Recommender Systems

Problem formulation

n_u = no. users
n_m = no.movies
$r(i,j)=1$ if user j has rated movie i
$y(i,j)=1$ = rating given by user j to movie i (defined only if $r(i,j)=1$)

For each user j, learn a parameter $\theta^{(j)}$. Predict user j as rating movie i with $(\theta^{(j)})^T x^{(i)}$.

$\theta^{(j)}$ = parameters vector for user j
$x^{(i)}$ = feature vector for movie i
$m^{(j)}$ = no. of movies rated by user j

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

The difference between traditional linear regression is the absence of divisor m.

optimization and update

Further, Optimization objective:
$min_{\Theta}\ J=\frac{1}{2}\ \sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}((\theta^{(j)})^T(x^{(i)})-y^{(i,j)})^2+\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2$

Gradient descent update:
$\theta_k^{(j)}=\theta_k^{(j)}-\alpha\sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)}-y^{(i,j)})x_k^{(i)}$ for $k=0$
$\theta_k^{(j)}=\theta_k^{(j)}-\alpha(\sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)}-y^{(i,j)})x_k^{(i)}+\lambda\theta_k^{(j)})$ for $k\neq 0$

Given Θ, to learn $x^{(i)}$:
$min_\Theta\ J = \frac{1}{2}\ \sum_{i=1}^{n_m}\sum_{j:r(i,j)=1}((\theta^{(j)})^T(x^{(i)}) - y^{(i,j)})^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}(x_k^{(j)})^2$

Now, we can update $\theta \rightarrow x \rightarrow \theta \rightarrow$ ... step by step.

Collaborative filtering

minimize X and Θ simultaneously:
$J(x^{(1)}, ..., x^{(n_m)}, \theta^{(1)}, ..., \theta^{(n_u)}) = \frac{1}{2}\sum_{(i,j):r(i,j)=1}((\theta^{(i)})^T x^{(i)} - y^{(i,j)})^2 +$
$\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}(x_k^{(j)})^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2$

It's important to note that we drop $x_0 = 1$

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

predict: $\theta^T x$

Low Rank Matrix Factorization

$Y = X\Theta^T$
Predicting how similar two movies i and j are can be done using the distnce between their respective feature vectors x. Specifically, we are looking for a small value of $||x^{(i)} - x^{(j)}||$

Implementation Detail: Mean Normalization

The new user don't have record $r_{(i,j)} = 1$, which means θ won't be updated.
A method is that assigned by average value of $r_{(i,j)} = 1$, of course it's for vectorization. $\mu_i = \frac{\sum_{j:r(i,j)=1} Y_{i,j}}{\sum_j r(i,j)}$
Or replace Y by $Y - \mu$, and prediction should be $(\theta^{(j)})^T x^{(i)} + \mu_i$