Recommender Systems

Given
● Users: U1 , … , Un
● Movies: M1 , … , Mm
● Ratings: Rij

Goal: Recommend movies to users

Challenges:
● Scale (millions of users, millions of movies)
● Cold Start (change in user base, change in content)
● Sparse Data (Not many users rank movies)

Neighborhood Methods
● (user, user) similarity measure
    ○ i.e. recommend same movies to similar users (requires info about users)
● (item, item) similarity measure
    ○ i.e. recommend movies that are similar (requires info about movies)

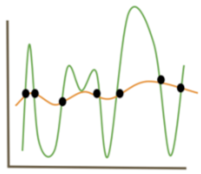
Pros:
● Intuitive / easy to explain
● No training
● Handles new users/items
Challenges:
● Users rate differently (bias)
● Ratings change over time (bias)

Feature Extraction - Content-Based:
Realistically:
● It's difficult to characterize movies and users with the right features
● Characterization of users and movies may not be accurate
    ○ If you are using genres for example, movies with varying degree of "comedy" will get the tag "comedy".
Goal:
● Discover the best features in an automated way

Content-Based: assume you have features for movies - want to learn features for users
Collaborative filtering: want to learn features for both users and movies

$$P^{(j)} = \arg\min_{P} \frac{1}{\|M^{(j)}\|} \sum_{i \in M^{(j)}} (P^T Q^{(i)} - r_{ij})^2 + \lambda \|p\|^2$$

Regularization Term: a penalty on the size of the parameter p



Feature Extraction - Collaborative Filtering
Can’t use SVD because R is sparse… BUT, we can formulate an optimization problem to solve:

$$\min_{p,q} \sum_{i,j \in R} (r_{ij} - p_i^T q_j)^2 + \lambda(\|p\|_F^2 \, \|p\|_F^2)$$

R = PQ

To solve, take derivatives wrt P & Q. Then, just like Expectation-Maximization Algorithm from GMM:
1. Start with random Q
2. Get P
3. Improve Q
4. Repeat 2 & 3

Linear Regression

Motivation: Suppose we are given a curve y = h(x), how can we evaluate whether it is a good fit to our data?
Compare h(xi) to yi for all i.
Goal: For a given distance function d, find h where L is smallest.

$$L(h) = \sum_i d(h(x_i), y_i)$$

We want to minimize:

$$L(h) = \sum_i d(h(x_i), y_i)$$

We want to maximize:

L(h) = P(Y | h)

Assumptions Let’s start by assuming our data was generated by a linear function plus some noise:

$$\vec{y} = h_\beta(\mathrm{X}) + \vec{\epsilon}$$

$h(x) = \beta_1 x$	✔
$h(x) = \beta_0 + \beta_1 x$	✔
$h(x) = \beta_0 + \beta_1 x + \beta_2 x^2$	✔
$h(x) = \beta_1 \log(x) + \beta_2 x^2$	✔
$h(x) = \beta_0 + \beta_1 x + \beta_1^2 x$	✘

1. The relation between x (independent variable) and y (dependent variable) is linear in a parameter $\boldsymbol{\beta}$.
2. ϵ i are independent, identically distributed random variables following a N(0, σ2) distribution. (Note: σ is constant)