x)$	$(y - h_\theta(x))^2$
1	2	1	10	1	6	36
1	3	2	15	4	1	1
1	2	3	17	9	-3	9
1	3	4	20	16	-30	900

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{n} (y^{(i)} - h_\theta(x^{(i)}))^2 + \lambda \sum_{j=1}^{m} \theta_j^2\right]$$

$h = \theta_0 x_1 + \theta_1 x_1 (x_2)^2$

$= (2)(1) + (1)(2)(1) = 2 + 2 = 4$ | 12

$= (2)(1) + (1)(3)(4) = 2 + 12 = 14$ | 12

$= (2)(1) + (1)(2)(9) = 2 + 18 = 20$ | -54

$= (2)(1) + (1)(3)(16) = 2 + 48 = 50$ | -1440

$= \frac{946}{4} + \frac{10(1)^2}{4}$

$= 236.5 + 2.5 = \underline{239}$ ↓

Scanned with CamScanner

$$\frac{dJ}{d\Theta_0} = \frac{2}{m} \sum_i (y^{(i)} - h_\theta(x)) \cancel{+ \frac{2\lambda\Theta_j}{m}}$$

$$= \frac{2}{m}(-26)$$

$$= \frac{-52}{4} = \underline{\underline{-13}}$$

$$\Theta_{0\,new} = \Theta_{0\,old} - \alpha \cdot (-13)$$

$$= 2 - 0.01(-13) = 2 + 0.13$$

$$= \underline{\underline{2.13}}$$

$$\frac{dJ}{d\Theta_1} = \frac{2}{m} \sum (y^{(i)} - h_\theta(x)) x_1 (x_2)^2 + \frac{\lambda 2\theta}{m}$$

$$= \frac{2}{4}(-1470) + \frac{(10)(2)(1)}{4}$$

$$= \frac{-2940}{4} + 5$$

$$= -735 + 5$$

$$= (-730)$$

$$\Theta_{1_{new}} = \Theta_{1_{old}} - \alpha\left(\frac{dJ}{d\theta_1}\right)$$

$$= 1 - 0.01(-730)$$

$$= 1 + 7.3 = \underline{\underline{8.3}}$$

Content Base filtering

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

Gradient descent updates

$$\Theta^j_k = \Theta^j_k - \alpha \sum_{i:r(i,j)=1} \left((\Theta^j)^T x^i - y^{(i,j)}\right) x^{(i)}_k$$

for $K=0$

$$\Theta^j_k = \Theta^j_k - \alpha\left(\sum_{i:r(i,j)} \left((\Theta^j)^T x^i - y^{(i,j)}\right) x_k + \lambda \Theta^j_k\right)^2$$

for $K \neq 0$

Basic Collaborative Filtering

Movie	User1	User2	User3	User4	(romance) x_1	(a x_2
"	5	5	0	0	?	?
"	5	?	?	0	?	?
"	?	4	0	?	?	?
"	0	0	5	4	?	?
"	0	0	5	?	?	?

if we have no features then we use Basic collaborative filtering.

in which firstly we set features means tunes the features value with the help of thetas. we set fix value of thetas and make this regression problem.

For example

set thetas for all users (set first value zero because x_0)

$$\theta^1 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \quad \theta^2 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \quad \theta^3 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}, \quad \theta^4 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

$x_0 = 0.5$, $\boxed{x_1 = 1}$, $x_2 = 2$

$h_2(\theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2$

users ↓	θ_0	θ_1	θ_2	Alice	$(\theta^i)^T x^i - y$ (1 user)	(x_1)	(x_2)
$\theta^{(1)}$	0	5	0	5	0	0	0
$\theta^{(2)}$	0	5	0	5	0	0	0
~~$\theta^{(3)}$~~	~~0~~	~~0~~		~~?~~	~~?~~		
$\theta^{(4)}$	0	0	5	0	10	10	20
$\theta^{(5)}$	0	0	5	0	10	10	20

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

$$= \frac{1}{2} (20)^2 + \frac{0.5}{2} (1+2)^2$$

$$= \frac{400}{2} + 2.25 = \underline{202.25}$$

↓ minimize

Gradient descent

$$\frac{dJ}{dx_2} = \sum_{j:r(i,j)} \left((\theta^i)^T x^i - y^{(i,j)}\right) x_1 + \lambda\, \theta_1$$

$$= 0 + (0.5)(1)$$

$$= 1.5$$

$$x_{2\,new} = x_{1\,old} - \alpha \left(\frac{dj}{dx_1}\right)$$

$$= 1 - (0.01)(1.5)$$

$$= 1 - 0.015 = \boxed{0.985} \quad x_2$$

$$\frac{dJ}{dx_2} = \frac{1}{\cancel{2}} \sum_{j:\delta(i,j)=1} ((\Theta^j)^T x^{(i)} - y^{(i,j)}) x_2 + \frac{\lambda}{\cancel{2}} \sum_{k=1}^{n} (\Theta_k \ldots$$

$= 100 + (0.5)(2)$

$= 102.5$

$$X_{2\,new} = X_{2\,old} - (\alpha)\left(\frac{dJ}{dx_2}\right)$$

$= 2 - (0.01)(102.5)$

$= 2 - 1.025 = \underline{\underline{0.975}}$ $\underline{\underline{X_2}}$

Given $\Theta^{(1)} \ldots \Theta^{(nu)}$, to learn $x^{(1)} \ldots x^{(nm)}$

$$\min_{x^{(1)} \ldots x^{(nm)}} \frac{1}{2} \sum_{i=1}^{nm} \sum_{j,\delta(i,j)=1} ((\Theta^j)^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{nm} \sum_{k=1} (x_k^{(i)} \ldots$$

CF Basic Algorithm

1. Assum thetas initial $\Theta^1 \ldots \Theta^{(nu)}$
2. Predict features: $x^{(1)} \ldots x^{(nm)}$
3. Using predict features predict thetas $x^{(1)} \ldots x^{(nm)}$
   as estimate $\Theta^{(1)} \ldots \Theta^{(nm)}$

Scanned with CamScanner

Keep improving both thetas and features by repeating step 2 and 3

$\Theta \rightarrow x \rightarrow \Theta \rightarrow x \rightarrow \Theta \rightarrow x \rightarrow \dots$

Collaborative filtering (A better Approach)

Given (features $x^{(1)} \dots x^{(nm)}$) estimates (thetas $\Theta^{1} \dots \Theta^{(nu)}$)

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

Given (thetas $\Theta^{1} \dots \Theta^{nm}$) estimates (Features $x^{1} \dots x^{nm}$)

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

Minimizing $x^1 \dots x^{(nm)}$ and $\Theta^1 \dots \Theta^{(nu)}$ Simultaneously:

$$J(x^{(1)} \dots x^{nm}, \Theta^1 \dots \Theta^{nu}) = \frac{1}{2} \sum_{(i,j): r(i,j)=1} \left((\Theta^{j})^T x^{i} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{nm} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{nu} \sum_{k=1}^{n} \left(\Theta_k^{j} \right)^2$$

$$\min_{\substack{x^1 \dots x^{nm} \\ \Theta^1 \dots \Theta^{nu}}} J$$