## 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 $\theta^{(j)}$:

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

## Optimization algorithm:

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

## Gradient descent update:

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

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

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

$\frac{\partial}{\partial \theta_k^{(j)}} J(\theta^{(1)},\ldots,\theta^{(n_u)})$

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

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

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

Andrew Ng

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

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

- $m^{(j)}$ = no. of movies rated by user $j$

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

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

# Content-based recommender systems

$n_u = 4$, $n_m = 5$

$x_0 = 1$

| 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 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 |

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

$n = 2$

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.

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

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

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

Andrew Ng

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

$n = 2$

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.

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

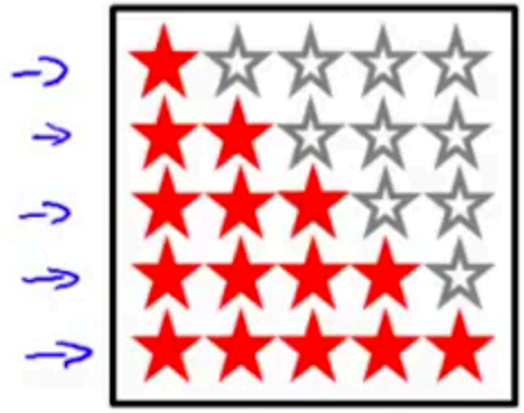
Andrew Ng

# 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 | ? | ? | 0 |
| Cute puppies of love | ? | 4 | 0 | ? |
| Nonstop car chases | 0 | 0 | 5 | 4 |
| Swords vs. karate | 0 | 0 | 5 | ? |

$n_u = 4$ $n_m = 5$



→ $n_u$ = no. users

→ $n_m$ = no. movies

$r(i,j)$ = 1 if user $j$ has rated movie $i$

$y^{(i,j)}$ = rating given by user $j$ to movie $i$ (defined only if $r(i,j) = 1$)

Andrew Ng

## Optimization algorithm:

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

Gradient descent update:

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

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

Andrew Ng