MVA - Discrete Inference and Learning
Lecture 8
-
Recommender Systems

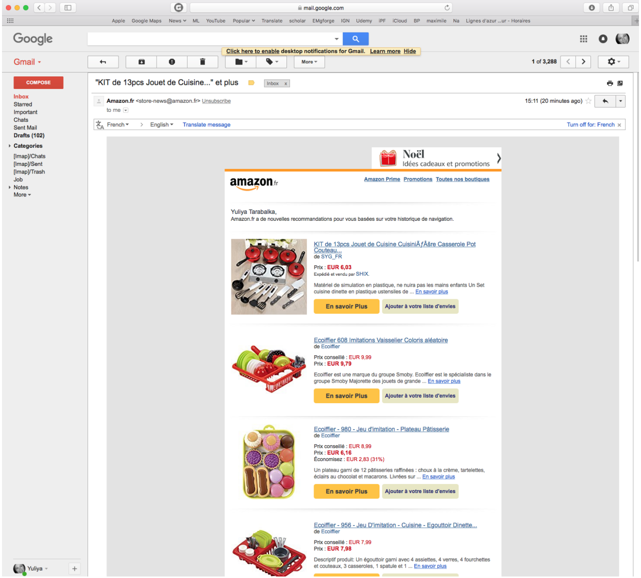
Yuliya Tarabalka
Inria Sophia Antipolis-Méditerranée - TITANE team
Université Côte d'Azur - France

Overview

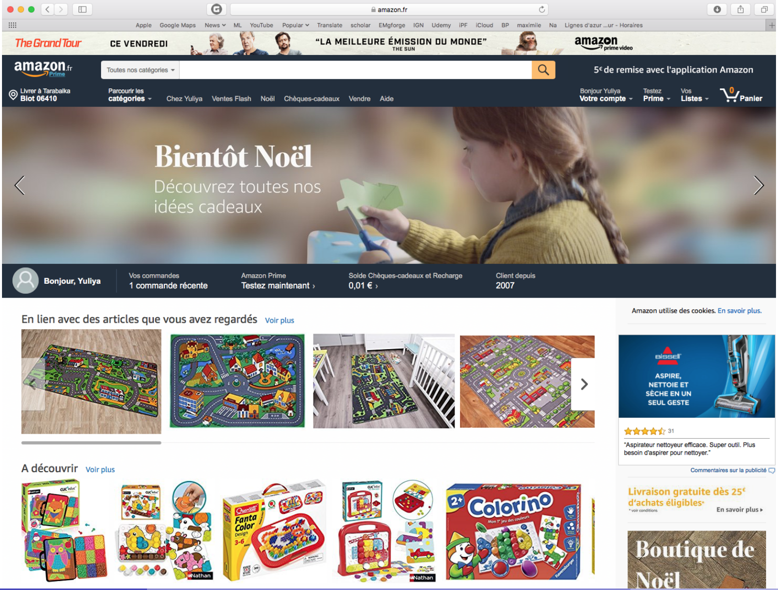
1. Problem formulation
2. Content-based recommendations
3. Collaborative filtering

Overview

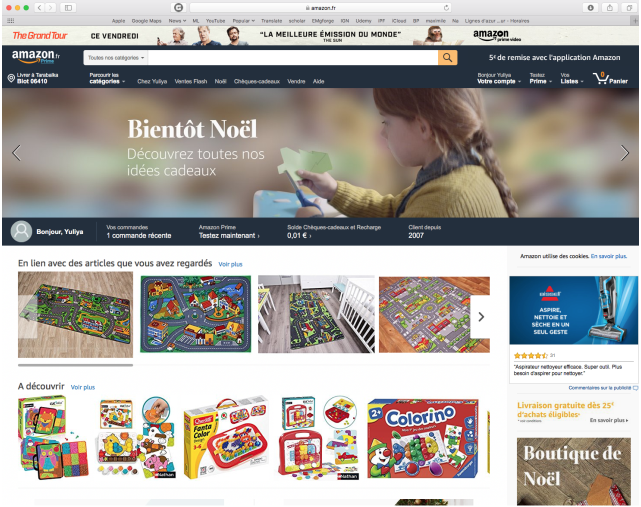
Motivation


mail.google.com
Google
Gmail
COMPOSE
Inbox
Starred
Important
Chats
Sent Mail
Drafts (102)
Categories
[Imap]/Chats
[Imap]/Sent
[Imap]/Trash
Job
Notes
More
Click here to enable desktop notifications for Gmail. Learn more Hide
More
1 of 3,288
"KIT de 13pcs Jouet de Cuisine..." et plus
Inbox
Amazon.fr <store-news@amazon.fr> Unsubscribe
to me
15:11 (20 minutes ago)
French English Translate message
Turn off for: French
Noël
Idées cadeaux et promotions
amazon.fr
Amazon Prime Promotions Toutes nos boutiques
Yuliya Tarabalka,
Amazon.fr a de nouvelles recommandations pour vous basées sur votre historique de navigation.
KIT de 13pcs Jouet de Cuisine CuisiniÃƒÂ¨re Casserole Pot Couteau...
de SYG_FR
Prix : EUR 6,03
Expédié et vendu par SHIX.
Matériel de simulation en plastique, ne nuira pas les mains enfants Un Set cuisine dinette en plastique ustensiles de ... En savoir plus
En savoir Plus
Ajouter à votre liste d'envies
Ecoiffier 608 Imitations Vaisselier Coloris aléatoire
de Ecoiffier
Prix conseillé : EUR 9,99
Prix : EUR 9,79
Ecoiffier est une marque du groupe Smoby. Ecoiffier est le spécialiste dans le groupe Smoby Majorette des jouets de grande ... En savoir plus
En savoir Plus
Ajouter à votre liste d'envies
Ecoiffier - 980 - Jeu d'imitation - Plateau Pâtisserie
de Ecoiffier
Prix conseillé : EUR 8,99
Prix : EUR 6,16
Économisez : EUR 2,83 (31%)
Un plateau garni de 12 pâtisseries raffinées : choux à la crème, tartelettes, éclairs au chocolat et macarons. Livrées sur ... En savoir plus
En savoir Plus
Ajouter à votre liste d'envies
Ecoiffier - 956 - Jeu D'imitation - Cuisine - Egouttoir Dinette...
de Ecoiffier
Prix conseillé : EUR 7,99
Prix : EUR 7,98
Descriptif produit: Un égouttoir garni avec 4 assiettes, 4 verres, 4 fourchettes et couteaux, 3 casseroles, 1 spatule et 1 ... En savoir plus
Yuliya

Motivation


amazon.fr
The Grand Tour
CE VENDREDI
"LA MEILLEURE ÉMISSION DU MONDE"
THE SUN
amazon prime video
Toutes nos catégories
5€ de remise avec l'application Amazon
Livrer à Tarabalka Biot 06410
Parcourir les catégories
Chez Yuliya
Ventes Flash
Noël
Chèques-cadeaux
Vendre
Aide
Bonjour Yuliya Votre compte
Testez Prime
Vos Listes
Panier
Bientôt Noël
Découvrez toutes nos idées cadeaux
Bonjour, Yuliya
Vos commandes 1 commande récente
Amazon Prime Testez maintenant ›
Solde Chèques-cadeaux et Recharge 0,01 € ›
Client depuis 2007
En lien avec des articles que vous avez regardés
Voir plus
Amazon utilise des cookies. En savoir plus.
BISSELL
ASPIRE, NETTOIE ET SÈCHE EN UN SEUL GESTE
"Aspirateur nettoyeur efficace. Super outil. Plus besoin d'aspirer pour nettoyer."
Commentaires sur la publicité
A découvrir
Voir plus
Livraison gratuite dès 25€ d'achats éligibles
En savoir plus ›
Colorino
Boutique de Noël

Personalized content



Adapt to general popularity pick based on user preferences

A more formal view

- User (requests content)
- Objects (that can be displayed)
- Context (device, location, time)
- Interface (mobile browser, tablet, viewport)

$\Downarrow$

Objective: recommend relevant objects

Challenges

- Scalability
    - Millions of objects
    - 100s of millions of users
- Cold start
    - Changing use base
    - Changing inventory (movies, stories, goods)
- Imbalanced dataset

Example: Predicting movie ratings

User rates movies using zero to five stars

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

Example: Predicting movie ratings

User rates movies using zero to five stars $\star$

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	?

Example: Predicting movie ratings

User rates movies using zero to five stars $\star$

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 = number of users
n_m = number of 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 $r(i, j) = 1$)

Example: Predicting movie ratings

User rates movies using zero to five stars $\star$

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 = number of users
n_m = number of 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 $r(i,j) = 1$)

Goal: replace ? by ratings

Overview

1. Problem formulation
2. Content-based recommendations
3. Collaborative filtering

Content-based recommender systems

How to predict ? ?

Content-based recommender systems

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0		
Romance for.	5	?	?	0		
Cute pup.of l.	?	4	0	?		
Nonst.car ch.	0	0	5	4		
Swords vs.kar.	0	0	5	?		

Content-based recommender systems

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	0.9	0
Romance for.	5	?	?	0	1.0	0.01
Cute pup.of l.	?	4	0	?	0.99	0
Nonst.car ch.	0	0	5	4	0.1	1.0
Swords vs.kar.	0	0	5	?	0	0.9

Content-based recommender systems

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	0.9	0
Romance for.	5	?	?	0	1.0	0.01
Cute pup.of l.	?	4	0	?	0.99	0
Nonst.car ch.	0	0	5	4	0.1	1.0
Swords vs.kar.	0	0	5	?	0	0.9

$$x^{(1)} = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}, ..., x^{(5)} = \begin{bmatrix} 1 \\ 0 \\ 0.9 \end{bmatrix}$$

Content-based recommender systems

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	0.9	0
Romance for.	5	?	?	0	1.0	0.01
Cute pup.of l.	?	4	0	?	0.99	0
Nonst.car ch.	0	0	5	4	0.1	1.0
Swords vs.kar.	0	0	5	?	0	0.9

$$x^{(1)} = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}, \ldots, x^{(5)} = \begin{bmatrix} 1 \\ 0 \\ 0.9 \end{bmatrix}$$

- For each user j, learn a parameter $\theta^{(j)} \in \mathbb{R}^3$
  - Linear regression problem

Content-based recommender systems

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	0.9	0
Romance for.	5	?	?	0	1.0	0.01
Cute pup.of l.	?	4	0	?	0.99	0
Nonst.car ch.	0	0	5	4	0.1	1.0
Swords vs.kar.	0	0	5	?	0	0.9

- For each user j, learn a parameter $\theta^{(j)} \in \mathbb{R}^3$
  - Linear regression problem
- Predict user j as rating movie i with $(\theta^{(j)})^T x^{(i)}$ stars

Content-based recommender systems

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	0.9	0
Romance for.	5	?	?	0	1.0	0.01
Cute pup.of l.	?	4	0	?	0.99	0
Nonst.car ch.	0	0	5	4	0.1	1.0
Swords vs.kar.	0	0	5	?	0	0.9

- Predict user j as rating movie i with $(\theta^{(j)})^T x^{(i)}$ stars

$$x^{(3)} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix} \longleftrightarrow \theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \qquad (\theta^{(1)})^T x^{(3)} = 5 \times 0.99 = 4.95$$

Problem formulation

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

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

$\theta^{(j)} =$ parameter vector for user j

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

For user j, movie i, predicted rating: $(\theta^{(j)})^T(x^{(i)})$

$m^{(j)} =$ number of movies rated by user j

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

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

Problem formulation

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

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

$\theta^{(j)} =$ parameter vector for user j

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

For user j, movie i, predicted rating: $(\theta^{(j)})^T(x^{(i)})$

$m^{(j)} =$ number of movies rated by user j

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

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

Problem formulation

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

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

$\theta^{(j)} =$ parameter vector for user j

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

For user j, movie i, predicted rating: $(\theta^{(j)})^T(x^{(i)})$

$m^{(j)} =$ number of movies rated by user j

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

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

Optimization objective

To learn $\theta^{(j)} \in \mathbb{R}^{n+1}$ (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 $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$:

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

Optimization algorithm

Optimization objective $J(\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)})$:

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

Gradient descent update:
For $k = 0$:

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

For $k \neq 0$:

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

Optimization algorithm

Optimization objective $J(\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)})$:

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

Gradient descent update:
For $k \neq 0$:

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

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

Optimization algorithm

Optimization objective $J(\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)})$:

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

Gradient descent update:

For $k = 0$:

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

For $k \neq 0$:

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

Optimization algorithm

Optimization objective $J(\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)})$:

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

One can also use more advanced optimization algorithm to optimize this objective function

- Ex: stochastic gradient descent

Optimization algorithm

Where to get / How to estimate features $x^{(i)}$?

Overview

1. Problem formulation
2. Content-based recommendations
3. Collaborative filtering

Collaborative filtering - Problem motivation

Movie	Alice(1)	Bob(2)	Carol(3)	Dave(4)	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	0.9	0
Romance for.	5	?	?	0	1.0	0.01
Cute pup.of l.	?	4	0	?	0.99	0
Nonst.car ch.	0	0	5	4	0.1	1.0
Swords vs.kar.	0	0	5	?	0	0.9

- In most cases, we want much more than 2 features for each movie

Problem motivation

Movie	Alice(1) $\theta^{(1)}$	Bob(2) $\theta^{(2)}$	Carol(3) $\theta^{(3)}$	Dave(4) $\theta^{(4)}$	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	?	?
Romance for.	5	?	?	0	?	?
Cute pup.of l.	?	4	0	?	?	?
Nonst.car ch.	0	0	5	4	?	?
Swords vs.kar.	0	0	5	?	?	?

Problem motivation

Movie	Alice(1) $\theta^{(1)}$	Bob(2) $\theta^{(2)}$	Carol(3) $\theta^{(3)}$	Dave(4) $\theta^{(4)}$	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	?	?
Romance for.	5	?	?	0	?	?
Cute pup.of l.	?	4	0	?	?	?
Nonst.car ch.	0	0	5	4	?	?
Swords vs.kar.	0	0	5	?	?	?

- Suppose users told us how much they like romantic & action movies

Problem motivation

Movie	Alice(1) $\theta^{(1)}$	Bob(2) $\theta^{(2)}$	Carol(3) $\theta^{(3)}$	Dave(4) $\theta^{(4)}$	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	?	?
Romance for.	5	?	?	0	?	?
Cute pup.of l.	?	4	0	?	?	?
Nonst.car ch.	0	0	5	4	?	?
Swords vs.kar.	0	0	5	?	?	?

- Suppose users told us how much they like romantic & action movies

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

Problem motivation

Movie	Alice(1) $\theta^{(1)}$	Bob(2) $\theta^{(2)}$	Carol(3) $\theta^{(3)}$	Dave(4) $\theta^{(4)}$	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	?	?
Romance for.	5	?	?	0	?	?
Cute pup.of l.	?	4	0	?	?	?
Nonst.car ch.	0	0	5	4	?	?
Swords vs.kar.	0	0	5	?	?	?

- Suppose users told us how much they like romantic & action movies

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

- We can then infer x_1 and x_2 for each movie

Problem motivation

Movie	Alice(1) $\theta^{(1)}$	Bob(2) $\theta^{(2)}$	Carol(3) $\theta^{(3)}$	Dave(4) $\theta^{(4)}$	x_1 (roman.)	x_2 (act)
Love at last	5	5	0	0	?	?
Romance for.	5	?	?	0	?	?
Cute pup.of l.	?	4	0	?	?	?
Nonst.car ch.	0	0	5	4	?	?
Swords vs.kar.	0	0	5	?	?	?

- Suppose users told us how much they like romantic & action movies

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

- We can then infer x_1 and x_2 for each movie
  - Ex: $(\theta^{(1)})^T x^{(1)} \approx 5, \ldots \Rightarrow x^{(1)} = [1 \quad 1.0 \quad 0.0]^T$

Optimization algorithm

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$, to learn $x^{(i)}$:

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

Optimization algorithm

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$, to learn $x^{(i)}$:

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

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$, to learn $x^{(1)}, ..., x^{(n_m)}$:

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

Collaborative filtering

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

can estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$,

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

Collaborative filtering

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

can estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$,

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

Collaborative filtering:

Guess θ

Collaborative filtering

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

can estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$,

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

Collaborative filtering:

Guess $\theta \Rightarrow x$

Collaborative filtering

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

can estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$,

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

Collaborative filtering:

Guess $\theta \Rightarrow x \Rightarrow \theta$

Collaborative filtering

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

can estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$,

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

Collaborative filtering:

Guess $\theta \Rightarrow x \Rightarrow \theta \Rightarrow x \Rightarrow \theta \Rightarrow x \Rightarrow ...$

Collaborative filtering optimization objective

Given $x^{(1)}, ..., x^{(n_m)}$, estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$:

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

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$, estimate $x^{(1)}, ..., x^{(n_m)}$:

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

Collaborative filtering optimization objective

Given $x^{(1)}, ..., x^{(n_m)}$, estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$:

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

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$, estimate $x^{(1)}, ..., x^{(n_m)}$:

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

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

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

Collaborative filtering optimization objective

Given $x^{(1)}, ..., x^{(n_m)}$, estimate $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$:

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

Given $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n_u)}$, estimate $x^{(1)}, ..., x^{(n_m)}$:

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

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

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

Collaborative filtering optimization objective

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

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

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

Collaborative filtering algorithm

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

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

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

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

Vectorization

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	?

Vectorization

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	?

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

Vectorization

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

Predicted ratings:

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

Vectorization

Predicted ratings:

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

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

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

Vectorization

Predicted ratings:

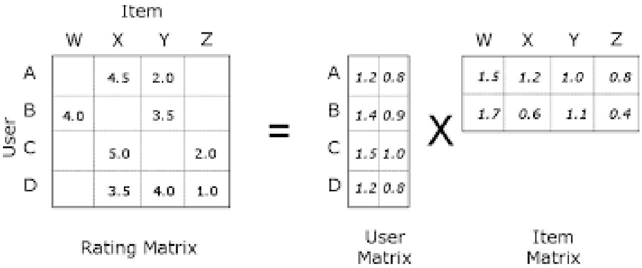
$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & ? \end{bmatrix} \qquad \begin{bmatrix} (\theta^{(1)})^T x^{(1)} & (\theta^{(2)})^T x^{(1)} & \dots & (\theta^{(n_u)})^T x^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T x^{(n_m)} & (\theta^{(2)})^T x^{(n_m)} & \dots & (\theta^{(n_u)})^T x^{(n_m)} \end{bmatrix}$$

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

$X\Theta^T$ is a low rank matrix

- Low rank matrix factorization

Low rank matrix factorization



Finding related movies

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

Finding related movies

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

- E.g. x_1 = romance, x_2 = action, x_3 = comedy, ...
  (capture the most important features/properties of movies)

Finding related movies

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

- E.g. x_1 = romance, x_2 = action, x_3 = comedy, ...
  (capture the most important features/properties of movies)

How to find movies j related to movie i?

Finding related movies

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

- E.g. x_1 = romance, x_2 = action, x_3 = comedy, ...
  (capture the most important features/properties of movies)

How to find movies j related to movie i?

- Small $||x^{(i)} - x^{(j)}|| \Rightarrow$ movies j and i are similar

Finding related movies

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

- E.g. x_1 = romance, x_2 = action, x_3 = comedy, ...
  (capture the most important features/properties of movies)

How to find movies j related to movie i?

- Small $||x^{(i)} - x^{(j)}|| \Rightarrow$ movies j and i are similar

5 most similar movies to movie i:

Finding related movies

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

- E.g. x_1 = romance, x_2 = action, x_3 = comedy, ...
  (capture the most important features/properties of movies)

How to find movies j related to movie i?

- Small $||x^{(i)} - x^{(j)}|| \Rightarrow$ movies j and i are similar

5 most similar movies to movie i:

- Find the 5 movies with the smallest $||x^{(i)} - x^{(j)}||$

Mean normalization

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	?	?

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

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

$n = 2$ $\quad \theta^{(5)} \in \mathbb{R}^2 \quad \theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

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

$(\theta^{(5)})^T x^{(i)} = 0$

Mean normalization

Mean Normalization:

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

learn $\Theta^{(j)}, x^{(i)}$

For user j, on movie i predict:

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

User 5 (Eve):

$$\Theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \underbrace{(\Theta^{(5)})^T(x^{(i)})}_{\to 0} + \mu_i$$