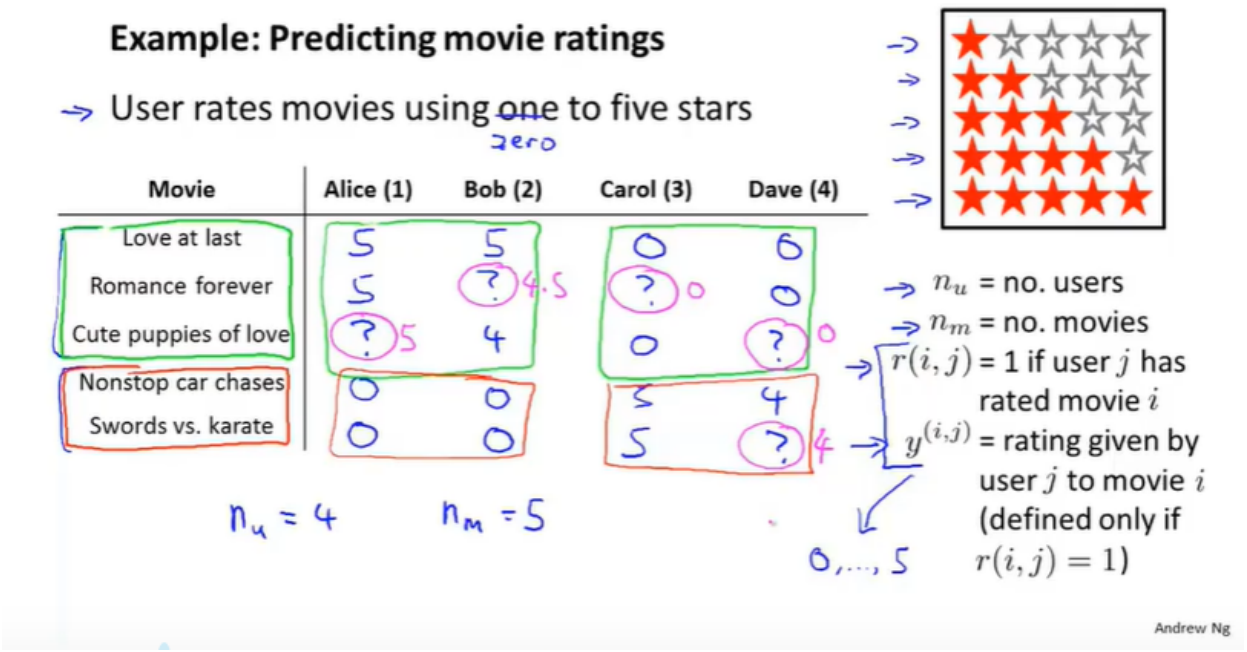
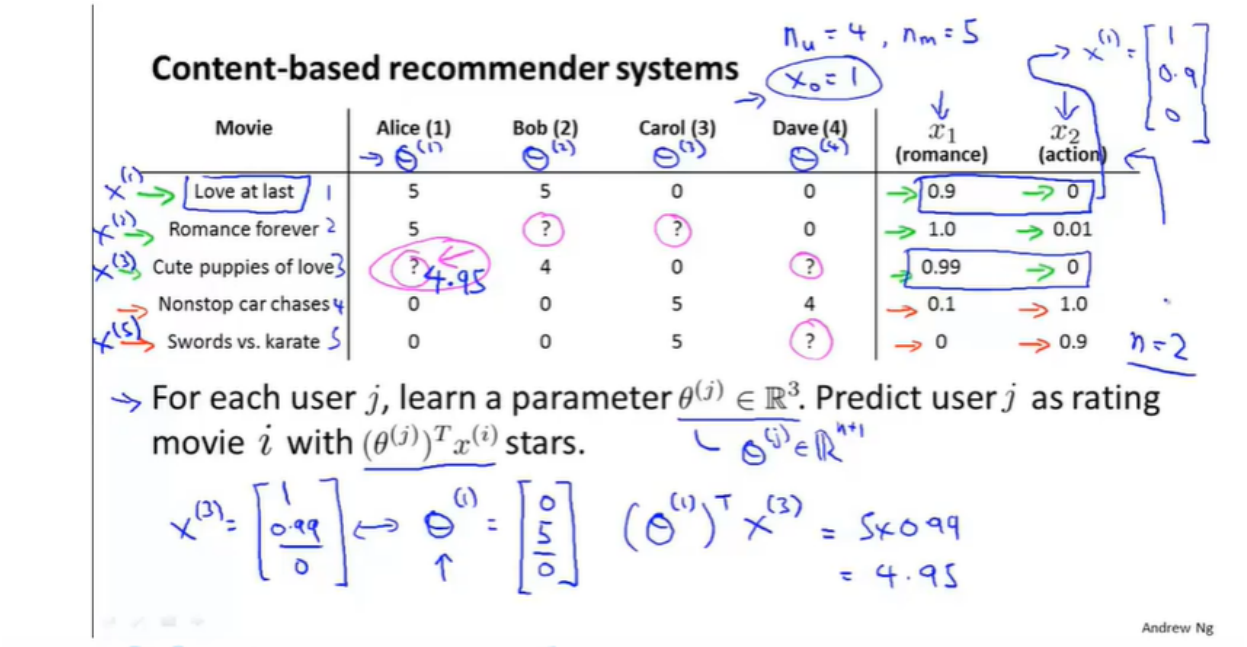


在预测电影评分这个例子中，我们应该预测用户会给他没有看过的电影打多少分，再推荐他会打高分的电影给他。



我们预测问号上的喜爱程度，是将电影中的成分当作特征，再添加一个截距项1，得到x向量，再用某种方式得到参数向量，之后我们利用参数向量的转置乘以x向量，得到了喜爱程度。

Problem formulation

- $r(i,j) = 1$ if user j has rated movie i (0 otherwise)
- $y^{(i,j)}$ = rating by user j on movie i (if defined)

- $\theta^{(j)}$ = parameter vector for user j
- $x^{(i)}$ = feature vector for movie i
- For user j, movie i, predicted rating: $(\theta^{(j)})^T(x^{(i)})$
- $m^{(j)}$ = no. of movies rated by user j

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

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

Andrew Ng

为了得到sita参数向量，最小化代价方程。将预测喜爱值看作线性回归问题，得到代价方程，加入正则化项，截距项不计入正则化项，所以k从1开始。代价方程中去掉m，因为是常数项。

Optimization objective:

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

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

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

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

Andrew Ng

第二个代价方程针对所有用户。

利用梯度下降来求解参数向量

k=0时没有正则化项，当k！=0，正则化项会对导数产生影响，所以更新梯度下降方程针对两种情况。