May 31, 2021

Movie	Alice	Bob	Carol	Dave	Romance x_1	Action x_2
$x^{(1)}$	5	5	0	0	?	?
$x^{(2)}$	5	?	?	0	?	?
$x^{(3)}$	?	4	0	?	?	?
$x^{(4)}$	0	0	5	4	?	?
$x^{(5)}$	0	0	5	?	?	?

If we get the ratings for each user on each movie and their parameter vectors, we can infer what the features x_1 and x_2 are for each movie.

e.g. $\theta_1 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$, $\theta_2 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$, $\theta_3 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$, $\theta_4 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$

$$\left.\begin{aligned} (\theta^{(1)})^T x^{(1)} &\approx 5 \\ (\theta^{(2)})^T x^{(1)} &\approx 5 \\ (\theta^{(3)})^T x^{(1)} &\approx 0 \\ (\theta^{(4)})^T x^{(1)} &\approx 0 \end{aligned}\right\} \therefore x^{(1)} = \begin{bmatrix} 1 \\ 1.0 \\ 0.0 \end{bmatrix}$$

Optimization Algorithm:

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

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

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

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

To Learn $x^{(1)} \ldots x^{(n_m)}$ and $\theta^{(1)} \ldots \theta^{(n_u)}$ Simultaneously:

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

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

$$+ \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(\theta_k^{(j)} \right)^2$$

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.

   i.e. For every $j = 1 \ldots n_u$, $i = 1 \ldots n_m$:

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

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

3. For a user w/ parameters θ and a movie w/ <u>learned</u> features x, predict a star rating of $\theta^T x$.