16. Recommender Systems

Example: Predict movie ratings

User rates movies using zero to five stars.

Movie	Alice	Bob	Carol	Dave
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

$n_u = 4$

$n_m = 5$

n_u = # users

n_m = # 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 $r(i,j) = 1$)

Stars 0, ..., 5

Content based recommendation system:

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

$x_0 = 1$ (intercept term) $\rightarrow x^{(1)} = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}$ $n = 2$ # of features

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

Example for Alice → Cute puppies of love:

$$x^{(2)} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix} \quad \theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \quad (\theta^{(1)})^T \cdot x^{(2)} = 5 \cdot 0.99 = 4.95$$

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: $(\theta^{(j)})^T x^{(i)}$
$m^{(j)}$ = # of movies rated by user j

Linear Regr. Problem → To learn $\theta^{(j)}$:

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

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

Sum over movies which have been rated by user.

Squared error

Optimization objective:

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

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

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

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

Gradient descent update:

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

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

Problem motivation:

movie	Alice $\theta^{(1)}$	Bob $\theta^{(2)}$	Carol $\theta^{(3)}$	Dave $\theta^{(4)}$	x_1 Romance (assumed initially unknown)	x_2 action (assumed initially unknown)
$x^{(1)}$ 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	?	?	?

$x_0 = 1$

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

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

User informs about Preferences $\rightarrow \theta^{(j)}$

5 multiplies associated to x_2

for $x^{(1)}$:

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

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

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

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

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

$$\min_{x^{(i)}} \frac{1}{2} \underbrace{\sum_{j:r(i,j)=1}}_{\text{only for movies which have a rating}} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2$$

not to far in squared error sense from actual value

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

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

Collaborative filtering:

Given $x^{(1)}, \dots, x^{(n_m)}$ (and movie ratings), $\rightarrow r(i,j)$, $y^{(i,j)}$

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

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

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

randomly guess $\theta \xrightarrow{\text{optimize}} x \xrightarrow{\text{optimize}} \theta \rightarrow x \rightarrow \dots$ will converge to a reasonable value of $x^{(i)}$ and $\theta^{(j)}$.

Collaborative filtering optimization objective

~~$x_0 = 1$~~ ~~$\theta_0 = 1$~~ not used

$x \in \mathbb{R}^n$

$\theta \in \mathbb{R}^n$

use the algorithm $*^1$ and $*^2$ and use them together

$$J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}) =$$

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

↳ combined $*^1$ and $*^2$

Simultaneously minimize $\min_{x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}} J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)})$

Collaborative filtering algorithm:

1) Initialize $x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}$ to small random values
2) minimize $J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)})$ using gradient descent (or an advanced optimization algorithm) for every $j = 1, \dots, n_u$, $i = 1, \dots, n_m$

~~$x_0 = 1$~~ $x \in \mathbb{R}^n$ $\theta \in \mathbb{R}^n$

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

$\frac{\partial}{\partial x_k^{(i)}} J(\dots)$

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

~~θ_0~~ θ_1 $\dots$ θ_n

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

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

Collaborative filtering:

Movie	Alice	Bob	Carol	Dave
Love at last	5	5	∅	∅
Romance forever	5	?	?	∅
Cute puppies of L.	?	4	∅	?
Nonstop car chases	∅	∅	5	4
Swords vs. karate	∅	∅	5	?

predict

predict

$\Rightarrow y = \begin{bmatrix} 5 & 5 & \emptyset & \emptyset \\ 5 & ? & ? & \emptyset \\ ? & 4 & \emptyset & ? \\ \emptyset & \emptyset & 5 & 4 \\ \emptyset & \emptyset & 5 & ? \end{bmatrix}$

$y^{(i,j)}$

Predicted Ratings:

$$\begin{bmatrix} (\theta^{(1)})^T x^{(1)} & (\theta^{(2)})^T x^{(2)} & \cdots & (\theta^{(n_u)})^T x^{(1)} \\ (\theta^{(1)})^T x^{(2)} & (\theta^{(2)})^T x^{(2)} & \cdots & (\theta^{(n_u)})^T x^{(2)} \\ \vdots & \vdots & & \vdots \\ (\theta^{(1)})^T x^{(n_m)} & (\theta^{(2)})^T x^{(n_m)} & \cdots & (\theta^{(n_u)})^T x^{(n_m)} \end{bmatrix} \Rightarrow (\theta^{(j)})^T (x^{(i)})$$

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

Low rank matrix factorization

Find related movies:

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

x_1 = romance, x_2 = action, x_3 = comedy, x_4 = ...

How to find movies j related movie i?

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

5 most similar movies to movie i:
Find 5 movies j with the smallest $\|x^{(i)} - x^{(j)}\|$.

Users who have not rated any movies:

	Alice	Bob	Carol	Dave	Eve	
Love at last	5	5	∅	∅	?	∅
Romance forever	5	?	?	∅	?	∅
Cute puppies of l.	?	4	∅	?	?	∅
Nonstop car chases	∅	∅	5	4	?	∅
Swords vs. karate	∅	∅	5	?	?	∅ ← define initially ∅ not useful because *)

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

$$\min_{x^{(1)},\dots,x^{(n_m)} \atop \theta^{(1)},\dots,\theta^{(n_u)}} \underbrace{\frac{1}{2}\sum_{(i,j):r(i,j)=1}\left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2}_{\text{User has not rated any movie, therefore } r(i,j) \text{ is always } \emptyset} + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2 + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2}_{\text{only relevant}}$$

$$n = 2 \quad \theta^{(5)} \in \mathbb{R}^2 \Rightarrow \theta^{5} = \begin{bmatrix} \emptyset \\ \emptyset \end{bmatrix}$$

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

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

Mean normalization:

average rating: $(5+5+\emptyset+\emptyset)/4 = 2.5$

subtract average rating from Rating: $5-2.5$, $\emptyset-2.5$

$$Y = \begin{bmatrix} 5 & 5 & \emptyset & \emptyset & ? \\ 5 & ? & ? & \emptyset & ? \\ ? & 4 & \emptyset & ? & ? \\ \emptyset & \emptyset & 5 & 4 & ? \\ \emptyset & \emptyset & 5 & \emptyset & ? \end{bmatrix} \quad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \Rightarrow Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

$0-2.25$

Learn $\theta^{(j)}, x^{(i)}$ — pretended to be ratings of users

For user j, on movie i predict:

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

User 5 (Eve)

$$\theta^{(5)} = \begin{bmatrix} \emptyset \\ \emptyset \end{bmatrix} \Rightarrow \underbrace{(\theta^{(5)})^T (x^{(i)})}_{\emptyset} + \mu_i \Rightarrow \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix}$$

which is the average rating of the movies