# Machine Learning 12

Kihyun Shin
DMSE, HBNU

TheMAD

# Recommender systems
# - Collaborative filtering -

# Predicting movie ratings

## User rates movies using 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 | ? |

$y(i,j)$

$r(i,j) = 1$ $\quad$ $r(i,j) = 0$

$n_u = 4$ : **no. of users**

$n_m = 5$ : **no. of movies**

$r(i,j) = 1$ : **if user $j$ has rated movie $i$** $\quad r(1,1) = 1, \quad r(3,1) = 0$

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

$y^{(3,2)} = 4, \quad r(4,3) = 5$

# What if we have features of the movies?

| 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 |

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

$x^{(3)} = \begin{bmatrix} 0.99 \\ 0 \end{bmatrix}$

$n_u = 4$ : no. of users $n_m = 5$ : no. of movies $n = 2$ : no. of features

For user 1: Predict rating for movies $i$ as: $w^{(1)} \cdot x^{(i)} + b^{(1)}$ → Just linear regression

$w^{(1)} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ $b^{(1)} = 0$ $x^{(3)} = \begin{bmatrix} 0.99 \\ 0 \end{bmatrix}$ $\boldsymbol{w^{(1)} \cdot x^{(3)} + b^{(1)} = 4.95}$

For user $j$: Predict user $j$'s rating for movie $i$ as: $w^{(j)} \cdot x^{(i)} + b^{(j)}$



TheMAD

# Cost function

## Notation:

$r(i,j) = 1$ if user $j$ has rated movie $i$ (0 otherwise)

$y^{(i,j)} =$ rating given by user $j$ on movie $i$ (if defined)

$x^{(i)} =$ feature vector for movie $i$

For user $j$ and moive $i$, predict rating: $w^{(j)} \cdot x^{(i)} + b^{(j)}$

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

To learn $w^{(j)}, b^{(j)}$

No. of features

$$\min_{w^{(j)},b^{(j)}} J\left(w^{(j)}, b^{(j)}\right) = \frac{1}{2\cancel{m^{(j)}}} \sum_{i:r(i,j)=1} \left(w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2\cancel{m^{(j)}}} \sum_{k=1}^{n} \left(w_k^{(j)}\right)^2$$

Constant (can be removed)

# Cost function

**To learn parameter** $w^{(j)}, b^{(j)}$ **for user** $j$**:**

$$J\left(w^{(j)}, b^{(j)}\right) = \frac{1}{2} \sum_{i:r(i,j)=1} \left(w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(w_k^{(j)}\right)^2$$

**To learn parameter** $w^{(1)}, b^{(2)}, w^{(2)}, b^{(2)}, \dots, w^{(n_u)}, b^{(n_u)}$ **for all users:**

$$J\begin{pmatrix} w^{(1)}, \dots, w^{(n_u)} \\ b^{(1)}, \dots, b^{(n_u)} \end{pmatrix} = \frac{1}{2} \sum_{i:r(i,j)=1}^{n_u} \left(\underbrace{w^{(j)} \cdot x^{(i)} + b^{(j)}}_{f(x)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} \left(w_k^{(j)}\right)^2$$

# Problem motivation

When we don't have any information about features,

| Movie | Alice (1) | Bob (2) | Carol (3) | Dave (4) | $x_1$ (romance) | $x_2$ (action) |
|---|---|---|---|---|---|---|
| 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 | ? | ? | ? |

We still can predict them based on user's ratings to each movies.

Using $w^{(j)} \cdot x^{(i)} + \cancel{b^{(j)}}$

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

$$b^{(1)} = 0 \quad b^{(2)} = 0 \quad b^{(3)} = 0 \quad b^{(4)} = 0$$

$$\left.\begin{aligned} w^{(1)} \cdot x^{(1)} &\approx 5 \\ w^{(2)} \cdot x^{(1)} &\approx 5 \\ w^{(3)} \cdot x^{(1)} &\approx 0 \\ w^{(4)} \cdot x^{(1)} &\approx 0 \end{aligned}\right\} x^{(1)} = \begin{bmatrix}1\\0\end{bmatrix}$$

# Cost function

**Given** $w^{(1)}, b^{(1)}, w^{(2)}, b^{(2)}, \ldots, w^{(n_u)}, b^{(n_u)}$

**To learn** $x^{(i)}$:

$$J\left(x^{(i)}\right) = \frac{1}{2} \sum_{j:r(i,j)=1} \left(w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^{(i)}\right)^2$$

**To learn** $x^{(1)}, x^{(2)}, \ldots, x^{(n_m)}$:

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

# Collaborative filtering

| | | $j=1$ | $j=2$ | $j=3$ |
|---|---|---|---|---|
| | | **Alice** | **Bob** | **Carol** |
| $i=1$ | **Movie 1** | 5 | 5 | ? |
| $i=2$ | **Movie 2** | ? | 2 | 3 |

**Cost function to learn** $w^{(1)}, b^{(2)}, w^{(2)}, b^{(2)}, \dots, w^{(n_u)}, b^{(n_u)}$ **:**

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

**Cost function to learn** $x^{(1)}, x^{(2)}, \dots, x^{(n_m)}$**:**

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

**Put them together:**

$$\min_{\substack{w^{(1)},\dots,w^{(n_m)}\\ b^{(1)},\dots,b^{(n_m)}\\ x^{(1)},\dots,x^{(n_m)}}} J(w,b,x)=\frac{1}{2}\sum_{(i,j):r(i,j)=1}\left(w^{(j)}\cdot x^{(i)}+b^{(j)}-y^{(i,j)}\right)^2+\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(w_k^{(j)}\right)^2+\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

# Gradient descent

## Linear regression

Repeat {

$$w_i = w_i - \alpha \frac{\partial}{\partial w_i} J(w, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

}

$$w_i^{(j)} = w_i^{(j)} - \alpha \frac{\partial}{\partial w_i^{(j)}} J(w, b, x)$$

$$b^{(j)} = b^{(j)} - \alpha \frac{\partial}{\partial b^{(j)}} J(w, b, x)$$

$$x_k^{(i)} = b^{(j)} - \alpha \frac{\partial}{\partial b^{(j)}} J(w, b, x)$$

$x$ is a also parameter for collaborative filtering

# Binary labels

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

# Example applications

1. Did user $j$ purchase an item after being shown ?

2. Did user $j$ fav/like an item ?

3. Did user $j$ spend at least 30 sec with an item?

4. Did user $j$ click on an item ?

Meaning of ratings:

1 - engaged after being shown item

0 - did not engage after being shown item

? - item not yet shown

# From regression to binary classification

**Previously:**

Predict $y^{(i,j)}$ as $w^{(j)} \cdot x^{(i)} + b^{(j)}$

**For binary labels:**

Predict that the probability of $y^{(i,j)} = 1$

is given by $\mathrm{g}(w^{(j)} \cdot x^{(i)} + b^{(j)})$

where $\mathrm{g}(\mathrm{z}) = \frac{1}{1+e^{-z}}$

# Cost function for binary application

## Previous cost function:

$$\frac{1}{2}\sum_{(i,j):r(i,j)=1}\left(w^{(j)}\cdot x^{(i)}+b^{(j)}-y^{(i,j)}\right)^2+\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(w_k^{(j)}\right)^2+\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

**Loss for binary labels:** $y^{(i,j)}: f_{(w,b,x)}(x) = g(w^{(j)}\cdot x^{(i)}+b^{(j)})$

$$L\left(f_{(w,b,x)}(x), y^{(i,j)}\right) = -y^{(i,j)}\log\left(f_{(w,b,x)}(x)\right) - (1-y^{(i,j)})\log(1-f_{(w,b,x)}(x))$$

$$J(w,b,x) = \sum_{(i,j):r(i,j)=1} L\left(f_{(w,b,x)}(x), y^{(i,j)}\right)$$

$$\searrow g(w^{(j)}\cdot x^{(i)}+b^{(j)})$$


TheMAD

# Users who have not rated any movies

| 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 | ? | ? |

$$\min_{\substack{w^{(1)},\dots,w^{(n_m)}\\ b^{(1)},\dots,b^{(n_m)}\\ x^{(1)},\dots,x^{(n_m)}}} J(w,b,x) = \frac{1}{2}\sum_{(i,j):r(i,j)=1}\left(w^{(j)}\cdot x^{(i)} + b^{(j)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(w_k^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

$$w^{(5)} = \begin{bmatrix}0\\0\end{bmatrix} \qquad b^{(5)} = 0 \qquad w^{(5)}\cdot x^{(i)} + b^{(5)} = 0$$

**First, suggest w, b as 0, and then Eve's rating became all 0** →

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

# Mean Normalization

$$\begin{bmatrix} 5 & 5 & 0 & 0 & 0 \\ 5 & ? & ? & 0 & 0 \\ ? & 4 & 0 & ? & 0 \\ 0 & 0 & 5 & 4 & 0 \\ 0 & 0 & 5 & ? & 0 \end{bmatrix} \qquad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}}_{= y_{i,j}} \begin{matrix} = 2.5 \\ = 2.5 \\ = 2 \\ = 2.25 \\ = 1.25 \end{matrix}$$

**For user $j$, on movie $i$ predict:**

$$w^{(j)} \cdot x^{(i)} + b^{(j)} + \mu_i$$

**User 5 (Eve):** **(recalculate ratings based on average)**

$$w^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad b^{(5)} = 0 \qquad \underbrace{w^{(5)} \cdot x^{(1)} + b^{(5)}}_{= 0} + \mu_1 = 2.5$$

**Both row (new movie) and column (new user) are available**

# Summary

- Recommender systems predict user preferences based on past ratings or interactions.
- If item features are available, rating prediction can be treated as a linear regression problem.
- Collaborative filtering learns both user preferences and item features from rating data.
- For binary labels, recommender systems predict engagement probability using logistic regression.
- Mean normalization helps handle new users or items with little/no rating history.