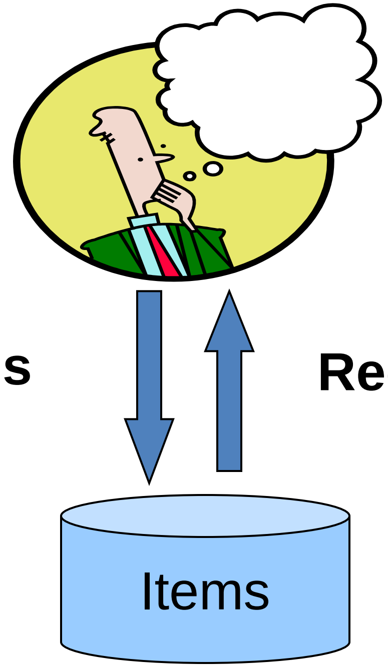
# Recommender Systems

# Motivation


Aplicaciones en Internet

# Recommender systems

Searches
Downloads
Ratings




Recommendations

Products, webs, news, videos, …

**Examples:**

amazon.com
PANDORA
StumbleUpon
del.icio.us
NETFLIX
movielens
helping you find the *right* movies
last.fm
the social music revolution
Google News
YouTube
XBOX LIVE

## From scarcity to abundance

Exhibition space is a scarce resource in traditional commerce

- So is the weather on TV, trailers in the movies, etc.

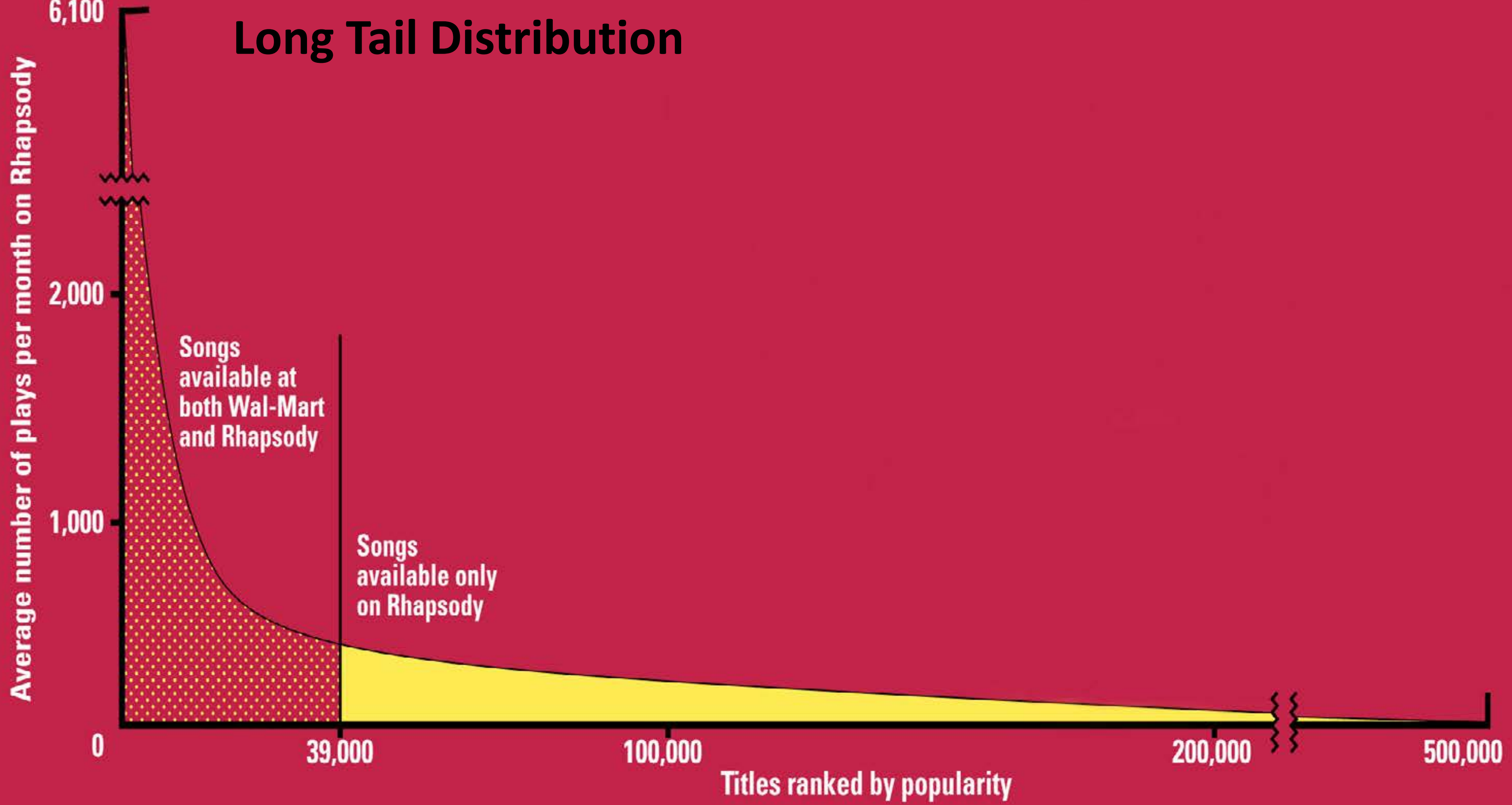
The website allows the dissemination of product information at almost no cost

- From scarcity to abundance

More options require better filters

- Recommendation engines (Recommendation engines)

# Long Tail Distribution



## Types of recommendations

### Manual

- Favourite lists
- Prepared by critics and experts

### Simple aggregators

- Top 10, Most Popular, Most Recent

### Adjusted to the interests of each user

- Amazon, Netflix,....

# Example: Predicting movie ratings

User rates movies using one to five stars

| Movie | Alice (1) | Bob (2) | Carol (3) | Dave (4) |
|---|---|---|---|---|
| Pretty Woman | | | | |
| Titanic | | | | |
| Algo para Recordar | | | | |
| La Jungla de Cristal | | | | |
| Skyfall | | | | |

$n_u$ = 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 $r(i,j) = 1$)

In our notation, r (i, j) = 1 if the user j has scored the movie i, and y(i, j) is his score for that movie. Consider the following example (number of movies $n_m$ = 2, number of users $n_u$ = 3):

| | Usuario 1 | Usuario 2 | Usuario 3 |
|---|---|---|---|
| Movie 1 | 0 | 1 | ? |
| Movie 2 | ? | 5 | 5 |

What are the values for $r(2,1)$ and $y^{(2,1)}$?

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


# Recommender Systems

## Content-based recommendations

# Content-based recommender systems

| Movie | Alicia (1) | Paco (2) | Elena (3) | Pepe (4) | $x_1$ (romance) | $x_2$ (action) |
|---|---|---|---|---|---|---|
| Pretty Woman | 5 | 5 | 0 | 0 | 0.9 | 0 |
| Titanic | 5 | ? | ? | 0 | 1.0 | 0.2 |
| Algo para Recordar | ? | 4 | 0 | ? | 1.0 | 0 |
| La Jungla de Cristal | 0 | 0 | 5 | 4 | 0 | 1.0 |
| Skyfall | 0 | 0 | 5 | ? | 0.1 | 0.9 |

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.

Consider the ratings:

| Movie | Alicia (1) | Paco (2) | Elena (3) | Pepe (4) | $x_1$ (romance) | $x_2$ (action) |
|---|---|---|---|---|---|---|
| Pretty Woman | 5 | 5 | 0 | 0 | 0.9 | 0 |
| Titanic | 5 | ? | ? | 0 | 1.0 | 0.2 |
| Algo para Recordar | ? | 4 | 0 | ? | 1.0 | 0 |
| La Jungla de Cristal | 0 | 0 | 5 | 4 | 0 | 1.0 |
| Skyfall | 0 | 0 | 5 | ? | 0.1 | 0.9 |

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

- ❑ $\theta^{(3)}$ = *[0; 5; 0]*
- ❑ $\theta^{(3)}$ = *[1; 0; 4]*
- ❑ $\theta^{(3)}$ = *[0; 0; 1]*
- ❑ $\theta^{(3)}$ = *[0; 0; 5]*

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

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

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

**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:

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

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

# Recommender Systems

## Colaborative Filtering


Internet Applications

# Problem motivation

| Movie | Alice (1) | Bob (2) | Carol (3) | Dave (4) | $x_1$ (romance) | $x_2$ (action) |
|---|---|---|---|---|---|---|
| Pretty Woman | 5 | 5 | 0 | 0 | 0.9 | 0 |
| Titanic | 5 | ? | ? | 0 | 1.0 | 0.2 |
| Algo para Recordar | ? | 4 | 0 | ? | 1.0 | 0 |
| La Jungla de Cristal | 0 | 0 | 5 | 4 | 0 | 1.0 |
| Skyfall | 0 | 0 | 5 | ? | 0.1 | 0.9 |

# Problem motivation

| Movie | Alice (1) | Bob (2) | Carol (3) | Dave (4) | $x_1$ (romance) | $x_2$ (action) |
|---|---|---|---|---|---|---|
| Pretty Woman | 5 | 5 | 0 | 0 | 0.9 | 0 |
| Titanic | 5 | ? | ? | 0 | 1.0 | 0.2 |
| Algo para Recordar | ? | 4 | 0 | ? | 1.0 | 0 |
| La Jungla de Cristal | 0 | 0 | 5 | 4 | 0 | 1.0 |
| Skyfall | 0 | 0 | 5 | ? | 0.1 | 0.9 |

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

# Optimization algorithm

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

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

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

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

## Collaborative filtering

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

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

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

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

Suppose we use gradient descent to minimize

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

Which of the following is a correct gradient update rule for i≠0?

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

# Recommender Systems

## Collaborative filtering algorithm


Internet Applications

## Collaborative filtering optimization objective

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

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

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

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

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

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

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

**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). E.g. for every $j=1,\dots,n_u, i=1,\dots,n_m$ :

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

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

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

Internet Applications

# Recommender Systems

## Vectorization: Low rank matrix factorization

# Collaborative filtering

| Movie | Alice (1) | Bob (2) | Carol (3) | Dave (4) |
|---|---|---|---|---|
| Pretty Woman | 5 | 5 | 0 | 0 |
| Titanic | 5 | ? | ? | 0 |
| Algo para Recordar | ? | 4 | 0 | ? |
| La Jungla de Cristal | 0 | 0 | 5 | 4 |
| Skyfall | 0 | 0 | 5 | ? |

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

## Collaborative filtering

Predicted ratings:

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

$$\begin{bmatrix} (\theta^{(1)})^T(x^{(1)}) & (\theta^{(2)})^T(x^{(1)}) & \dots & (\theta^{(n_u)})^T(x^{(1)}) \\ (\theta^{(1)})^T(x^{(2)}) & (\theta^{(2)})^T(x^{(2)}) & \dots & (\theta^{(n_u)})^T(x^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T(x^{(n_m)}) & (\theta^{(2)})^T(x^{(n_m)}) & \dots & (\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

Define

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

How can we compactly express

$$\begin{bmatrix} (x^{(1)})^T(\theta^{(1)}) & \ldots & (x^{(1)})^T(\theta^{(n_u)}) \\ \vdots & \ddots & \vdots \\ (x^{(n_m)})^T(\theta^{(1)}) & \ldots & (x^{(n_m)})^T(\theta^{(n_u)}) \end{bmatrix}$$

- ❑ $X\Theta$
- ❑ $X\Theta^T$
- ❑ $X^T\Theta$
- ❑ $\Theta^T X^T$

## Finding related movies

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

How to find movies $j$ related to movie $i$?

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


Aplicaciones en Internet

# Recommender Systems

## Implementation detail: Mean normalization

# Users who have not rated any movies

| Movie | Alice (1) | Bob (2) | Carol (3) | Dave (4) | Eve (5) |
| --- | --- | --- | --- | --- | --- |
| Pretty Woman | 5 | 5 | 0 | 0 | ? |
| Titanic | 5 | ? | ? | 0 | ? |
| Algo para Recordar | ? | 4 | 0 | ? | ? |
| La Jungla de Cristal | 0 | 0 | 5 | 4 | ? |
| Skyfall | 0 | 0 | 5 | ? | ? |

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

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

# Mean Normalization:

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

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

User 5 (Eve):

We have talked about normalisation on average. However, we do not apply feature scaling, that is, we have not scaled the scores by dividing them by the range (max - min). This is because

- This type of scaling is not useful when the value to predict is a real number.
- Movie scores are already comparable (e. g. from 0 to 5 stars), so they are already on the same scale.
- Subtracting the mean is mathematically equivalent to dividing by rank.
- In this way the algorithm is more computationally.

**Inclusion of biases:**

Motivation:

- Some users tend to give higher scores and others tend to give lower scores: user bias.
- Some movies also tend to receive better ratings than others: biased movies.

We can introduce these biases into our formulation:

- User bias j: $b_j$
- Movie i bias: $b_i$

# Inclusion of biases:

Prediction without biases: $\hat{y}^{(i,j)} = \mu_i + \left(\theta^{(j)}\right)^T x^{(i)}$

Prediction including biases: $\hat{y}^{(i,j)} = \mu_i + b_j + b_i + \left(\theta^{(j)}\right)^T x^{(i)}$

Cost function without biases:

$$J(\Theta, X) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\mu_i + \left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \left\|\theta^{(j)}\right\|^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \left\|x^{(i)}\right\|^2$$

Cost function including biases:

$$J\left(\Theta, X, \{b_j\}, \{b_i\}\right) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\mu_i + b_j + b_i + \left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)}\right)^2$$
$$+ \frac{\lambda}{2} \sum_{j=1}^{n_u} \left\|\theta^{(j)}\right\|^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \left\|x^{(i)}\right\|^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} {b_j}^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} {b_i}^2$$