**Content-based Recommendations**

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)

= parameter vector for user j

x\_i = feature vector for movie i

For user j, movie i, predicted rating:

m\_j = number of movies rated by user j

To learn (parameter for user j):

To learn :

+

**Collaborative Filtering**

Optimization algorithm

Given, to learn :

Given (and movie ratings), can estimate

+

Given , can estimate

+

Minimizing and simultaneously:

J() =++

min J(…)

CF Algorithm

1. Initialize and to small random values.
2. Minimize J() using gradient descent(or advanced optimization algorithm). E.g. for every j = 1, …, , i = 1, …, :

:= – () +

:= – ( + )

Note: x and , there is no x0 = 1.

1. For a user with parameters and movie with (learned) features x, predict a star rating of

**Low Rank Matrix Factorization**

Finding related movies

For each product i, we learn a feature vector

How to find movies j related to movie i?

Smallest ||xi-xj||, xj is related to xi.

Mean normalization

Grouping all star scores into a matrix Y and computing each movie’s average score and stack it in a vector . After that, Y’ = Y –; Use Y’ as the data to compute the.

For each user j, on movie i predict:

+