Content Based Recommendations

Movie	Alice	Bob	Carol	Dave	x_1 (romance)	x_2 (action)
Love at Last	5	5	0	0	0.9	0
Romance Forever	5	?	?	0	1.0	0.01
Cute Puppies of Love	?	4	0	?	0.99	0
Non-stop Car chases	0	0	5	4	0.1	1.0
Swords v/s karates	0	0	5	?	0	0.9

features

Our objective is to populate the ? with ratings that they did not ~~wath.~~ watch.

n_u = number of users

n_m = number of movies

$r^{(i,j)} = 1$ if user j has rated movie i.

$y^{(i,j)}$ = rating given by user j to movie i (defined only if $r^{i,j}=1$) $0 \le y^{(i,j)} \le 5$.

m_j = number of movies rated by user j.

x_i = is the feature corresponding to genre i. x_0 is the feature associated with intercept term.

Let x^i be the feature vector associated with movie i. Thus,

$$x^1 = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}, \quad x^2 = \begin{bmatrix} 1 \\ 1.0 \\ 0.01 \end{bmatrix}, \quad x^3 = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}, \quad x^4 = \begin{bmatrix} 1 \\ 0.1 \\ 1.0 \end{bmatrix}, \quad x^5 = \begin{bmatrix} 1 \\ 0 \\ 0.9 \end{bmatrix}$$

Regression Problem is: For each user j, learn a parameter θ^j belongs ~~where~~ the $\mathbb{R}^{n+1}$, where n is the no. of features except the feature associated with the intercept term, predict user j as rating movie i with $(\theta^j)^T x^i$.

← inner product

Transpose of θ^j

Suppose θ^1 is $\begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$. Since $x^3 = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}$, Alice would rate $(\theta^1)^T x^3 = 5 \times 0.99 = 4.5$ stars to Cute puppies of love. We need to learn θ^j.

To learn $\theta^{(j)}$:

actual rating by user j on movie i

$$\min_{\theta^{(j)}} \left[\sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 / 2m^{(j)} + \frac{\lambda}{2m^{(j)}} \sum_{k=1}^{n} \left(\theta_k^{(j)}\right)^2 \right] \quad (\theta^{(j)} \in \mathbb{R}^{n+1},$$

where n is the number of features)

regularization term

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

$\Rightarrow$ To learn $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(n_u)}$:

$$\underbrace{\min_{\theta^{(1)},\ldots,\theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)}\right)^2}_{J(\theta^{(1)},\ldots,\theta^{(n_u)})} \quad \text{— (1)}$$

Extra sum is for all users.

Gradient-descent update

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) x_k^{(i)} \quad (\text{for } k=0)$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \underbrace{\left(\sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) \cdot x_k^{(i)} + \lambda \theta_k^{(j)} \right)}_{\frac{\partial}{\partial \theta_k^{(j)}} J(\theta^{(1)},\ldots,\theta^{(n_u)})} \quad (\text{for } k \neq 0)$$

Collaborative Filtering

It is difficult to have $x_1, x_2, \ldots$ feature because it is not possible for each user to see all the movie. We have no idea how romantic or action each movie is?

x_1 (romance)	x_2 (action)
?	?
?	?
?	?
?	?
?	?

However, suppose our users have told us how much they like romantic movies & how much they like action movies?

Feature vectors

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

$\theta^{(i)}$ is user i's preference over action & romance movies.

Alice (user 1) likes romantic movie..

Carol (user 3) ... action ...

Now, given movie 1, (Love at last)

we need to find x' such that,

$$(\theta^{(1)})^T x' \simeq 5$$
$$(\theta^{(2)})^T x' \simeq 5$$
$$(\theta^{(3)})^T x' \simeq 0$$
$$((\theta^{(4)})^T x' \simeq 0$$

Optimization problem

Given $\theta^{(1)}, \ldots, \theta^{(n_u)}$ i.e., users' preferences for different genres, to learn $x^{(i)}$:

$$\min_{x^{(i)}} \frac{1}{2} \underbrace{\sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2}_{\text{Choose features } x^i \text{ so that all users that have rated the movies such that it is not too far in the squared error sense from the actual value that the user has rated.}} + \underbrace{\frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2}_{\text{regularisation term is added to prevent the features from becoming to big}}$$

Given $(\theta^{(1)}, \theta^{(2)}, \ldots \theta^{(n_u)})$, to learn $x^{(1)}, \ldots, x^{(n_m)}$:

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

Content Based → Given $(x^{(1)}, x^{(2)}, \ldots x^{(n_m)})$ → and movie ratings, can estimate $\theta^{(1)}, \ldots, \theta^{(n_u)}$ (parameters)

Collaborative Filtering → Given $\theta^{(1)}, \ldots, \theta^{(n_u)}$, can estimate $x^{(1)}, \ldots, x^{(n_m)}$.

Guess θ → learn x → θ → x .....

Collaborative Filtering ⟺ Content based filtering will be used to ~~simultaneously~~ back & forth learn about movie ratings & parameters. If used simultaneously?

Minimizing $x^{(1)}, \ldots, x^{(n_m)}$ and $\theta^{(1)}, \ldots, \theta^{(n_u)}$ simultaneously:

instead of going back & forth.

$$J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)})$$

$$= \frac{1}{2} \sum_{(i,j): r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2 \quad \text{— (3)}$$

$$\min_{\substack{x^{(1)}, \ldots, x^{(n_m)} \\ \theta^{(1)}, \ldots, \theta^{(n_u)}}} J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)})$$

* Here $x \in \mathbb{R}^n$ & $\theta \in \mathbb{R}^n$. We are doing away with features & parameters associated with intercept term.

Collaborative filtering algorithm

1. Initialize $x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)}$ to small random values.
2. Minimize $J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)})$ using gradient descent (or an advanced optimization ~~problem~~ algorithm). E.g. for every $j = 1, \ldots, n_u$, $i = 1, \ldots, n_m$:

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

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

Users who have not rated any movies

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \quad \text{last column} \rightarrow \text{Eve (user 5)}$$

Since $(i,j): r(i,j) = 1$ does not happen for user 5, the first term does not apply in the minimization problem in (3).

The only term that user 5 affects is $\frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)}\right)^2$

$$\Rightarrow \frac{\lambda}{2}\left[\left(\theta_1^{(5)}\right)^2 + \left(\theta_2^{(5)}\right)^2\right]$$

will be minimized if $\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. So, $(\theta^{5})^T x^{(i)} = 0$ for every movie i. This does not seem useful.

Mean Normalization

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \quad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \rightarrow Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

$Y - \mu$

$\downarrow$

Learn θ^{j}, x^{i}

For user j, on movie i predict:

$$(\theta^{(j)})^T (x^{(i)}) + \mu_i$$

User 5 (Eve): $\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} + \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$

$= \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \simeq \underline{\mu}$ $\rightarrow$ for new user who have not rated any movies, the movie will be rated on avg.

3. For a user with parameters θ and a movie with (learned) features x, predict a star rating $\theta^T x$. (3)

Alternative to Collaborative Filtering

Take all the elements of the table and create a matrix Y

$$= \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

Predicted ratings:

$$\begin{bmatrix} (\theta^{(1)})^T x^{(1)} & (\theta^{(2)})^T x^{(2)} & \cdots & (\theta^{(n_u)})^T x^{(1)} \\ (\theta^{(1)})^T x^{(2)} & (\theta^{(2)})^T x^{(2)} & \cdots & (\theta^{(n_u)})^T x^{(2)} \\ \vdots & & \ddots & \vdots \\ (\theta^{(1)})^T x^{(n_m)} & (\theta^{(2)})^T x^{(n_m)} & \cdots & (\theta^{(n_u)})^T x^{(n_m)} \end{bmatrix}$$

Low rank matrix factorization $\leftarrow X \cdot \Theta$

where $(\theta^{j})^T x^{i}$ is the rating given by user j to movie i.

$$X = \begin{bmatrix} -(x^{(1)})^T- \\ -(x^{(2)})^T- \\ \vdots \\ -(x^{(n_m)})^T- \end{bmatrix} \qquad \Theta = \begin{bmatrix} | & | & & | \\ \theta^{(1)} & \theta^{(2)} & \cdots & \theta^{(n_u)} \\ | & | & & | \end{bmatrix}$$

Finding related movies

For each product i, we learn a feature vector $x^{(i)} \in \mathbb{R}^n$

x_1 = romance, x_2 = action, x_3 = comedy, x_4 = ----

How to find movies j related to movie i?

$\|x^{(i)} - x^{(j)}\|$ small $\Rightarrow$ movie j & i are similar.

5 most similar movies to movie i:

$\rightarrow$ Find the 5 movies j with the smallest $\|x^{(i)} - x^{(j)}\|$.