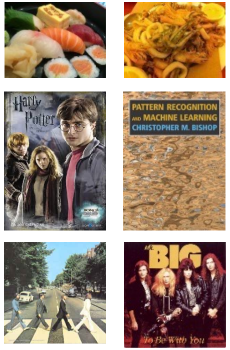
Recommendation Systems

Recommendation Systems

Problem formulation

Predicting movie ratings

User rates movies using zero to five stars



Training set

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

Nice

Boring

Recommendation Systems

Problem formulation

n_u = no. users = 4

n_m = no. movies = 5

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

$y^{(i,j)}$ = rating given by user j to movie i (defined only if $r(i,j)=1$) ϵ {0,..,5}

Predicting movie ratings

User rates movies using zero to five stars

Training set

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

Nice

Boring

In our notation, $r(i,j)=1$ if user j has rated movie i, and $y^{(i,j)}$ is his rating on that movie. Consider the following example (no. of movies $n_m=1$, no. of users $n_u=3$):

	User 1	User 2	User 3
Movie 1	0	1	?
Movie 2	?	5	5

What is $r(2,1)$? How about $y^{(2,1)}$?

- $r(2,1)=0,\ y^{(2,1)}=1$
- $r(2,1)=1,\ y^{(2,1)}=1$
- $r(2,1)=0,\ y^{(2,1)}=\text{undefined}$
- $r(2,1)=1,\ y^{(2,1)}=\text{undefined}$

$n_u = 4 \ , \ n_m = 5$

Content-based recommender systems

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	?

Content-based recommender systems

$n_u = 4$, $n_m = 5$

n=2

x_0=1

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

Suppose we have also a set of features for each movie (<u>related to the content!</u>):

x_1 measure the degree to be a romantic movie

x_2 measure the degree to be an action movie

"Love at last" as a degree of "romanticness" equal to 0.9 and it isn't an action movie

→ Each movie can be represented as feature vector $x^{(i)}=[x_0, x_1, x_2]^T=[1, x_1, x_2]^T$

→ $x^{(3)}$=[1,0.99,0] is the feature vector related to the film "Cute puppies of love"

Content-based recommender systems

$n_u = 4$, $n_m = 5$

n=2

$x_0=1$

Movie	Alice (1) $\theta^{(1)}$	Bob (2) $\theta^{(2)}$	Carol (3) $\theta^{(3)}$	Dave (4) $\theta^{(4)}$	x_1 (romance)	x_2 (action)
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

We can treat predicting the rating of each user as a linear regression problem.

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.

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

Content-based recommender systems

$n_u = 4$, $n_m = 5$

$n=2$

$\boxed{x_0=1}$

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

The main problem in this case is: How learn $\theta^{(1)} \dots \theta^{(4)}$?

Consider the following set of movie ratings:

Movie	Alice (1)	Bob (2)	Carol (3)	David (4)	(romance)	(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

Which of the following is a reasonable value for $\theta^{(3)}$? Recall that $x_0 = 1$.

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

Submit

Skip

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: $(\theta^{(j)})^T(x^{(i)})$ $\qquad \theta^{(j)} \in \mathbb{R}^{n+1}$

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

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

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

Optimization objective:

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

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

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

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

Optimization algorithm:

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

Gradient descent update:

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

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

Content Based Recommendation
Take Home Message

- We have features to capture content of the movies (e.g., given by analysis of the problem by expert or obtained through features extraction)
- We have some ratings (sparse information)
- We use regression on the feature to predict ratings. Parameters are learned considering the given ratings