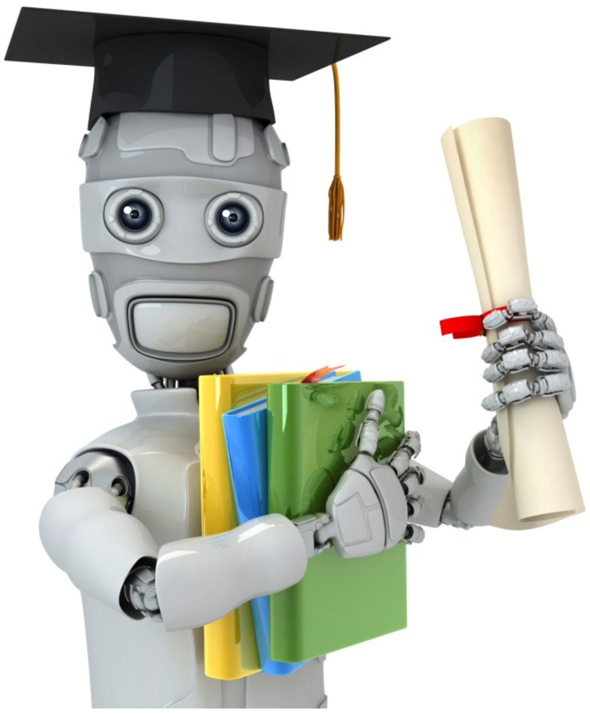


Machine Learning

Recommender Systems

Problem formulation

问题规划

Example: Predicting movie ratings

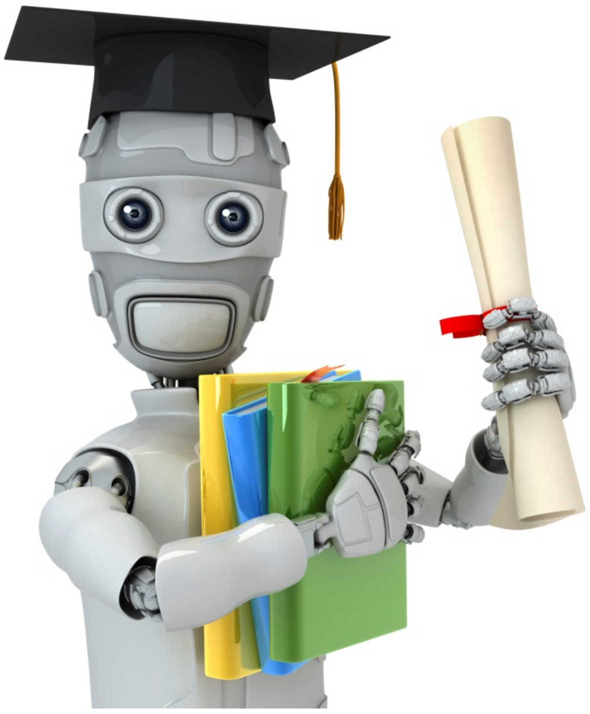
- User rates movies using ~~one~~ zero to five stars

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	? 4.5	? 0	0
Cute puppies of love	? 5	4	0	? 0
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	? 4

$n_u = 4$ $\quad$ $n_m = 5$

- n_u = no. users
- n_m = no. movies
- 是否评价 $r(i,j)$ = 1 if user j has rated movie i
- 评价分数 $y^{(i,j)}$ = rating given by user j to movie i (defined only if $r(i,j) = 1$)

0,…, 5



Machine Learning

Recommender Systems

Content-based recommendations

基于内容的推荐算法 ($x \to \theta$)

Content-based recommender systems

$n_u = 4,\ n_m = 5$

$x_0 = 1$

$x^{(1)} = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}$

Movie	Alice (1) $\theta^{(1)}$	Bob (2) $\theta^{(2)}$	Carol (3) $\theta^{(3)}$	Dave (4) $\theta^{(4)}$	x_1 (romance)	x_2 (action)
$x^{(1)}$ Love at last 1	5	5	0	0	0.9	0
$x^{(2)}$ Romance forever 2	5	?	?	0	1.0	0.01
$x^{(3)}$ Cute puppies of love 3	? 4.95	4	0	?	0.99	0
Nonstop car chases 4	0	0	5	4	0.1	1.0
$x^{(5)}$ Swords vs. karate 5	0	0	5	?	0	0.9

每部电影有两个特征

$n = 2$

线性回归

For each user j, learn a parameter $\theta^{(j)} \in \mathbb{R}^3$. Predict user j as rating movie i with $(\theta^{(j)})^T x^{(i)}$ stars.

$\theta^T x^{(i)}$

$(\theta^{(j)})^T x^{(i)}$

$\theta^{(j)} \in \mathbb{R}^{n+1}$

$x^{(i)}$: 第 i 部电影 / 第 i 个特征

$x^{(3)} = \begin{bmatrix} 1\ (x_0) \\ 0.99 \\ 0 \end{bmatrix} \longleftrightarrow \theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$

$(\theta^{(1)})^T x^{(3)} = 5 \times 0.99 = 4.95$

Andrew Ng

$h_\theta(x^{(i)}) = \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \theta_3 x_3^{(i)} + \cdots$

Alice: $h_\theta(x^{(1)}) = \theta_1^{(1)} x_1^{(1)} + \theta_2^{(1)} x_2^{(1)} + \theta_3^{(1)} x_3^{(1)}$ $\quad \theta^{(1)} = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$, $x^{(1)} = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}$

Bob: $h_\theta(x^{(2)}) = \theta_1^{(2)} x_1^{(2)} + \theta_2^{(2)} x_2^{(2)} + \theta_3^{(2)} x_3^{(2)}$ $\quad \theta^{(2)} = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$, $x^{(2)} = \begin{bmatrix} 1 \\ 1.0 \\ 0.01 \end{bmatrix}$

Problem formulation

- $r(i,j) = 1$ if user j has rated movie i (0 otherwise) 所有样本,叫做一个θ向量
- $y^{(i,j)} =$ rating by user j on movie i (if defined)
- $\theta^{(j)}$ = parameter vector for user j 每个用户有一个θ的向量(与前面不同)
- $x^{(i)}$ = feature vector for movie i
- For user j, movie i, predicted rating: $(\theta^{(j)})^T(x^{(i)})$

$\theta^{(j)} \in \mathbb{R}^{n+1}$

- $m^{(j)}$ = no. of movies rated by user j 用户j评价所有电影数量

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

针对每个用户j进行训练所有评价过的电影 $\sum_{i:r(i,j)=1}$

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

用户的参数 电影各个特征 $x^{(i)}$

Andrew Ng

针对所有用户 $\sum_{i=1}^{m}$

[code to compute $J(\theta)$]

$$J(\theta) = \left[-\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) \log 1 - h_\theta(x^{(i)})\right] + \boxed{\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2}$$

$h_\theta(x^{(i)}) = \theta^T x^{(i)}$

用户各个特征 $x^{(i)}$

Optimization objective: 代价函数

To learn $\theta^{(j)}$ (parameter for user j): 单个用户，给那个用户所有电影的特征$x^{(i)}$

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

→ 得到最优$\theta^{(j)}$

To learn $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(n_u)}$: 多个用户，给那些用户所有电影的特征$x^{(i)}$ → 得到最优$\theta^{(j)}$

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

$\theta^{(1)}, \dots, \theta^{(n_u)}$

Andrew Ng

Optimization algorithm:

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

Gradient descent update:

除所有第k个特征参数

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1}\left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)x_k^{(i)} \quad (\text{for } k = 0)$$

第0项不用正则化

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \underbrace{\left(\sum_{i:r(i,j)=1}\left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)x_k^{(i)} + \lambda\theta_k^{(j)}\right)}_{\frac{\partial}{\partial\theta_k^{(j)}} J(\theta^{(1)},\dots,\theta^{(n_u)})} \quad (\text{for } k \neq 0)$$

$\cancel{\frac{1}{m^{(j)}}}$

Andrew Ng

线性回归导数包括该项

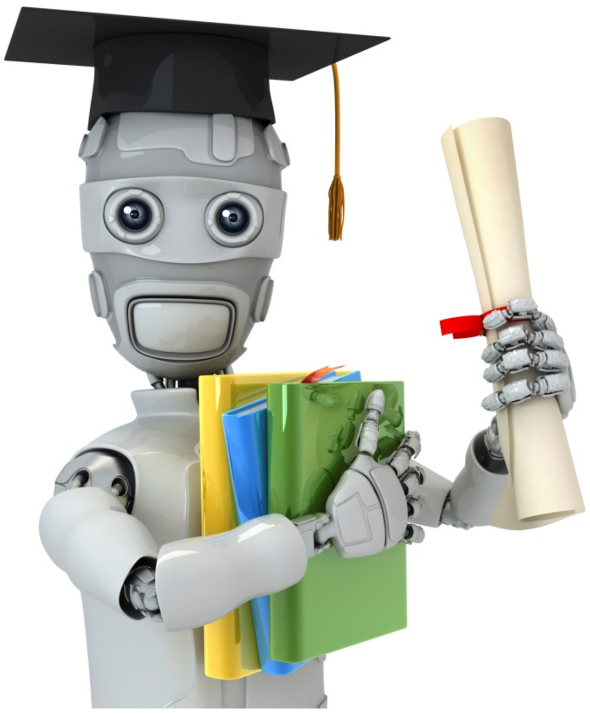
本例去掉了该项的所有函数

多个用户进行训练 ... $i:r(i,j)=1$

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

用户的参数　电影各个特征 $x^{(i)}$

Andrew N



Machine Learning

Recommender Systems

Collaborative filtering

协同过滤：自行学习所要使用的特征 $(\theta \to x)$

Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

Problem motivation

$x_0 = 1$

Movie	Alice (1) $\theta^{(1)}$	Bob (2) $\theta^{(2)}$	Carol (3) $\theta^{(3)}$	Dave (4) $\theta^{(4)}$	x_1 (romance)	x_2 (action)
$x^{(1)}$ ~~Love at last~~	5	5	0	0	~~?~~ 1.0	~~?~~ 0.0
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

$$x^{(1)} = \begin{bmatrix} 1 \\ 1.0 \\ 0.0 \end{bmatrix}$$

$x^{(1)}$?

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \quad \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

每个人,对不同电影的喜爱程度

$\theta^{(j)}$

$(\theta^{(1)})^T x^{(1)} \approx 5$

$(\theta^{(2)})^T x^{(1)} \approx 5$

$(\theta^{(3)})^T x^{(1)} \approx 0$

$(\theta^{(4)})^T x^{(1)} \approx 0$

用户对电影的评分

$\theta^{(j)}$ $x^{(i)}$

Optimization algorithm

已知 待求

Given $\theta^{(1)},\dots,\theta^{(n_u)}$, to learn $x^{(i)}$: 每部电影，对于所有用户喜爱程度$\Sigma\theta^{(j)}$

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

得电影特征$x^{(i)}$

所有电影，对于所有用户喜爱程度$\Sigma\theta^{(j)}$

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

得电影特征$x^{(i)}$

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

Optimization objective: 代价函数

To learn $\theta^{(j)}$ (parameter for user j): 每个用户，对那个电影都特征$\Sigma x^{(i)}$

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

得喜爱程度$\theta^{(j)}$

To learn $\theta^{(1)},\theta^{(2)},\dots,\theta^{(n_u)}$: 多个用户，对那个电影都特征$\Sigma x^{(i)}$ 得喜爱程度$\theta^{(j)}$

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

$\Theta^{(1)}, \dots, \Theta^{(n_u)}$

Collaborative filtering

已知电影特征 $\Sigma x^{(i)} \rightarrow \Sigma \theta^{(j)}$

$r^{(i,j)}$

$y^{(i,j)}$

Given $x^{(1)}, \ldots, x^{(n_m)}$ (and movie ratings),

can estimate $\theta^{(1)}, \ldots, \theta^{(n_u)}$

相互协同

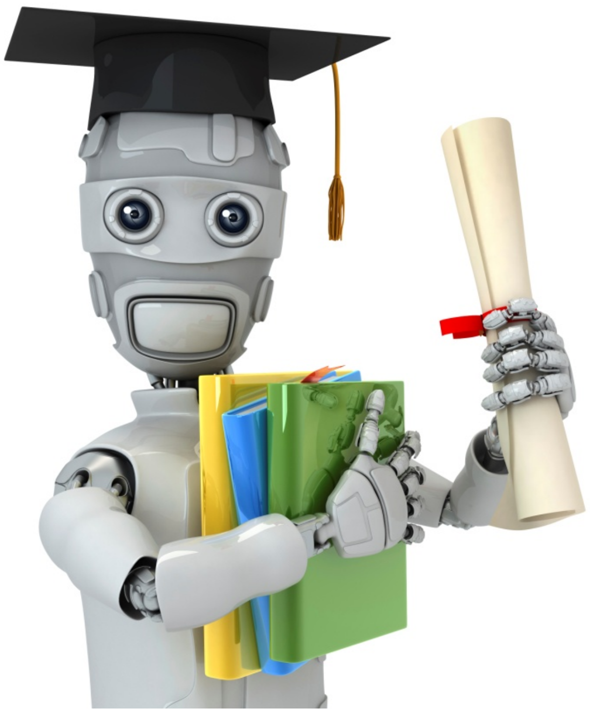
Given $\theta^{(1)}, \ldots, \theta^{(n_u)}$,

已知用户喜好特征 $\Sigma \theta^{(j)} \rightarrow \Sigma x^{(i)}$

can estimate $x^{(1)}, \ldots, x^{(n_m)}$

Guess $\theta \rightarrow x \rightarrow \theta \rightarrow x \rightarrow \theta \rightarrow x \rightarrow \ldots$

随机猜取 θ



Machine Learning

Recommender Systems

Collaborative filtering algorithm

协同过滤算法

Collaborative filtering optimization objective

$(i,j): r(i,j)=1$ 取消 x_0

交替计算 θ、x $\quad \theta \to x \to \theta \to x \cdots$

$\cancel{x_0}$ $\quad x \in \mathbb{R}^n \quad \cancel{x \in \mathbb{R}^{n+1}}$

$\cancel{\theta_0}$ $\quad \theta \in \mathbb{R}^n$

$x_1 = 1$

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

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

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

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

同时计算出 θ, x

Minimizing $x^{(1)},\ldots,x^{(n_m)}$ and $\theta^{(1)},\ldots,\theta^{(n_u)}$ simultaneously:

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

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

$\theta \to x \to \theta \to x \to \cdots$

Andrew Ng

取消 x_0?：现在是计算所有特征 (θ, x)，没必要硬编码加一

算法可以自行训练得到这个特征值 $(x_1=1)$

Collaborative filtering algorithm

$x_0 \cancel{= 1}$ $\quad x \in \mathbb{R}^n, \ \theta \in \mathbb{R}^n$

初始化为小的随机数

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 (or an advanced optimization algorithm). E.g. for every $j = 1, \ldots, n_u, i = 1, \ldots, n_m$:

$\cancel{\theta_0}$, θ_1, …, θ_n

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

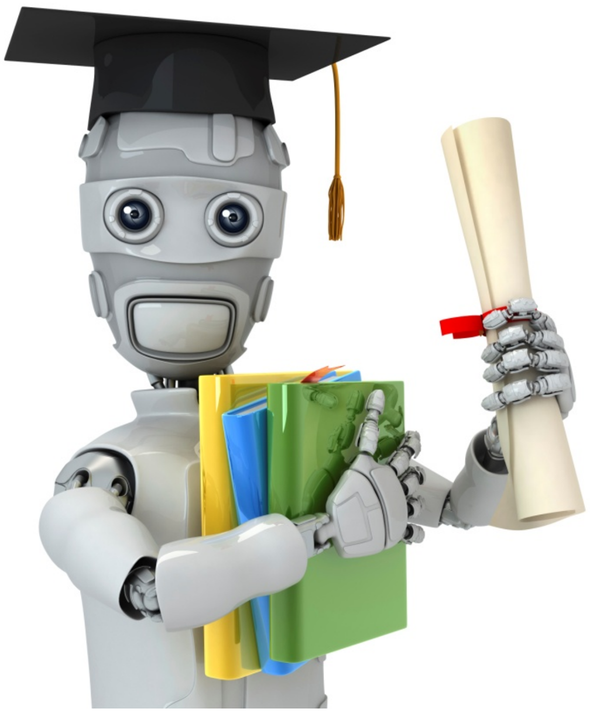
梯度下降更新参数值

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

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

预测用户j对电影i的评分

$(\theta^{(j)})^T (x^{(i)})$



Machine Learning

Recommender Systems

Vectorization: Low rank matrix factorization

低秩矩阵分解

Collaborative filtering

$\theta^{(j)}$

$n_m = 5$
$n_u = 4$

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

$x^{(i)}$

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

$y^{(i,j)}$

Y[i, j]

Andrew Ng

Collaborative filtering

$X\Theta^T$

$(\theta^{(j)})^T(x^{(i)})$

(i,j)

Predicted ratings:

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix} \qquad \begin{bmatrix} (\theta^{(1)})^T(x^{(1)}) & (\theta^{(2)})^T(x^{(1)}) & \dots & (\theta^{(n_u)})^T(x^{(1)}) \\ (\theta^{(1)})^T(x^{(2)}) & (\theta^{(2)})^T(x^{(2)}) & \dots & (\theta^{(n_u)})^T(x^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T(x^{(n_m)}) & (\theta^{(2)})^T(x^{(n_m)}) & \dots & (\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

$$X = \begin{bmatrix} — (x^{(1)})^T — \\ — (x^{(2)})^T — \\ \vdots \\ — (x^{(n_m)})^T — \end{bmatrix} \qquad \Theta = \begin{bmatrix} — (\theta^{(1)})^T — \\ — (\theta^{(2)})^T — \\ \vdots \\ — (\theta^{(n_u)})^T — \end{bmatrix}$$

Low rank matrix factorization

低秩矩阵分解

Andrew Ng

$$x^{(1)} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_n^{(1)} \end{bmatrix} \qquad (x^{(1)})^T = [x_1^{(1)} \; x_2^{(1)} \; \cdots \; x_n^{(1)}]$$

θ的转

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(n_m)})^T \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots \\ x_1^{(n_m)} & x_2^{(n_m)} & \cdots & x_n^{(n_m)} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} (\theta^{(1)})^T \\ (\theta^{(2)})^T \\ \vdots \\ (\theta^{(n_u)})^T \end{bmatrix}$$

Finding related movies 为了得到某电影的所有特征$x^{(i)}$

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

$\rightarrow$ x_1 = romance, x_2 = action, x_3 = comedy, x_4 = ...

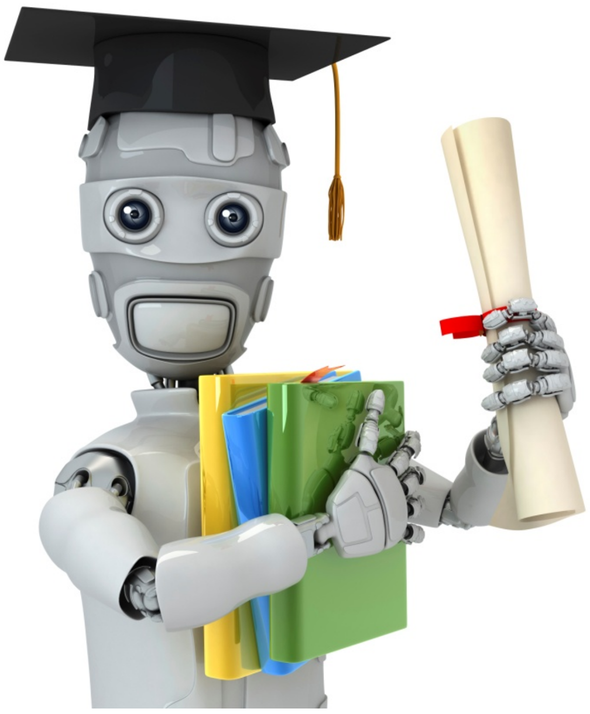
How to find movies j related to movie i? 找到电影j与电影i相关?

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

电影相似度=电影的特征向量距离

5 most similar movies to movie i:

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



Machine Learning

Recommender Systems

Implementational detail: Mean normalization

均值归一化

Users who have not rated any movies

某用户没有评分任何一部电影

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	Eve (5)
Love at last	5	5	0	0	?
Romance forever	5	?	?	0	?
Cute puppies of love	?	4	0	?	?
Nonstop car chases	0	0	5	4	?
Swords vs. karate	0	0	5	?	?

$$\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}$$

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix}$$

对于第5个用户(j=5)而言 r(i,5)全都=0 ⇒ $\sum_{(i,5):r(i,5)=1}$ 无相加项

更新$\theta^{(5)}$时的唯一项

取j=5

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

两个特征 n=2

$\theta^{(5)} \in \mathbb{R}^2$

$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

不含$\theta^{(5)}$项

$\frac{\lambda}{2}\left[(\theta_1^{(5)})^2 + (\theta_2^{(5)})^2\right]$

$(\theta^{(5)})^T x^{(i)} = 0$

求min，该项越小越好

无意义

Andrew Ng

Mean Normalization:

使得每部电影均分为0

$$
Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \begin{matrix} 2.5 \\ 2.5 \\ 2 \\ \vdots \\ \end{matrix} \qquad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \xrightarrow{Y-\mu} Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}
$$

For user j, on movie i predict:

预测 → $(\theta^{(j)})^T(x^{(i)}) + \mu_i$

学习参数 learn $\theta^{(j)}$, $x^{(i)}$

User 5 (Eve):

$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\underbrace{(\theta^{(5)})^T(x^{(i)})}_{\to 0} + \boxed{\mu_i} = \mu_i$ 不0

Andrew Ng

某部没评分的同一部电影 → 归一化行

某电影没有评分值 → 归一化列