GENERAL ASSEMBLY

INTRO TO DATA SCIENCE LECTURE 16: RECOMMENDATION SYSTEMS

Customers Who Bought This Item Also Bought

Pitch Dark (NYRB Classics)
› Renata Adler
Paperback
$11.54

How Literature Saved My Life
› David Shields
(60)
Hardcover
$18.08

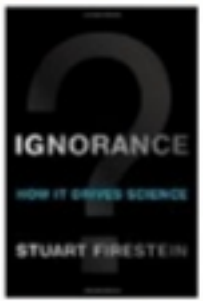
Bleeding Edge
Thomas Pynchon
Hardcover
$18.05

The Flamethrowers: A Novel
› Rachel Kushner
(17)
Hardcover
$15.79

Inspired by Your Wish List

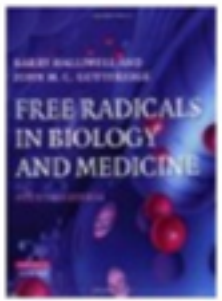
You wished for | Customers who viewed this also viewed

The Secret Anarchy of Science
› Michael Brooks
Paperback
★★★★☆ (6)



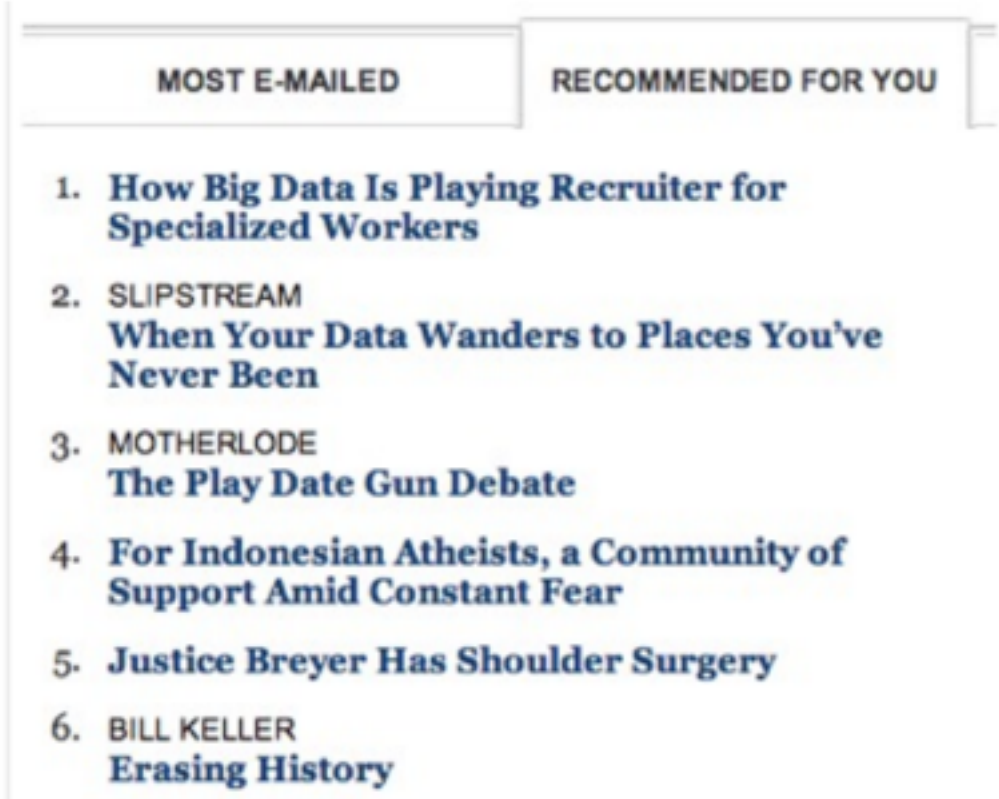
Ignorance: How It Drives Science
› Stuart Firestein
Hardcover
★★★★½ (31)
~~$21.95~~ $13.02

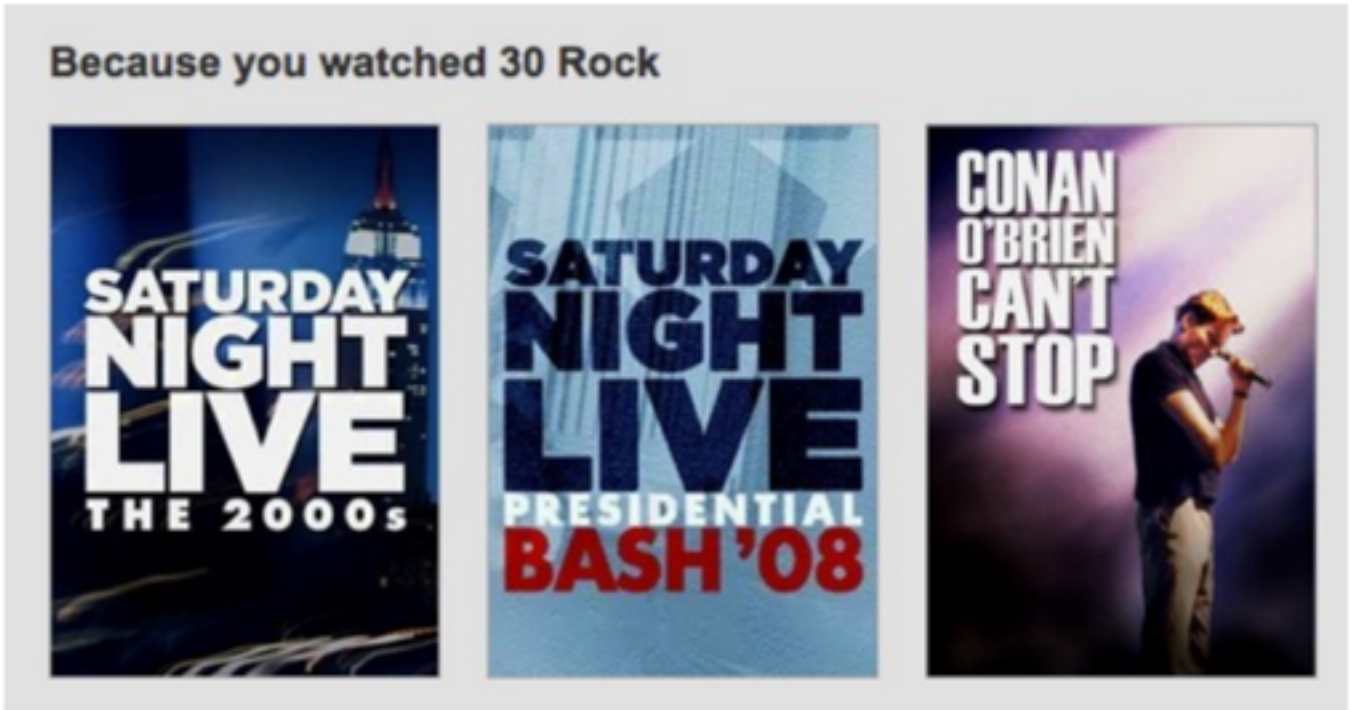
13 Things that Don't Make Sense: The...
› Michael Brooks
Paperback
★★★★☆ (65)
~~$15.95~~ $12.49



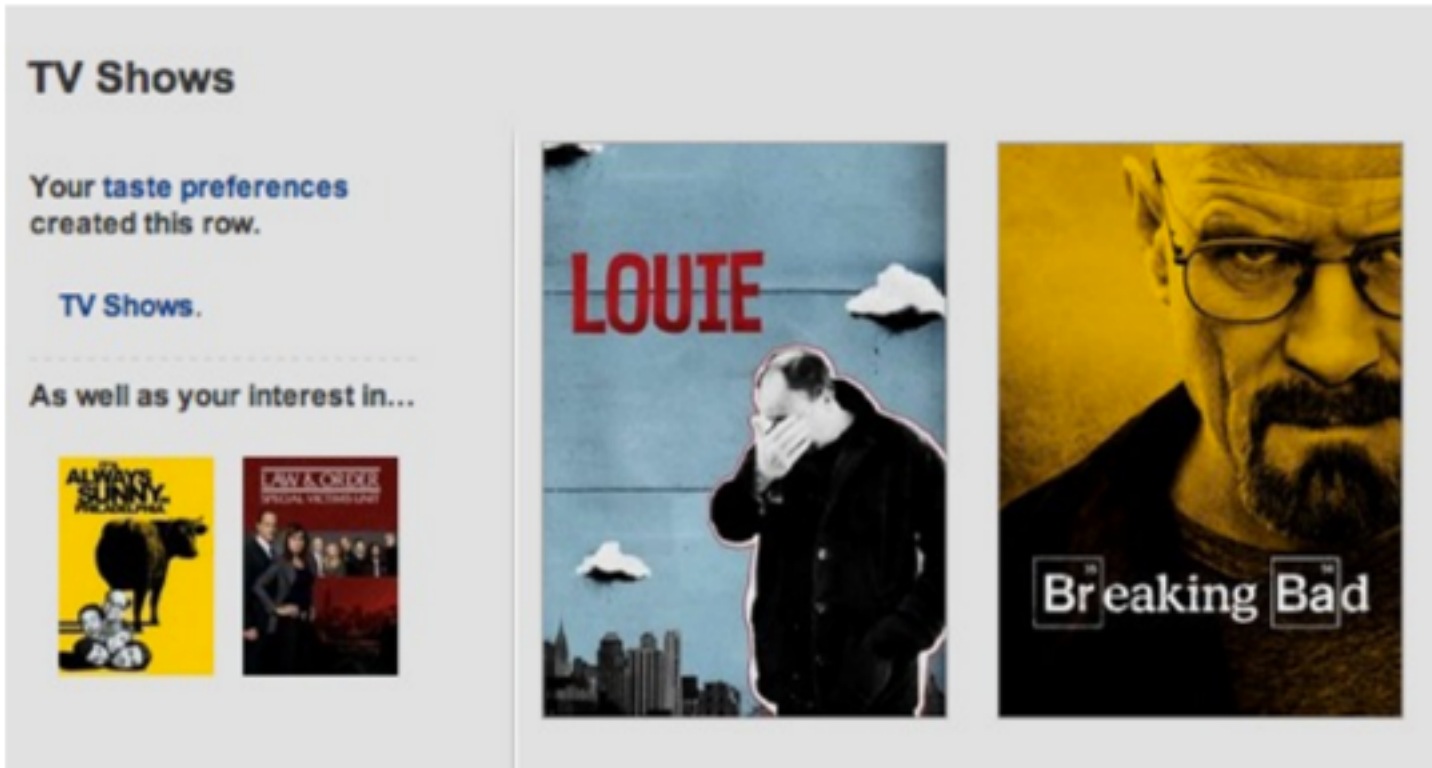
Free Radicals in Biology and Medicine
Barry Halliwell, John Gutteridge
Paperback
★★★★★ (6)
~~$90.00~~ $75.78

Nonsense on Stilts: How to Tell...
› Massimo Pigliucci
Paperback
★★★½☆ (35)
~~$20.00~~ $11.94


MOST E-MAILED
RECOMMENDED FOR YOU
1. How Big Data Is Playing Recruiter for Specialized Workers
2. SLIPSTREAM
When Your Data Wanders to Places You've Never Been
3. MOTHERLODE
The Play Date Gun Debate
4. For Indonesian Atheists, a Community of Support Amid Constant Fear
5. Justice Breyer Has Shoulder Surgery
6. BILL KELLER
Erasing History


Because you watched 30 Rock
SATURDAY NIGHT LIVE THE 2000s
SATURDAY NIGHT LIVE PRESIDENTIAL BASH '08
CONAN O'BRIEN CAN'T STOP

RECOMMENDATION SYSTEMS



Key Concepts - What is a Recommendation system

- The purpose of a recommendation system is to recommend new products to user
- Products may anything, but good examples are books (Amazon), movies (Netflix), & songs (iTunes)
- Recommendations form a substantial part of the revenue stream for companies

Key Concepts - What is a Recommendation system

- Recommendation is, however, widely used in industry
- It is an important application
- Not considered part of ‘academic’ machine learning
- No sklearn algorithms as found for other machine learning topics

KEY CONCEPTS - PROBLEM FORMULATION - EXAMPLE PREDICTING MOVIE RATINGS

- 4 users
- Rate movies on a scale of 0 to 5
- A -1 indicates that the user has not rated that movie

Problem: Given the data predict what the missing movie ratings should be, and then recommend the unwatched movies back to the users based upon the predicted ratings.

User	Susan	Mary	Phil	Greg
Movie				
Big	5	5	0	0
The Notebook	5	-1	-1	0
Barefoot in the Park	-1	4	0	-1
The Bourne Legacy	0	0	5	4
The International	0	0	5	-1

Can we predict the unknown ratings, and where high, suggest these movies to users?

Key Concepts - Recommendation Systems

There are two main approaches to building recommendation systems:

1. Content-based

2. Collaborative Filtering

CONTENT-BASED RECOMMENDATION

KEY CONCEPTS - CONTENT-BASED RECOMMENDATION

- Content-based systems assume the existence of features, here denoted “Romantic” and “Action”

User	Susan	Mary	Phil	Greg	Romantic	Action
Movie						
Big	5	5	0	0	0.90	0.00
The Notebook	5	-1	-1	0	1.00	0.01
Barefoot in the Park	-1	4	0	-1	0.99	0.00
The Bourne Legacy	0	0	5	4	0.10	1.00
The International	0	0	5	-1	0.00	0.90

KEY CONCEPTS - CONTENT-BASED RECOMMENDATION

User	Susan	Mary	Phil	Greg	Romantic	Action
Movie						
Big	5	5	0	0	0.90	0.00
The Notebook	5	-1	-1	0	1.00	0.01
Barefoot in the Park	-1	4	0	-1	0.99	0.00
The Bourne Legacy	0	0	5	4	0.10	1.00
The International	0	0	5	-1	0.00	0.90

- For each movie we can construct a feature vector, consisting of 1 (intercept), the romantic content, and the action content

Big - [1 , 0.9, 0.0]
The Notebook - [1, 1.0, 0.01]

KEY CONCEPTS - CONTENT-BASED RECOMMENDATION

- Could consider the problem as a simple linear regression problem, and provide each user with their own regressor (set of parameters)

- Each user has a set of parameters, θ

- Example: Suppose we want to predict what Susan thinks of 'Barefoot in the Park'?
  - The feature vector for that movie is [1, 0.99, 0.0]
  - If Susan has a set of parameters, θ = [0, 5, 0]

- Then by deriving the inner product of the 2 vectors we would get:

$$0 * 1 + 5 * 0.99 + 0 * 0.0 = 4.95$$

KEY CONCEPTS - CONTENT-BASED RECOMMENDATION

- The parameters that each user has represent their like/dislike of the movie features
- So Susan likes Romantic movies, and dislikes Action movies as reflected in her θ parameters
- Unrated movies can therefore be rated by using the features of the unrated movie
- You could solve this by having a linear regression problem for each user

KEY CONCEPTS - CONTENT-BASED RECOMMENDATION

To learn θ:

$$\min \theta^j \frac{1}{2} \sum_{i:r(i,j)==1} (\theta^{jT} x^i - y^{i,j})^2 + \frac{\lambda}{2} \sum_{k=1}^{N} (\theta_k^j)^2$$

where:

- j is the user
- i is the movie
- $r(i,j) == 1$ if user j has rated movie i
- θ^j is the parameter vector for user j
- x^i is the feature vector for movie i
- $y^{i,j}$ is the rating by user j for movie i, if user j has rated that movie
- $\frac{\lambda}{2} \sum_{k=1}^{N} (\theta_k^j)^2$ is the regularization term

Key Concepts - Content-Based Recommendation

To predict a new rating for a new movie:

Determine the feature values for the movie, e.g. romantic content vs action content vs...

Hence, we have a feature vector, x for the new movie

We have the vector of θ's for the user

The predicted rating is just $y^{i,j} = \theta^{jT} x^i$

KEY CONCEPTS - CONTENT-BASED RECOMMENDATION

To learn $\theta^j (i.e. \theta^1, \theta^2, \theta^3, \ldots, \theta^{nu})$ for all users

$$\min \theta^1, \ldots, \theta^{nu} \frac{1}{2} \sum_{j=1}^{nu} \sum_{i:r(i,j)==1} (\theta^{jT} x^i - y^{i,j})^2 + \frac{\lambda}{2} \sum_{j=1}^{nu} \sum_{k=1}^{N} (\theta_k^j)^2$$

where:

- j is the user
- nu is the number of users
- i is the movie
- $r(i,j) == 1$ if user j has rated movie i
- θ^j is the parameter vector for user j
- x^i is the feature vector for movie i
- $y^{i,j}$ is the rating by user j for movie i, if user j has rated that movie
- $\frac{\lambda}{2} \sum_{j=1}^{nu} \sum_{k=1}^{N} (\theta_k^j)^2$ is the regularization term

KEY CONCEPTS - CONTENT-BASED RECOMMENDATION

Find the minimum using gradient descent using the following update equations (which are the partial derivatives):

Gradient descent update equations:

For the intercept term:

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

For the other terms:

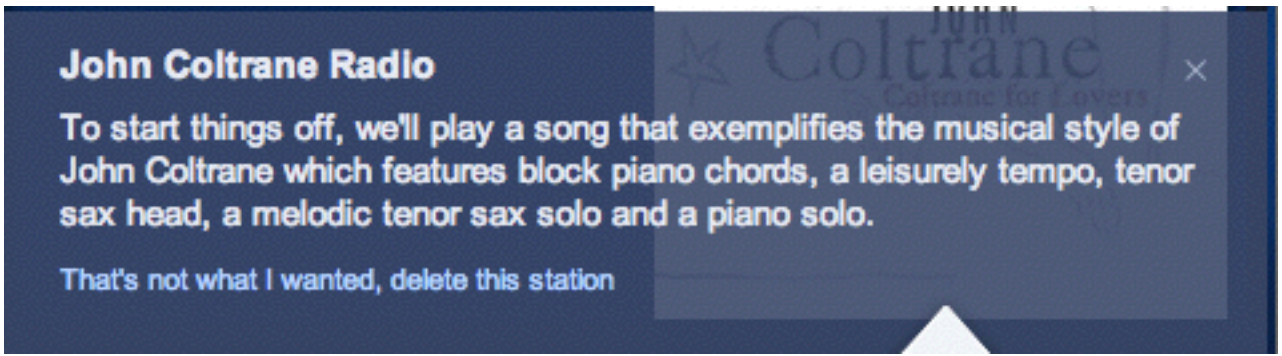
$$\theta_k^j = \theta_k^j - \alpha \left[\sum_{i:r(i,j)==1} (\theta^{jT} x^i - y^{i,j}) x_k^i + \lambda \theta_k^j \right] \text{ (for } k \neq 0)$$

α is the learning rate

Key Concepts - Example of Content-based Recommendation - Pandora

- Pandora uses content-based filtering
- A trained music analyst scores each song based upon hundreds of distinct musical characteristics
- These attributes, or genes, capture not only a song's musical identity, but also many significant qualities that are relevant to understanding a listener's musical preferences

Pandora


John Coltrane Radio
To start things off, we'll play a song that exemplifies the musical style of John Coltrane which features block piano chords, a leisurely tempo, tenor sax head, a melodic tenor sax solo and a piano solo.
That's not what I wanted, delete this station

Key Concepts - Advantages of Content-based Recommendation Systems

- Previous ratings are not required

Key Concepts - Disadvantages of Content-based Recommendation Systems

- You have to create the feature space in the first place
- It is hard to create cross-content recommendations (e.g. books/music/films)
- User profiles can also be created an utilized in much the same way as item features

COLLABORATIVE FILTERING

KEY CONCEPTS - COLLABORATIVE FILTERING

- Collaborative filtering does not rely on a feature set

- Obtaining features, as we saw the content-based system, is both time consuming and expensive

KEY CONCEPTS - COLLABORATIVE FILTERING

2 Principal Methods:

1. Nearest-neighbor

2. Matrix factorization

KEY CONCEPTS - COLLABORATIVE FILTERING

The classical dataset is a matrix:

278858 users →

271379 books ↓

	0	1	2	3	4	5	6	7	8	9
Blind Descent (Anna Pigeon Mysteries (Hardcover))	-1	-1	0	-1	-1	-1	-1	-1	-1	-1
Chinese Cinderella : The true story of an unwanted daughter	-1	-1	-1	7	-1	-1	-1	-1	-1	-1
Children: The Challenge	-1	-1	-1	-1	-1	-1	3	-1	-1	-1
Bad Boys and Tough Tattoos: A Social History of the Tattoo With Gangs, Sailors and Street-Corner Punks, 1950-1965 (Haworth Series in Gay & Lesbian Studies)	-1	-1	-1	-1	-1	-1	-1	-1	-1	7
Simple Loving: A Path to Deeper, More Sustainable Relationships	-1	-1	-1	-1	-1	-1	-1	7	-1	-1

-1	-1	0	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	7	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	3	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	7
-1	-1	-1	-1	-1	-1	-1	7	-1	-1

Where -1 means an unknown rating

KEY CONCEPTS - NEAREST NEIGHBOR

- You have users rating items
- Recommendation goes as follows:
    - To recommend to a first user find another user that has similar ratings for items
    - Having found a similar user, find an item:
        - that the second user rated highly
        - and has not been rated by the first user
    - Recommend this item to the first user

KEY CONCEPTS - NEAREST NEIGHBOR

- These methods are popular and easy to understand and implement

- Similarity between users is measured by:
  - Distance measures
  - Correlation
  - Jaccard
  - Cosine similarity
  - K-Nearest Neighbors
  - Other nearest neighbor algorithms

KEY CONCEPTS - COLLABORATIVE FILTERING - USER SIMILARITY

- Be aware of the strengths and weaknesses of the various similarity measures

- Try to understand the nature of the data:

  - grade-inflation: different users grade to their own scales (suits correlation)
  - dense vs sparse: dense may suit euclidean distance measures (few zeros)
  - sparsity tends to favor cosine similarity
  - outliers: you ‘have’ to like uncle Bob’s Christmas record (KNN may solve this)

KEY CONCEPTS - MATRIX FACTORIZATION

- We have seen that if we have the features for the movies we can predict the set of θ's for each user using linear regression

- If we do not have features, but could obtain a set of θ's from the users then we could estimate the features!

- It's a 'which came first the chicken or the egg problem'
  - if we have features we can estimate θ's
  - if we have θ's we can estimate features

KEY CONCEPTS - COLLABORATIVE FILTERING

- We can actually find sensible values for θ and sensible values for ‘features’ by starting an iterative process

- The θ’s is initially guessed!

- Features are then estimated, the θ’s are then re-estimated, then features are re-estimated, …etc…

- One algorithm that has this approach is called

Alternating Least Squares

KEY CONCEPTS - COLLABORATIVE FILTERING

- Obviously, to work, there needs to be a certain number of users who have a rated a certain number of items, AND

- A certain number of items that have been rated by a certain number of users

- Because the algorithm requires existing user generated data it falls within the notion of collaborative filtering
    - the users are ‘collaborating’ to get better ratings for everyone

KEY CONCEPTS - COLLABORATIVE FILTERING

- Actually, however, you can solve for both sets of values using a single cost function, and minimize this using gradient descent!
- We can simultaneously estimate both sets of values
- The algorithm will ‘discover features’ to use
- The algorithm is called Low Rank Matrix Factorization

KEY CONCEPTS - LOW RANK MATRIX FACTORIZATION

Given: $x^1, x^2, \ldots, x^{nm}$, **estimate** $\theta^1, \theta^2, \ldots, \theta^{nu}$ **for all users**

$$\min_{\theta^1,\theta^2,\ldots,\theta^{nu}} \frac{1}{2} \sum_{j=1}^{nu} \sum_{i:r(i,j)==1} (\theta^{jT} x^i - y^{i,j})^2 + \frac{\lambda}{2} \sum_{j=1}^{nu} \sum_{k=1}^{N} (\theta_k^j)^2$$

Given: $\theta^1, \theta^2, \ldots, \theta^{nu}$, **estimate** $x^1, x^2, \ldots, x^{nm}$ **for all users**

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

KEY CONCEPTS - LOW RANK MATRIX FACTORIZATION

Solve for x and θ simultaneously:

Define the cost function, J, as:

$$J(x^1, x^2, \ldots, x^{nm}, \theta^1, \theta^2, \ldots, \theta^{nu}) = \frac{1}{2} \sum_{(i,j):r(i,j)==1} (\theta^{jT} x^i - y^{i,j})^2 + \frac{\lambda}{2} \sum_{i=1}^{nm} \sum_{k=1}^{N} (x_k^i)^2 + \frac{\lambda}{2} \sum_{j=1}^{nu} \sum_{k=1}^{N} (\theta_k^j)^2$$

and minimize:

$$\min_{x^1, x^2, \ldots, x^{nm}, \theta^1, \theta^2, \ldots, \theta^{nu}} J(x^1, x^2, \ldots, x^{nm}, \theta^1, \theta^2, \ldots, \theta^{nu})$$

where:

- j is the user
- nu is the number of users
- nm is the number of movies
- i is the movie
- $r(i,j) == 1$ if user j has rated movie i
- θ^j is the parameter vector for user j
- x^i is the feature vector for movie i
- $y^{i,j}$ is the rating by user j for movie i, if user j has rated that movie

$\frac{\lambda}{2} \sum_{i=1}^{nm} \sum_{k=1}^{N} (x_k^i)^2$ and $\frac{\lambda}{2} \sum_{j=1}^{nu} \sum_{k=1}^{N} (\theta_k^j)^2$ are the regularization term for the x and θ parameters.

KEY CONCEPTS - LOW RANK MATRIX FACTORIZATION

Minimize using gradient descent:

Gradient Descent Update Equations

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

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

α is the learning rate

Key Concepts - Collaborative Filtering - Low Rank Matrix Factorization

1. Initialize the features, x, and the user parameters, θ, to small random values

2. Minimize the cost function, J, using gradient descent

3. For a user with parameters, θ, and a movie with (learned) features, x, predict a rating by multiplying θ with x

4. Improve by using mean normalization

KEY CONCEPTS - COLLABORATIVE FILTERING - ADVANTAGES

- Domain independent
- Requires no explicit user or item profiles to be created
- Combines predictive accuracy and (relative) scalability

Key Concepts - Collaborative Filtering - Advantages

- Won the Netflix prize!
- Collaborative filtering methods are generally regarded as the state-of-the-art in recommendation technology

Key Concepts - Collaborative Filtering - Disadvantages

- Lots of (high-dimensional) ratings data is needed
- The data is typically very sparse (in the Netflix prize dataset, 99% of possible ratings were missing)
- Need lots of data on new user or item before recommendations can be made

The Cold Start Problem

KEY CONCEPTS - COLLABORATIVE FILTERING - THE COLD START PROBLEM

- The cold start problem arises because we've been relying only on ratings data, or on explicit feedback from users

- Until a user rates several items, we don't know anything about his/her preferences!

- We can get around this by enhancing our recommendations using implicit feedback, which may include things like item browsing behavior, search patterns, purchase history, etc.

- Implicit feedback leads to less accurate ratings, but the data is much denser (and less invasive to collect)

- Implicit feedback can help to infer user preferences when explicit feedback is not available, therefore easing the cold start problem

KEY CONCEPTS - COLLABORATIVE FILTERING - HYBRID METHODS

- Hybrid filtering methods provide another way to get around the cold start problem by combining filtering methods (e.g., by using content-based info to “boost” a collaborative model)
- This content-based info can be item-based as above, or even user-based (e.g., demographic info)
- Hybrid methods can also make the data sparsity issue easier to deal with, by broadening the set of features under consideration