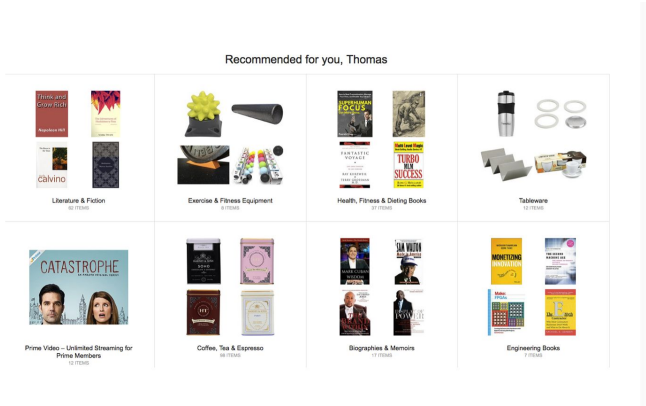
Building recommendation system

Mohit Gupta, Runzhi Yang

Why recommendation system



Goal

Goal: Guessing  ‘s rating on the movie 

Naive Approaches

r = μ + b_i + b_u

Prediction = Avg Rating of all movies + how Magic Mike is rated overall + how Yexiang rates movies



= 3.3 + 0.5 - 0.8

= 3.6

Cooler approaches

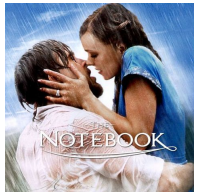
Neighborhood Models

Latent factor Models

Neighborhood method

Users   

Movies 



Observation - Yexiang and Mohit have similar tastes.

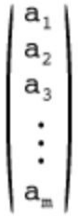
Problem Statement - Figure out Yexiang's rating for an unseen movie (say **Inception**!)

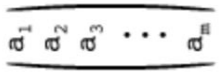
Method - Take weighted average of rating given by top K similar users.

Similarity Measure - Cosine Distance etc.

Latent Factor Method

We want to capture more information about both the item and user.





Funk SVD

$$r_{ui} = \mu + b_i + b_u + p_u^T q_i$$






SVD++

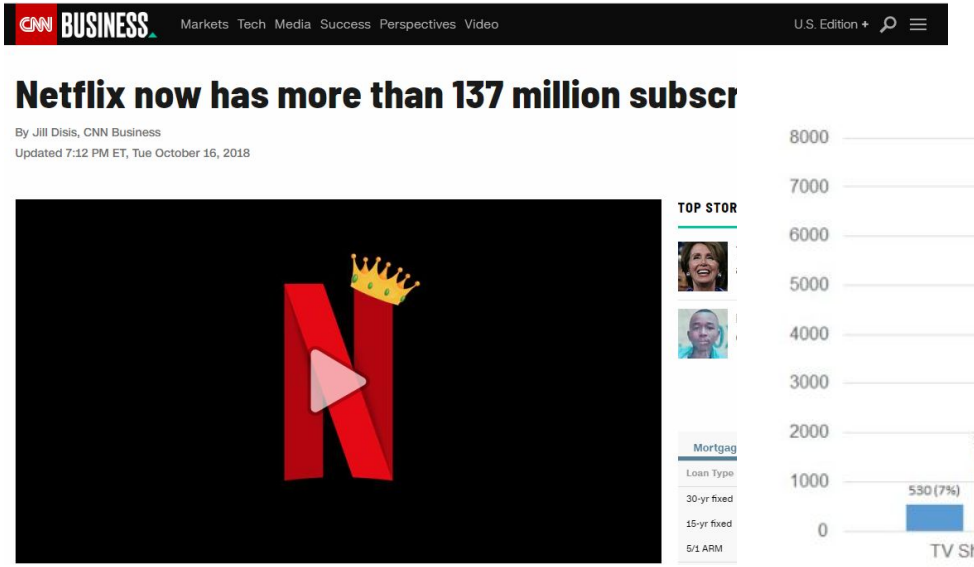
Implicit information

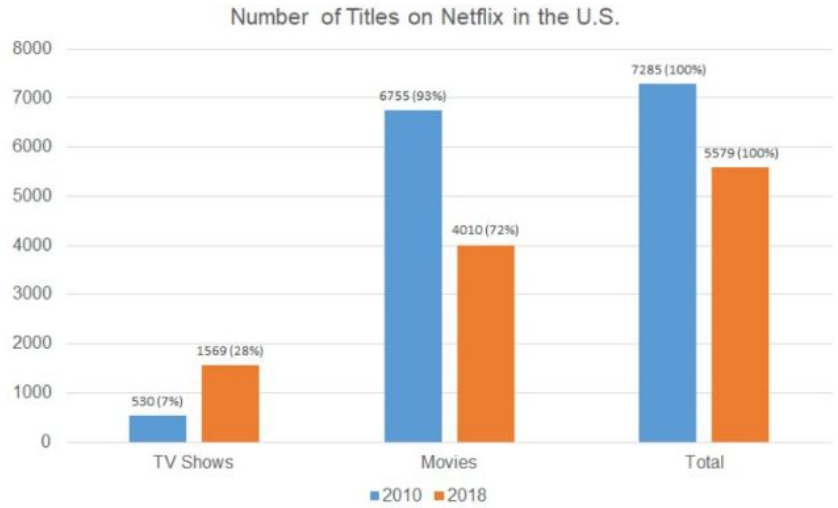
$$r_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j\right)$$

R(u)	The Notebook	Magic Mike	Lion King
Yexiang	4	?	2

N(u)	The Notebook	Magic Mike	Lion King
Yexiang	1	0	1

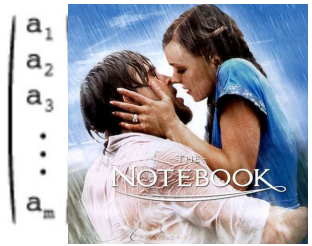
Not very practical



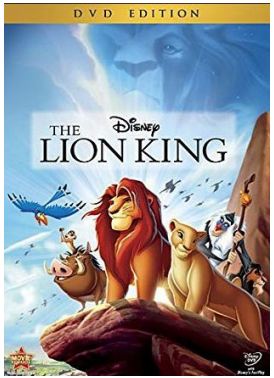


Asymmetric SVD

R(u)	The Notebook	Magic Mike	Lion King
Yexiang	4	?	2



+



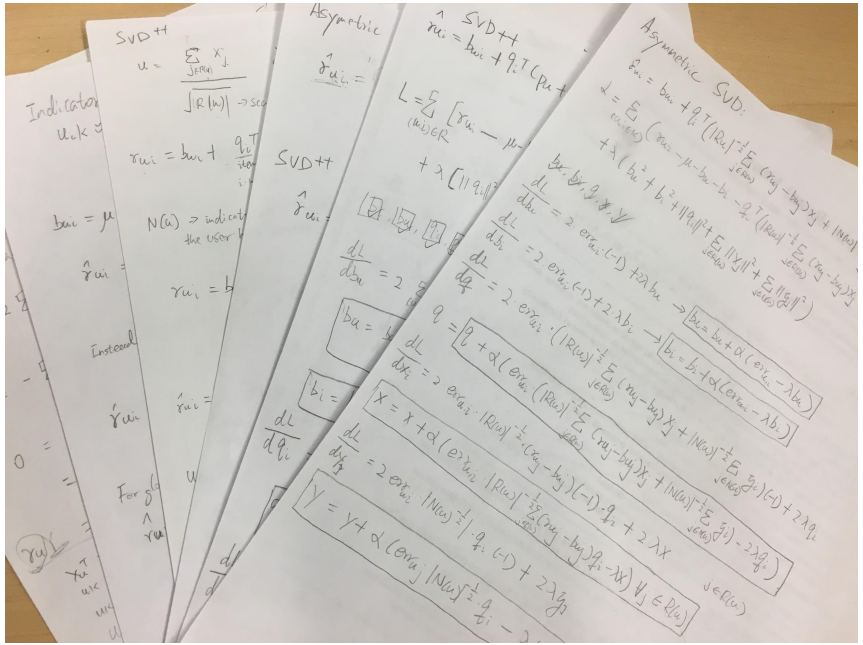
+ ……..

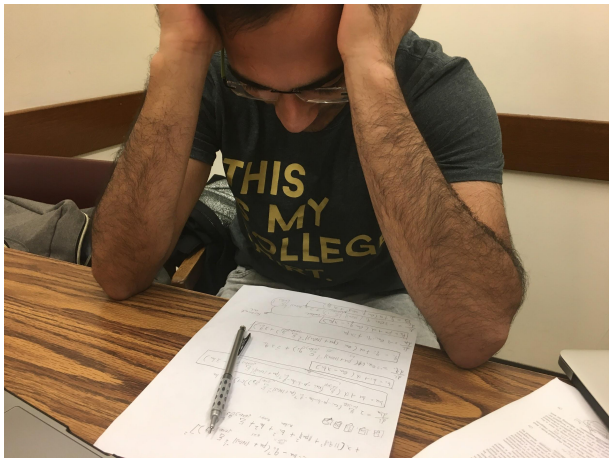
Asymmetric SVD Cont.

$$r_{ui} = b_{ui} + q_i^T \left(|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right)$$

$$b_{ui} = \mu + b_i + b_u$$

But how do you get the weights?





What are the results?

MSE	10 Iterations	50 Iterations	100 Iterations	Best Result
Funk SVD	1.01832	0.95265	0.93090	0.90125
SVD++	0.95797	0.90997	0.88431	0.88317
Asymmetric SVD	0.95293	0.91231	0.89169	0.88789

Thank you!