问题动机

Problem n	notivat	ion			. ↓	L	X ₀ =
Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)	
X(1) Love at last	75	p 5	n 0	n 0	A 1.0	40.	0
Romance forever	5	?	?	0	?	?	X0= [10]
Cute puppies of love	?	4	0	?	3	?	(0.0)
Nonstop car chases	0	0	5	4	5	? (Z(I)
Swords vs. karate	0	0	5	?	? .	3	× 10
$\Rightarrow \boxed{\theta^{(1)} =}$	$\theta^{(2)}$	$ \begin{array}{c} \begin{bmatrix} 0 \\ \hline 0 \end{bmatrix}, $	$\theta^{(3)} = 0$	$\theta^{(4)} =$		(1	$(\bigcirc^{(1)} X^{(1)})^{*} X^{(1)} $ $(\bigcirc^{(2)} X^{(1)})^{*} X^{(1)} $ Andrew Ng

假设我们并不知道电影中所包含的属性成分(特征),但我们有每个用户喜欢不同类型电影的参数sita向量,和每个用户给这部电影打的分值,所以我们可以倒推出一部电影的特征包含。

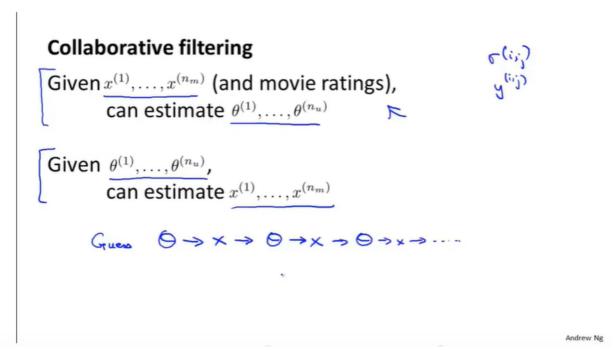
Optimization algorithm

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

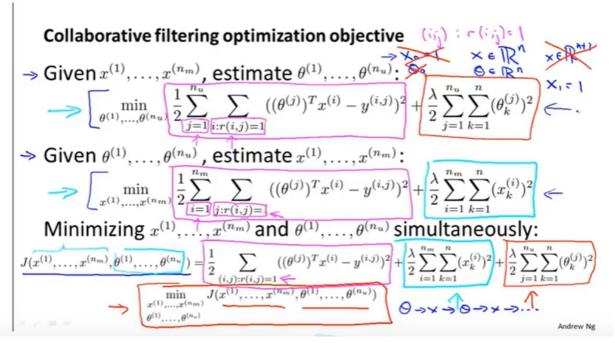
 $\Rightarrow \min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} (\underline{(\theta^{(j)})^T x^{(i)}} - \underline{y^{(i,j)}})^2 + \frac{\lambda}{2} \sum_{k=1}^n (x_k^{(i)})^2$

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

给定每个用户喜好的参数sita,最小化代价函数,学习每个电影的包含特征向量。



协同过滤过程:根据用户评分,先猜测出一些用户喜爱参数sita,再学习电影特征,再利用学习到的电影特征学习用户喜爱参数,进行迭代。



将两个式子结合起来,构成新的代价函数,不再需要在两个参数之间互相转换,这两个参数可以同时进行最小化

协同过滤算法流程

Collaborative filtering algorithm

Xoel Xell, Oell

00

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

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

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

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

x, sita向量取随机值,利用梯度下降最小化代价函数,再利用sita转置和x计算评分

低秩矩阵分解

寻找相似的电影: 找到特征参数距离最近的

Finding related movies

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

How to find movies j related to movie i?

small
$$\|x^{(i)} - x^{(j)}\| \rightarrow movie \hat{j}$$
 and i are "similar"

5 most similar movies to movie *i*:

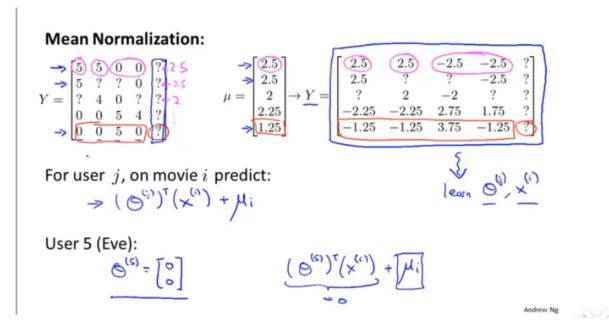
 \rightarrow Find the 5 movies j with the smallest $||x^{(i)} - x^{(j)}||$.

对于没有给过任何电影评分的用户,因为代价函数中的正则化项,所以用户的喜爱参数sita都会被置为0,所以最后计算预测评分时,sita转置乘以x,结果为0,对于没有给任何电影评分过的用户,预测评分都是0显然不合理,所以应该用均值归一化来解决这个问题。

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	Eve (5)	[0 0 0
Love at last	5	5	0	0	? 0	5 5 0 0 5
Romance forever	5	?	?	0	? (6	V 3 4 0 3 3
Cute puppies of love	?	4	0	?	? 👨	$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & 0 \\ 5 & ? & ? & 0 & 0 \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 4 & ? \end{bmatrix}$
Nonstop car chases	0	0	5	4	? 0	0 0 5 4 1
Swords vs. karate	0	0	5	?	? 🗅	
$\min_{\substack{x^{(1)},\dots,x^{(n_m)}\\\theta^{(1)},\dots,\theta^{(n_n)}}} \frac{1}{2}$	$\sum_{(i,j):r(i,j)=1}$	1	$(s) = \begin{bmatrix} o \\ o \end{bmatrix}$	$+\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^n$	L	$\frac{\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2}{\frac{\lambda}{2}\left[\left(\Theta_1^{(s)}\right)^2+\left(\Theta_2^{(s)}\right)^2\right]} \leftarrow$

均值归一化:

给评分矩阵都减去每一行的平均值,得到每个用户给一部电影打分的平均值μ,再用从未评分过的用户的喜爱参数sita转置乘以x,再加上平均值μ的得到结果预测评分,预测评分结果是平均值μ,这个预测是有意义的。



推荐系统中,关注没给任何电影评分过的用户,比关注没被评分过的电影有意义。没被评价过的电影可能不该推荐给任何人。