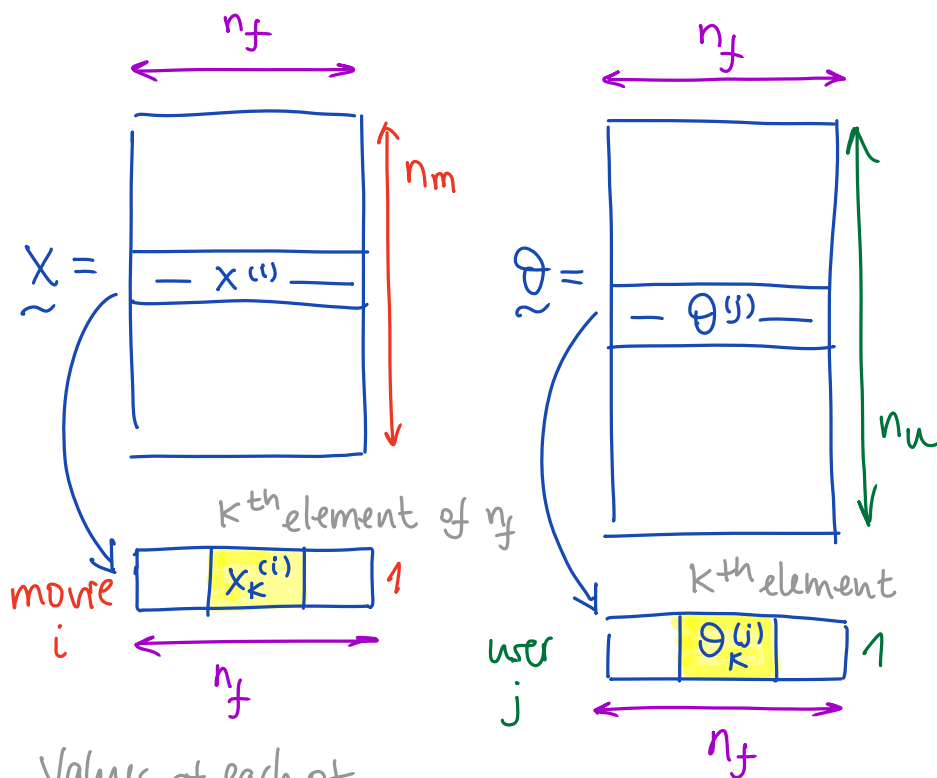
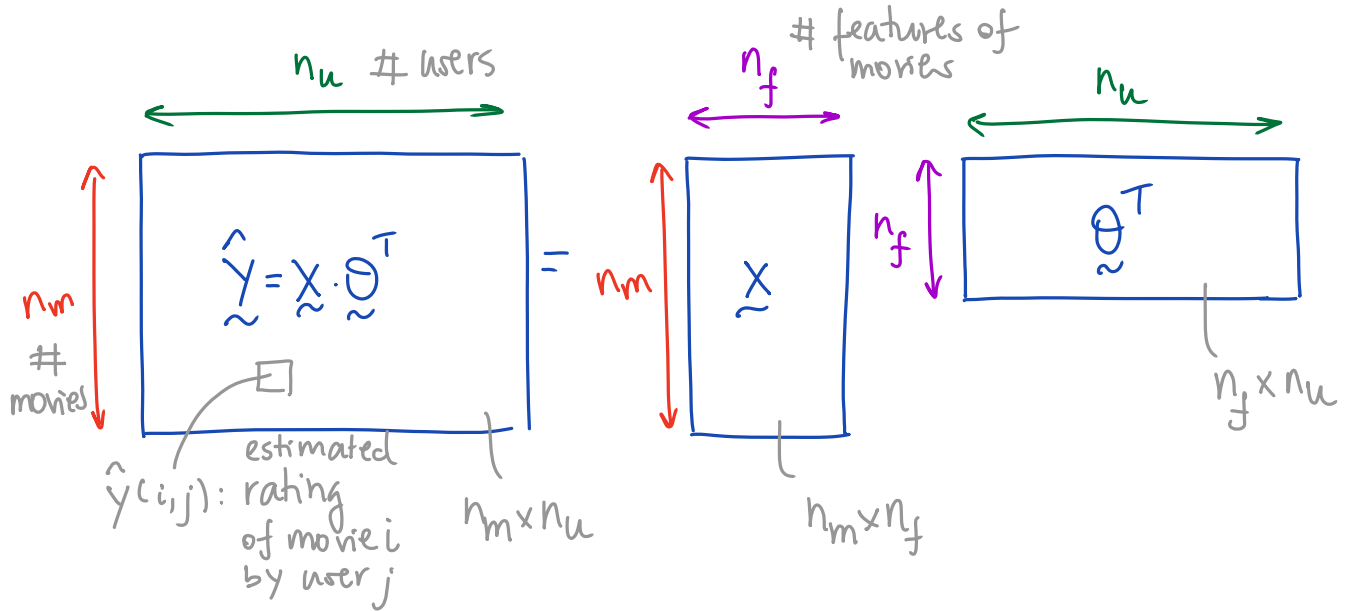


Recommender Systems - Low Rank Matrix Factorization for Collaborative Learning



Values of each of the n_f features for each movie i

Degree of affinity of each movie feature (n_f) given by each user j

Exercise sizes

$n_m = 1682$ movies

$n_u = 943$ users

$n_f = 100$ features for each movie

$n_f < n_m, n_u$,
I understand

Learnt automatically.

- romance
- modern
- sci-fi
- action
- blockbuster
- ...

Goal: find \underline{X} & $\underline{\Theta}$ so that the difference between \underline{Y} and $\hat{\underline{Y}} = \underline{X} \cdot \underline{\Theta}^T$ is minimum, being \underline{X} & $\underline{\Theta}$ the ones defined above.

Cost and Gradient (unregularized)

Even though the cost $J(\underline{X}, \underline{\Theta})$ is quite straightforward to compute vectorized, the gradient ∇J is a little bit tricky.

$$J(\underline{X}, \underline{\Theta}) = J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_m)}) =$$

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

$$= \frac{1}{2} \sum_{(i,j): r(i,j)=1} \left(\underline{X} \cdot \underline{\Theta}^T - \underline{Y} \right)^2$$

column vectors,
not as defined above!

matrices, as
defined above

$$\left(\frac{\partial J}{\partial x^{(i)}} \right)^T = \left[\frac{\partial J}{\partial x_1^{(i)}}, \dots, \frac{\partial J}{\partial x_K^{(i)}}, \dots, \frac{\partial J}{\partial x_{n_f}^{(i)}} \right] =$$

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

These have
always values

This has
values iff
 $r(i,j)=1$

Sum over all
j values, only
take values when $r(i,j)=1$

