COMP6237 Data Mining

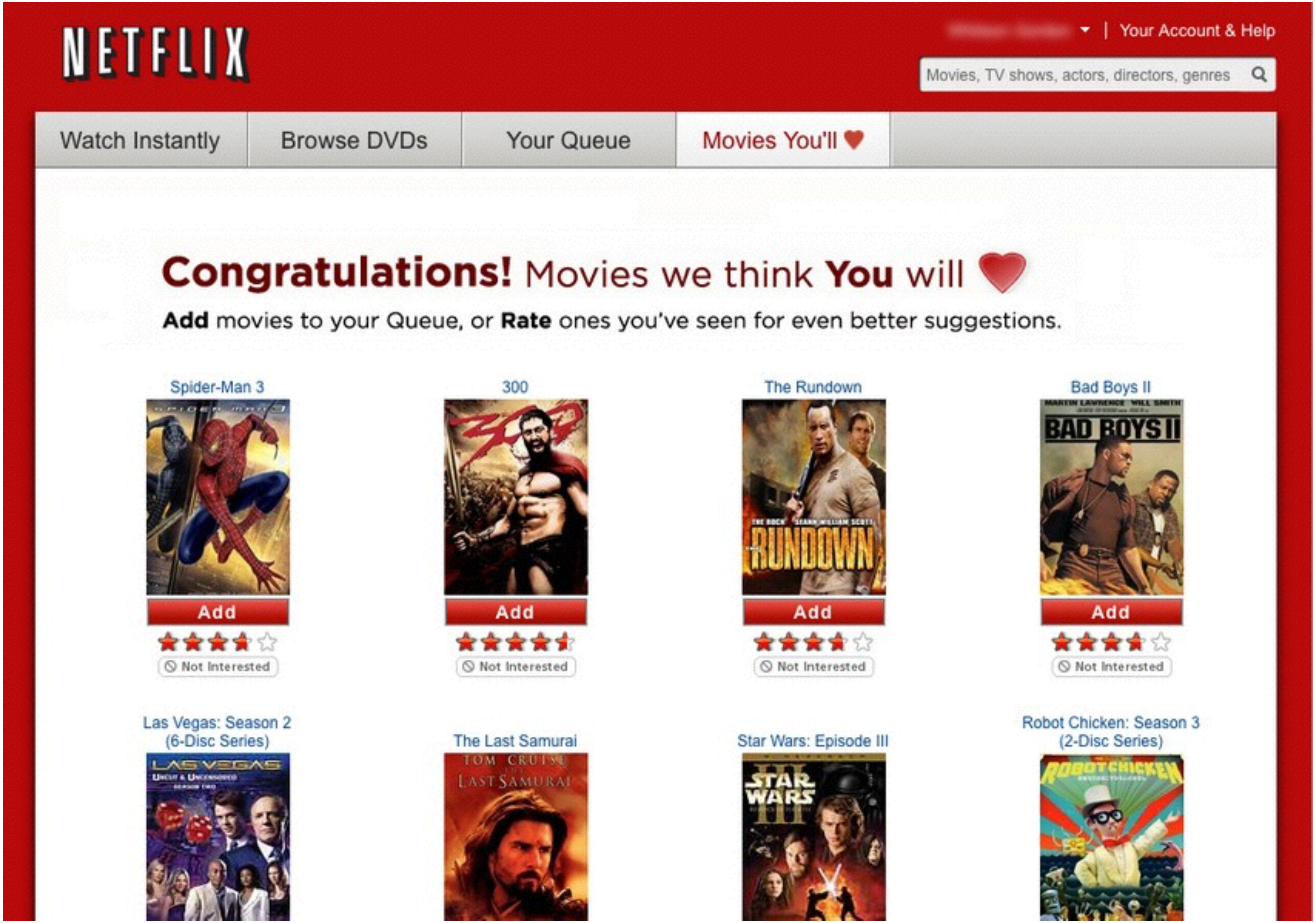
Recommending Netflix Movies

Jonathon Hare
jsh2@ecs.soton.ac.uk

Introduction

- Recap - recommender systems
- Introducing the Netflix Challenge
- The game changer: Approximate SVD
- What happened after the competition?

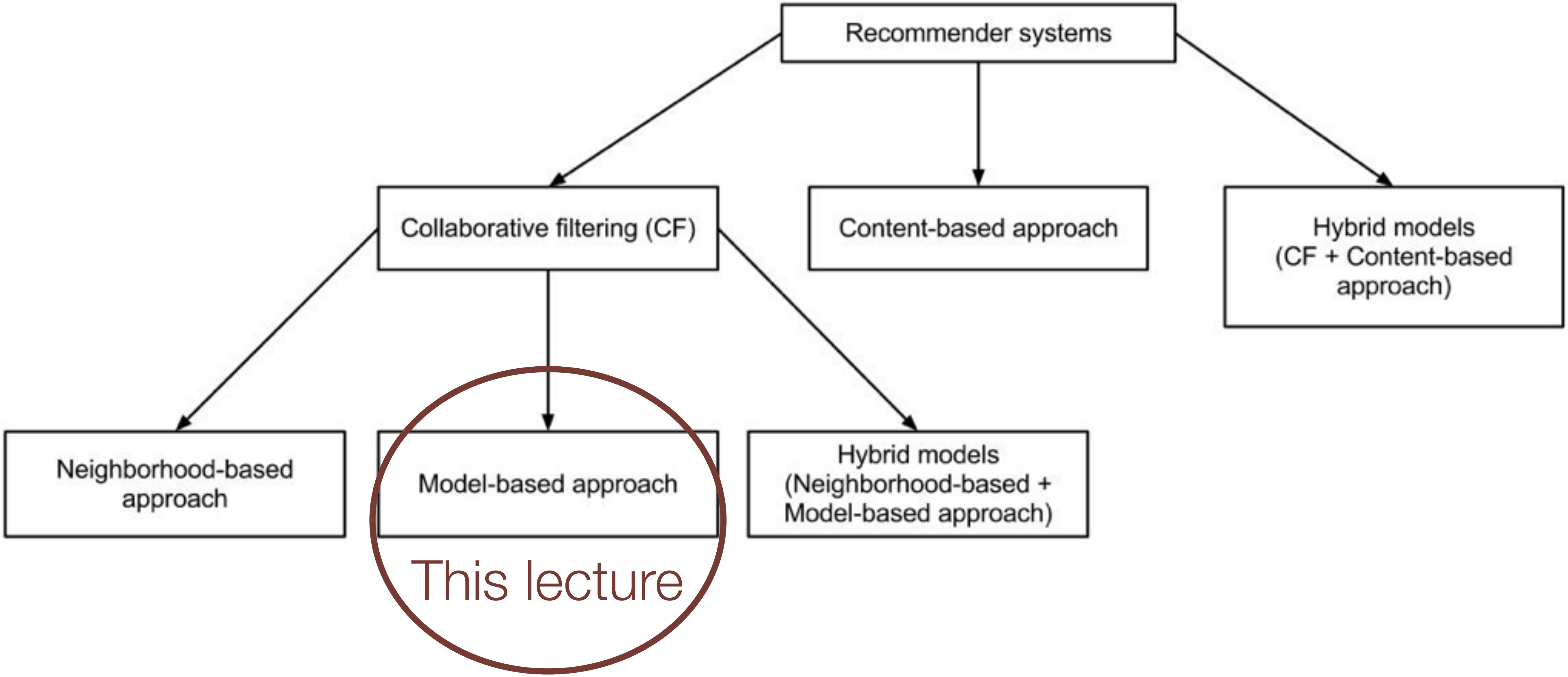
Netflix predicts movies you ♡ based on past **numeric ratings**



Congratulations! Movies we think **You** will ♥

Add movies to your Queue, or **Rate** ones you've seen for even better suggestions.

Recommender systems taxonomy


Recommender systems
Collaborative filtering (CF)
Content-based approach
Hybrid models (CF + Content-based approach)
Neighborhood-based approach
Model-based approach
Hybrid models (Neighborhood-based + Model-based approach)
This lecture

Assume we have sparse user preferences:

	Lady in the Water	Snakes on a Plane	Just My Luck	Superman Returns	The Night Listener	You, Me and Dupree
Lisa	2.5	3.5	3.0	3.5	3.0	2.5
Gene	3.0	3.5	1.5	5.0	3.0	3.5
Michael	2.5	3.0	○	3.5	4.0	○
Claudia	○	3.5	3.0	4.0	4.5	2.5
Mick	3.0	4.0	2.0	3.0	3.0	2.0
Jack	3.0	4.0	○	5.0	3.0	3.5
Toby	○	4.5	○	4.0	○	1.0

How can we predict the values of the unknowns?

The Netflix Challenge

- An open competition to develop the best collaborative filtering algorithm for film ratings
  - Launched in October 2006
  - Ran until late 2009
- Prizes:
  - Based on improvement over *Cinematch*
  - Annual progress prizes of $50000 based on beating Cinematch or previous winner by 1% RMSE
  - $1M Grand prize for the team to beat Cinematch by 10% RMSE

Dataset

- ***Training***:
  - 100,480,507 ratings from 480,189 users on 17,770 movies.
    - <user, movie, date of grade, grade>
    - user and movie are integer IDs
    - while grades are from 1 to 5 (integral) stars
- ***Probe****:* subset of 1,408,395 ratings from training

Dataset

- ***Qualifying***:
  - 2,817,131 triplets of <user, movie, date of grade>, split into:
    - ***Test****:* subset of 1,408,789 ratings used to determine winners
    - ***Quiz:*** subset of 1,408,342 ratings used to calculate leaderboard scores
  - True ratings and the split was only known to the judges

A Brief History

- Launched 2nd October 2006
- WXYZConsulting beat Cinematch on the 8th October
- By 15th October 3 teams had beaten Cinematic
    - One by over 1% enabling qualification for annual progress prize
- 27th October:
    - Over 10,000 teams registered
    - *Simon Funk*, proposes an approach that comes in 8th position & is based on matrix factorisation
        - Fully detailed in a series of blog posts from October to December
        - **Game-changing impact**

A Brief History

- 2007 Progress prize won by team KorBell (aka BellKor)
    - Blended techniques including matrix factorisation, NN and RBMs
- 2008 Progress prize won by team BellKor+BigChaos
    - Blended techniques from ’07 + improved factor models (matrix factorisation) and temporal modelling
- Grand prize awarded to “BellKor's Pragmatic Chaos” in July 2009

Key findings & Lessons Learned

- RMSE is not a good success measure
  - Doesn’t reflect user satisfaction
- Time matters
  - Just because I liked something in the past doesn’t mean I would like it now
- Matrix Factorisation can be very powerful
  - SVD-like solutions played a very big part…

Key findings & Lessons Learned

- One method is not enough
  - Teams that did well blended predictions across models (i.e. used ensemble methods)
- There are potentially better ways to improve recommendation
  - The Netflix competition data is both noisy and constrained; improving data quality and adding features could be way more powerful than attempting to find better models

Latent Factor Models for Recommendation

Recap: Latent Semantic Analysis

- LSA works by making a **low-rank approximation** under the following assumptions:
  - The original term-document matrix is **noisy**
    - anecdotal instances of terms are to be eliminated.
      - the approximated matrix is **de-noised**
  - The original term-document matrix is **overly sparse** relative to the "*true*" term-document matrix
    - We want to capture **synonymy**

Can we use a low-rank approximation for CF?

- Intuition: can apply LSA methodology to User-Movie Ratings matrix
  - The *concepts* will represent different categories of movies (although won’t necessarily be *understandable* as such*)*
- So, compute rank-k SVD of ratings matrix, then reconstruct rank-k estimate of original matrix & read out the predicted ratings
  - … easy?!

Problem 1: User-Movie Ratings Matrix is big

- … very big:
    - 480,189 * 17700 = 8.5 billion entries
        - (that’s 64GB, assuming double precision FP)
- Standard power/Arnoldi/Lanczos methods are not going to cut it!

Problem 2: Ratings Matrix has unknowns

- Matrix is very sparse
  - But the sparse values are not zero; they are unknown
  - With LSA we usually treated unknowns as zero, but that is not really the right thing to do here

Simon Funk (Brandyn Webb) & “Approximate SVD”

- Simon Funk’s first blog post roughly described how he solved the SVD using an incremental approach to LSA
  - This in turn was based on neural network research from the 80’s on incremental methods for finding eigenvectors
- Later blog posts show how this idea was further developed into a rather **elegant** formulation of an approximate SVD

“But, just because there are five hundred really complicated ways of computing singular value decompositions in the literature doesn't mean there isn't a really simple way too: Just take the derivative of the approximation error and follow it. This has the added bonus that we can choose to simply ignore the unknown error on the 8.4B empty slots.”

–Simon Funk, from “Netflix Update: Try This at Home”

Formulating SVD using Gradient Descent

- Let **R** be a user-movie ratings matrix

- Assume we want to perform the following decomposition: **R** = **UF** for matrix **R** of size *m* × *n*, **U** of size *m* × *c* and **F** of size *c* × *n,* with *c*<<rank(**R**), subject to min(||**R** - **UF**||$_F$)

  - (This is essentially just the definition of SVD, but with the square-root of the singular values rolled into both **U** and **F**)

- Can we solve for **U** and **F** using SGD?

$$w := w - \alpha \nabla Q_i(w)$$

Formulating SVD using Gradient Descent

- …yes!

$$
\begin{aligned}
\mathbf{U}_{IK} &:= \mathbf{U}_{IK} - \alpha \frac{\delta E^2}{\delta \mathbf{U}_{IK}} &&= \mathbf{U}_{IK} + 2\alpha(\mathbf{R}_{ij} - p_{ij})\mathbf{F}_{KJ} \\
\mathbf{F}_{KJ} &:= \mathbf{F}_{KJ} - \alpha \frac{\delta E^2}{\delta \mathbf{F}_{KJ}} &&= \mathbf{F}_{KJ} + 2\alpha(\mathbf{R}_{ij} - p_{ij})\mathbf{U}_{IK}
\end{aligned}
$$

- (see handout for derivation)

Going further: regularised approximate SVD

- Approaching the computation of SVD this way leads to some interesting additional possibilities
  - Firstly we can optimise only over known data
  - But we could modify the update rules to enforce additional constraints
    - add regularisation!
      - Might want to penalise the magnitude of the features
    - could even modify the objective (and re-derive update rules) to be non-linear (e.g. by introducing clipping or a sigmoidal activation function)

Epilogue: The end of the Netflix Challenge

The final solution

- “BellKor's Pragmatic Chaos” winning solution highlighted two key predictive approaches:
  - SVD/factorisation based latent factor models
  - Restricted Boltzmann Machine models
- These were coupled as an ensemble with highly optimised blending

Moving into production

- The factorisation and RBM methods were implemented in production at Netflix
- But the ensemble was not…
  - Offline studies showed it to be too computationally expensive at scale
  - Expected improvement not worth the engineering effort
  - Focus had shifted to other ways to make money that had more impact (i.e. roll out in the UK)

The death of the dataset

- Is there any customer information in the dataset that should be kept private?
  - “No, all customer identifying information has been removed; all that remains are ratings and dates. This follows our privacy policy, which you can review here. Even if, for example, you knew all your own ratings and their dates you probably couldn’t identify them reliably in the data because only a small sample was included (less than one-tenth of our complete dataset) and that data was subject to perturbation. Of course, since you know all your own ratings that really isn’t a privacy problem is it?”

The advent of de-anonymisation

- In 2007 Arvind Narayanan and Vitaly Shmatikov showed that it was possible to de-anonymise users from the Netflix dataset.
    - In particular, they cross-correlated the Netflix data against public IMDB ratings
        - Used a small subset of IMDB users as this was a proof of concept
            - identified a user with an eccentricity of 28 - that is 28 standard deviations from the closest to second best match…
                - for most users eccentricity is 2; **this is a statistically significant match!**

…this user was “interesting”…

- We can guess political orientation based on his strong opinions about “Power and Terror: Noam Chomsky in Our Times” and “Fahrenheit 9/11.”

- Strong guesses about his religious views can be made based on his ratings on “Jesus of Nazareth” and “The Gospel of John”.

- He did not like “Super Size Me” at all; perhaps this implies something about his physical size?

- Movies with predominantly gay themes, “Bent” and “Queer as folk” were rated one star out of five.

- He is a cultish follower of “Mystery Science Theater 3000”.

- …

…and then came the lawsuit

- On December 17, 2009, four Netflix users filed a class action lawsuit against Netflix, alleging that Netflix had violated U.S. fair trade laws and the Video Privacy Protection Act by releasing the datasets.
    - There was a lot of public debate about privacy for research participants
    - and the Federal Trade Commission became *interested*
- On March 12, 2010 the second Netflix challenge (announced the previous year) was cancelled
- On March 19, 2010, Netflix reached a settlement with the plaintiffs, after which they voluntarily dismissed the lawsuit.
- As a result of the lawsuit, the original dataset was removed from public circulation (although you can still find copies on the web)

Summary