Behind the Curtain: Data & Algorithms that power the Netflix User Experience

January, 2014

Xavier Amatriain
Director - Algorithms Engineering - Netflix

@xamat

Netflix Prize

COMPLETED

What we were interested in:

- High quality *recommendations*

Proxy question:

- Accuracy in predicted rating
- Improve by 10% = $1million!

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

NETFLIX

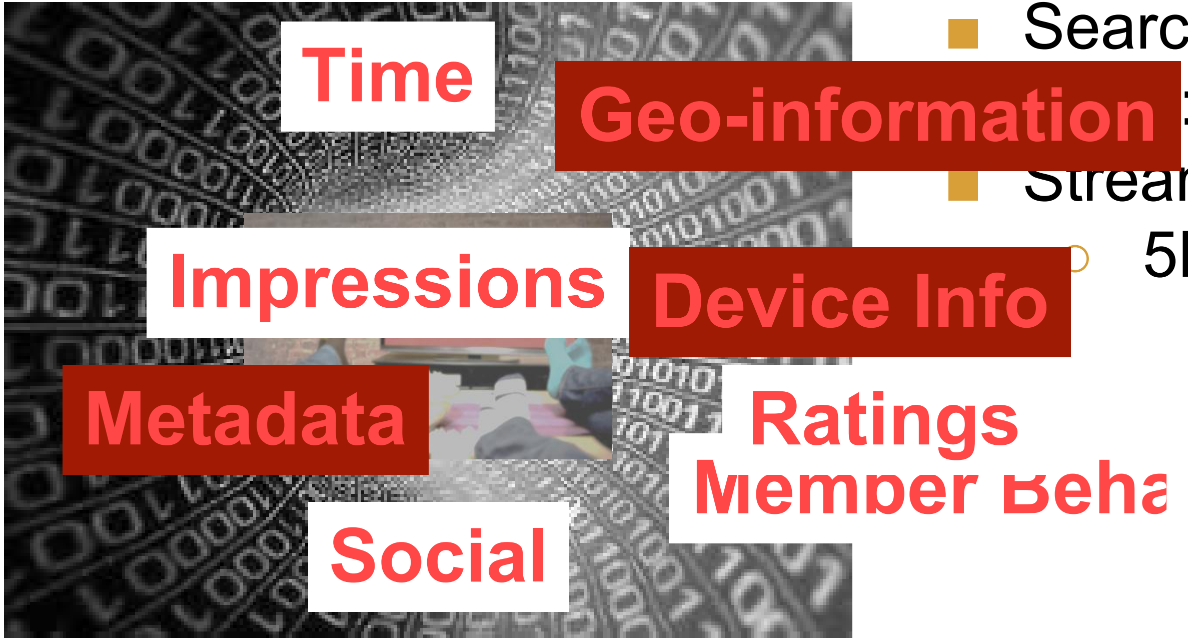
From the Netflix Prize to today

2006

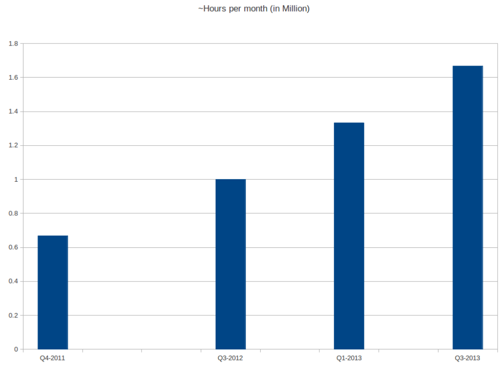
2013

Big Data @Netflix

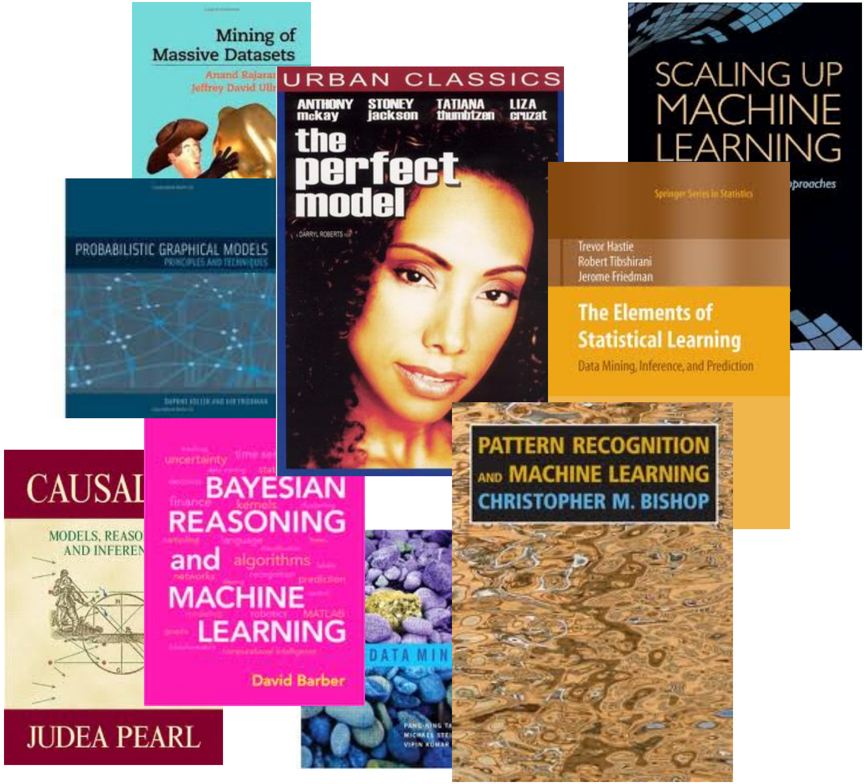
- > 40M subscribers
- Ratings: ~5M/day
- Searches: >3M/day
- : > 50M/day
- Streamed hours:
    - 5B hours in Q3 2013



Demographics

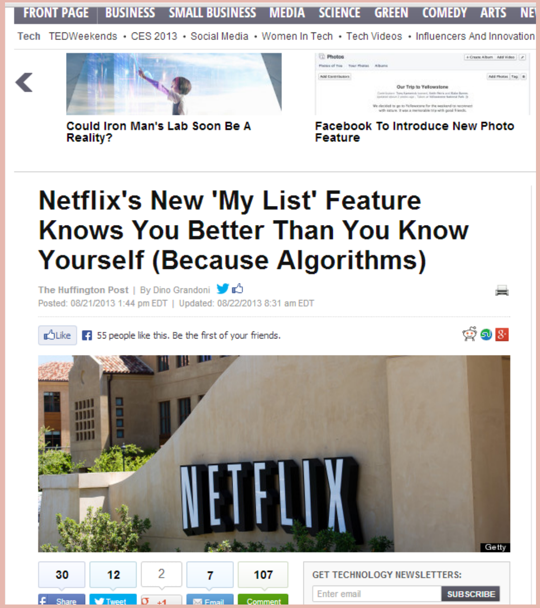


Smart Models



- Regression models (Logistic, Linear, Elastic nets)
- SVD & other MF models
- Factorization Machines
- Restricted Boltzmann Machines
- Markov Chains & other graph models
- Clustering (from k-means to HDP)
- Deep ANN
- LDA
- Association Rules
- GBDT/RF
- …

Behind the curtain: Netflix Algorithms



“In a simple Netlfix-style item recommender, we would simply apply some form of matrix factorization (i.e NMF)”

Machine Learning

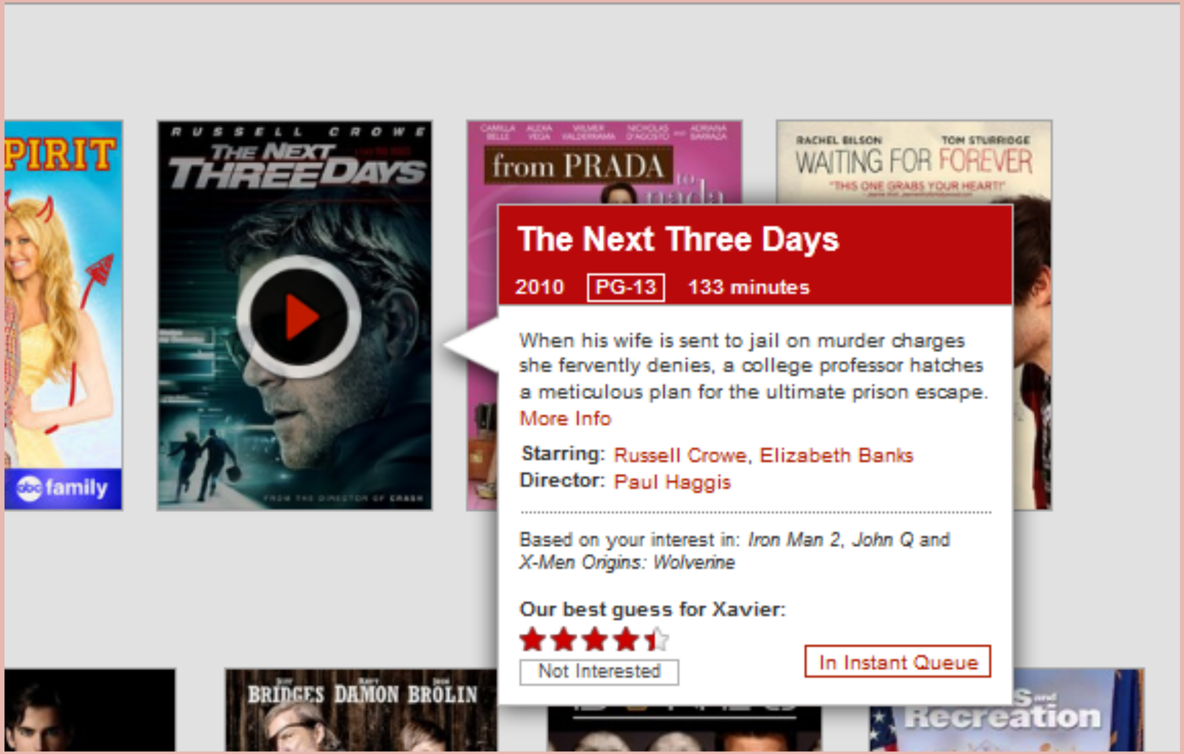
notes, thoughts, and practice of applied machine learning

Music Recommendations and the Logistic Metric Embedding

Posted on October 28, 2013

0

Rating Prediction



2007 Progress Prize

- Top 2 algorithms
  - SVD - Prize RMSE: 0.8914
  - RBM - Prize RMSE: 0.8990
- Linear blend Prize RMSE: 0.88
- Currently in use as part of Netflix' rating prediction component
- Limitations
  - Designed for 100M ratings, we have 5B ratings
  - Not adaptable as users add ratings
  - Performance issues

SVD - Definition

$$\boxed{A_{[n \times m]} = U_{[n \times r]} \, \Lambda_{[r \times r]} \, (V_{[m \times r]})}^T$$

- **A**: *n x m* matrix (e.g., n documents, m terms)
- **U**: *n x r* matrix (n documents, r concepts)
- **Λ**: *r x r* diagonal matrix (strength of each 'concept') (r: rank of the matrix)
- **V**: m x r matrix (m terms, r concepts)

SVD - Properties

- 'spectral decomposition' of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \times \begin{bmatrix} \lambda_1 & \emptyset \\ \emptyset & \lambda_2 \end{bmatrix} \times \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

- 'documents', 'terms' and 'concepts':
  - **U**: document-to-concept similarity matrix
  - **V**: term-to-concept similarity matrix
  - Λ: its diagonal elements: 'strength' of each concept

SVD for Rating Prediction

- User factor vectors $p_u \in \Re^f$ and item-factors vectors $q_v \in \Re^f$
- Baseline (bias) $b_{uv} = \mu + b_u + b_v$ (user & item deviation from average)
- Predict rating as $r'_{uv} = b_{uv} + p_u^T q_v$
- SVD++ (Koren et. Al) asymmetric variation w. implicit feedback

$$r'_{uv} = b_{uv} + q_v^T \left(|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right)$$

- Where
  - $q_v, x_v, y_v \in \Re^f$ are three item factor vectors
  - Users are not parametrized, but rather represented by:
    - R(u): items rated by user u & N(u): items for which the user has given implicit preference (e.g. rated/not rated)

Restricted Boltzmann Machines

- Restrict the connectivity in ANN to make learning easier.
  - Only one layer of hidden units.
    - Although multiple layers are possible
  - No connections between hidden units.
- Hidden units are independent given the visible states..
- RBMs can be stacked to form Deep Belief Networks (DBN) – 4^{th} generation of ANNs

Restricted Boltzmann Machines for Collaborative Filtering

Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Andriy Mnih AMNIH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
University of Toronto, 6 King's College Rd., Toronto, Ontario M5S 3G4, Canada

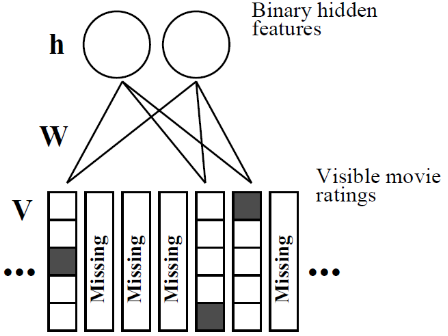


Figure 1. A restricted Boltzmann machine with binary hidden units and softmax visible units. For each user, the RBM only includes softmax units for the movies that user has rated. In addition to the symmetric weights between each hidden unit and each of the $K = 5$ values of a softmax unit, there are 5 biases for each softmax unit and one for each hidden unit. When modeling user ratings with an RBM that has Gaussian hidden units, the top layer is composed of linear units with Gaussian noise.

What about the final prize ensembles?

- Our offline studies showed they were too computationally intensive to scale
- Expected improvement not worth the engineering effort
- Plus, focus had already shifted to other issues that had more impact than rating prediction...

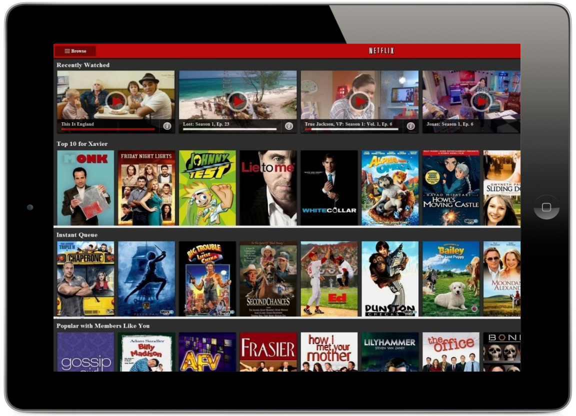
Ranking


Ranking
Rows
Netflix
Recently Watched
Top 10 for Justin
Popular on Facebook
Goofy TV Shows
Visually-striking Exciting Foreign Movies
Sci-Fi & Fantasy

Ranking

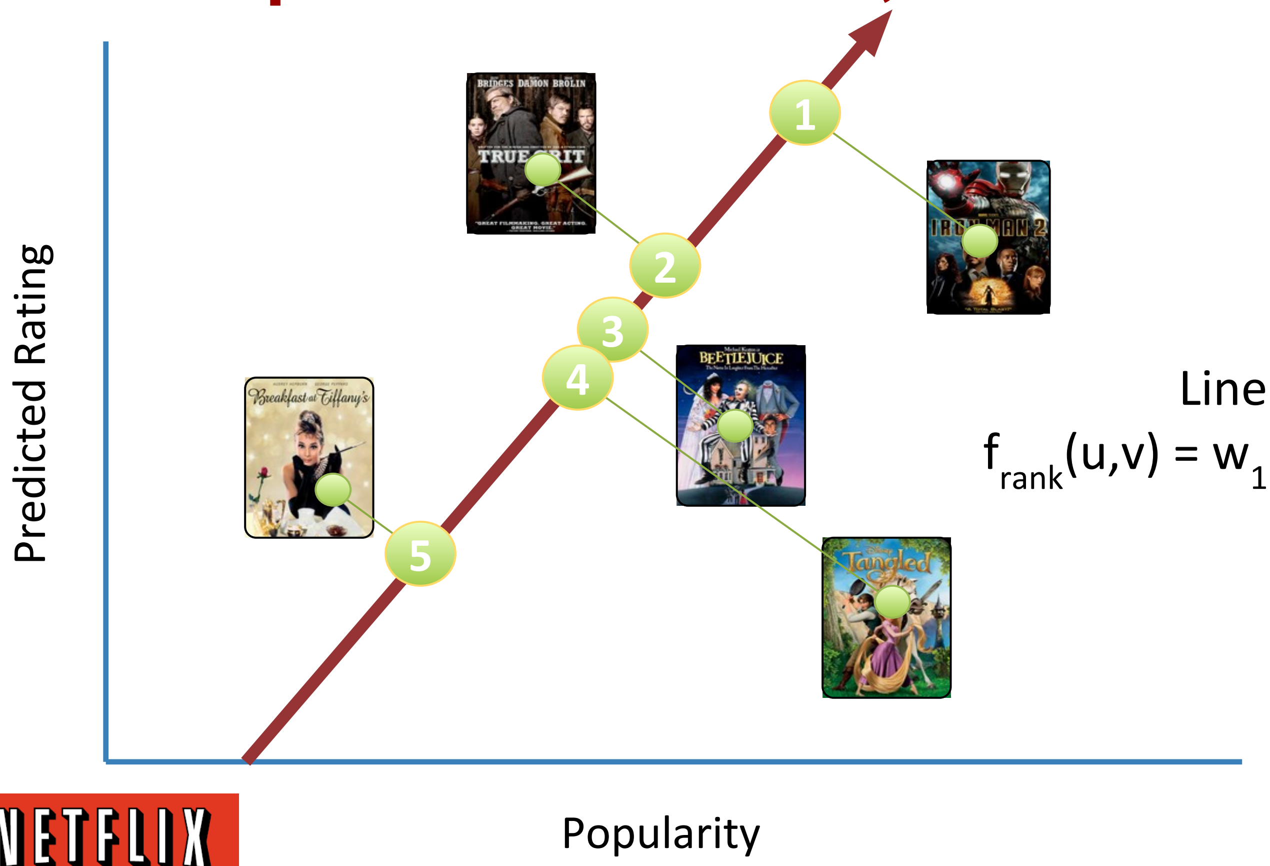
- Ranking = **Scoring** + **Sorting** + **Filtering** bags of movies for presentation to a user
- Key algorithm, sorts titles in most contexts
- Goal: Find the best possible ordering of a set of *videos* for a *user* within a specific *context* in real-time
- Objective: maximize consumption & “enjoyment”

- Factors
  - Accuracy
  - Novelty
  - Diversity
  - Freshness
  - Scalability
  - …



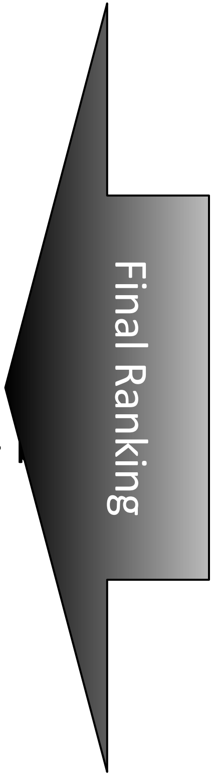
NETFLIX

Example: Two features, linear model

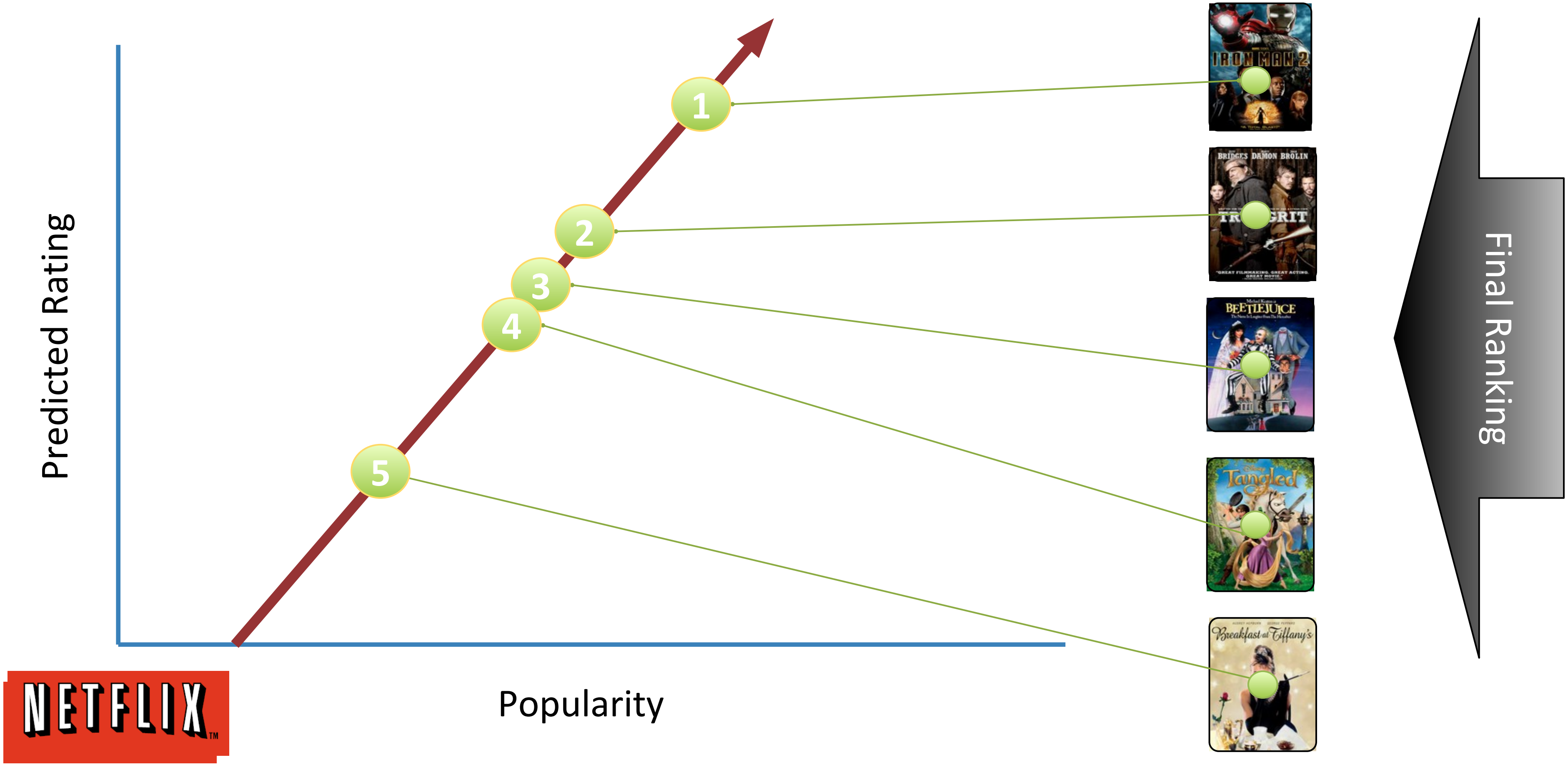


Linear Model:

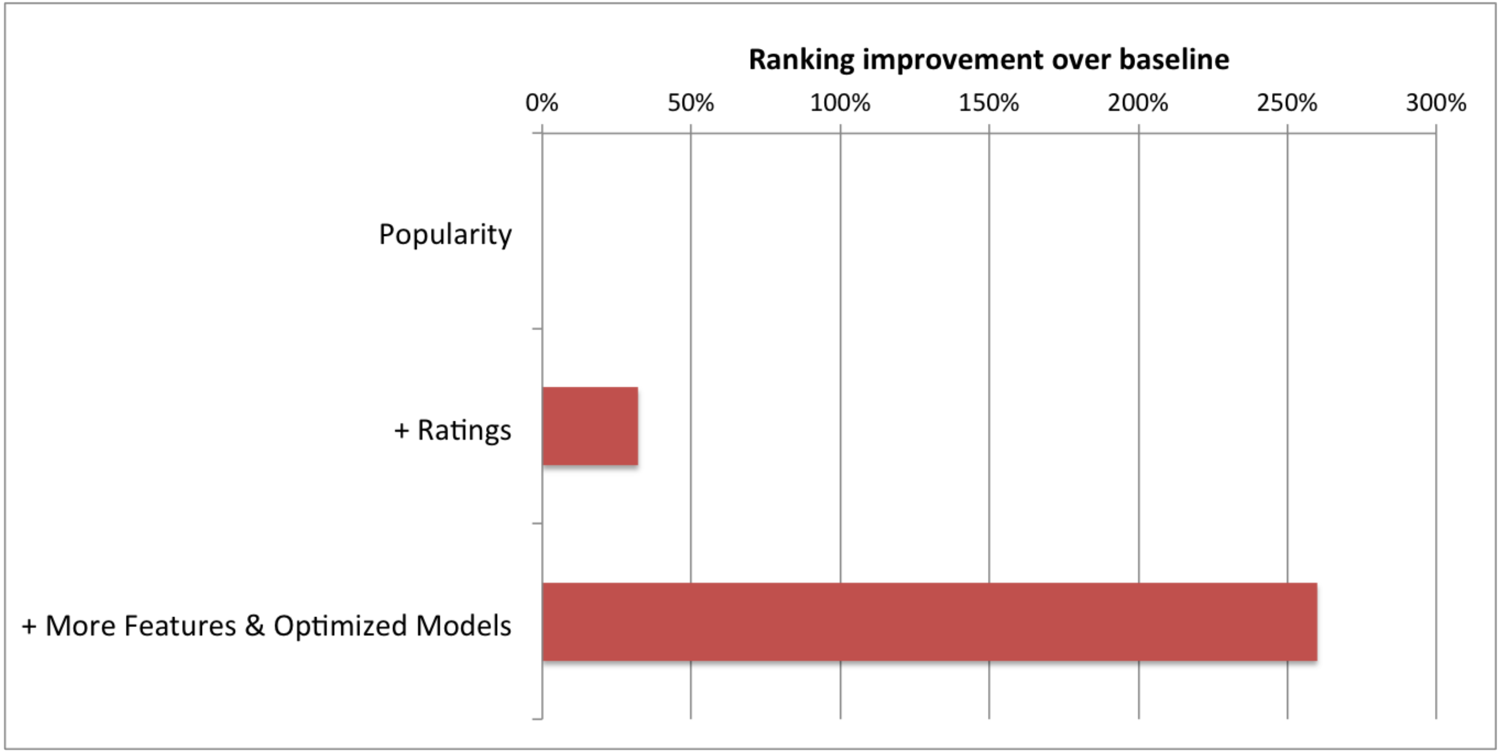
$f_{rank}(u,v) = w_1 p(v) + w_2 r(u,v) +$



Example: Two features, linear model



Ranking


Ranking improvement over baseline
0%
50%
100%
150%
200%
250%
300%
Popularity
+ Ratings
+ More Features & Optimized Models

Learning to Rank

- Ranking is a very important problem in many contexts (search, advertising, recommendations)
- Quality of ranking is measured using ranking metrics such as NDCG, MRR, MAP, FPC…
- It is hard to optimize machine-learned models directly on these measures
  - They are not differentiable
- We would like to use the same measure for optimizing and for the final evaluation

NETFLIX

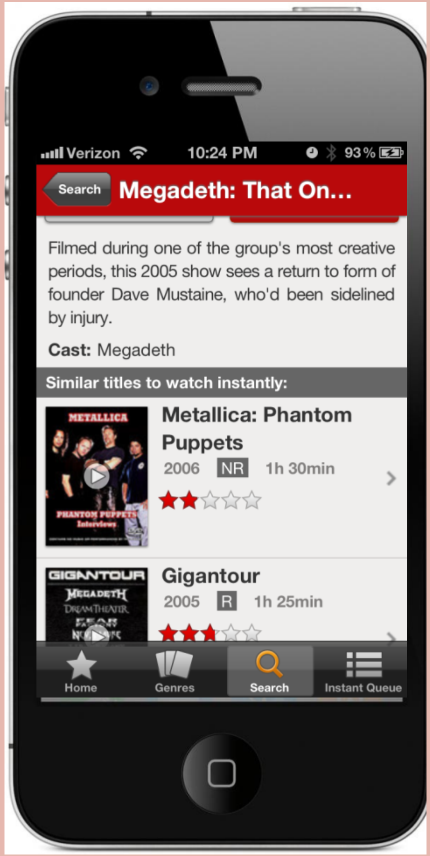
Learning to Rank Approaches

- ML problem: construct ranking model from training data

1. Pointwise (Ordinal regression, Logistic regression, SVM, GBDT, …)
   - Loss function defined on individual relevance judgment
2. Pairwise (RankSVM, RankBoost, RankNet, FRank…)
   - Loss function defined on pair-wise preferences
   - Goal: minimize number of inversions in ranking
3. Listwise
   - Indirect Loss Function (RankCosine, ListNet…)
   - Directly optimize IR measures (NDCG, MRR, FCP…)
     - Genetic Programming or Simulated Annealing
     - Use boosting to optimize NDCG (Adarank)
     - Gradient descent on smoothed version (CLiMF, TFMAP, GAPfm @cikm13)
     - Iterative Coordinate Ascent (Direct Rank @kdd13)

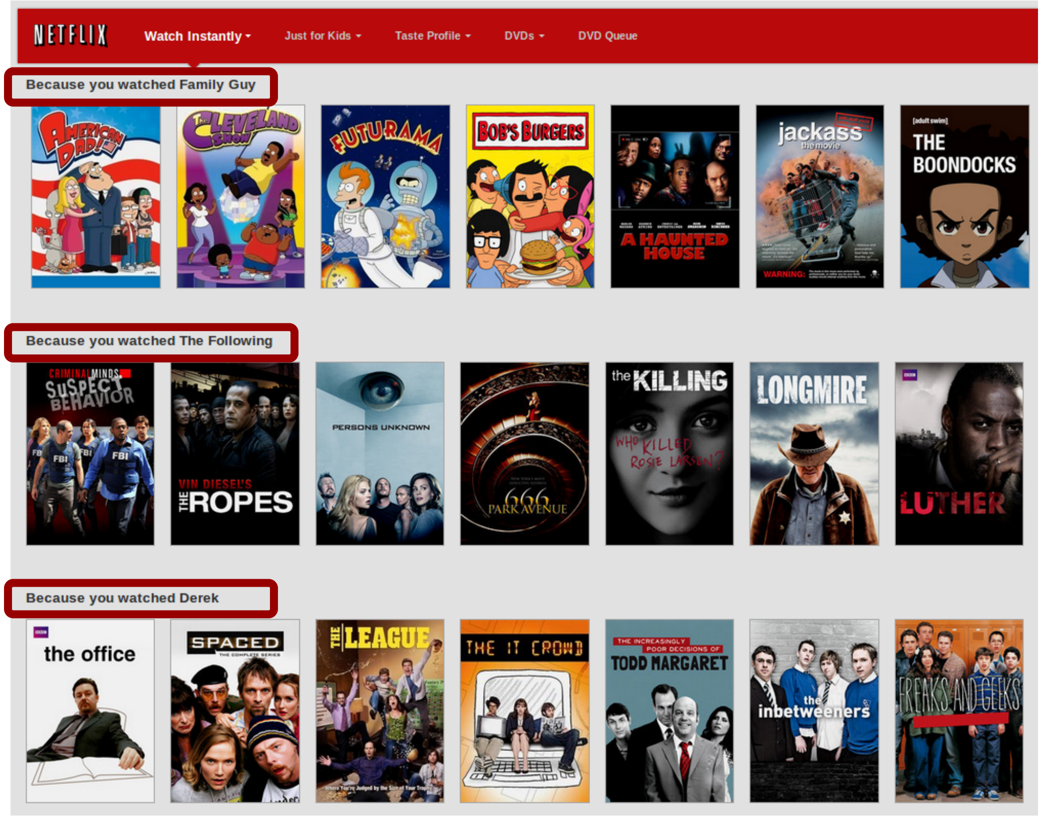
NETFLIX

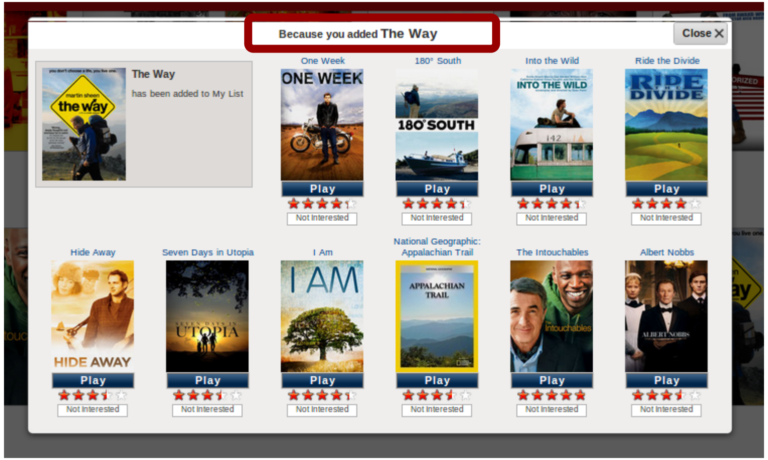
Similarity



NETFLIX

Similars

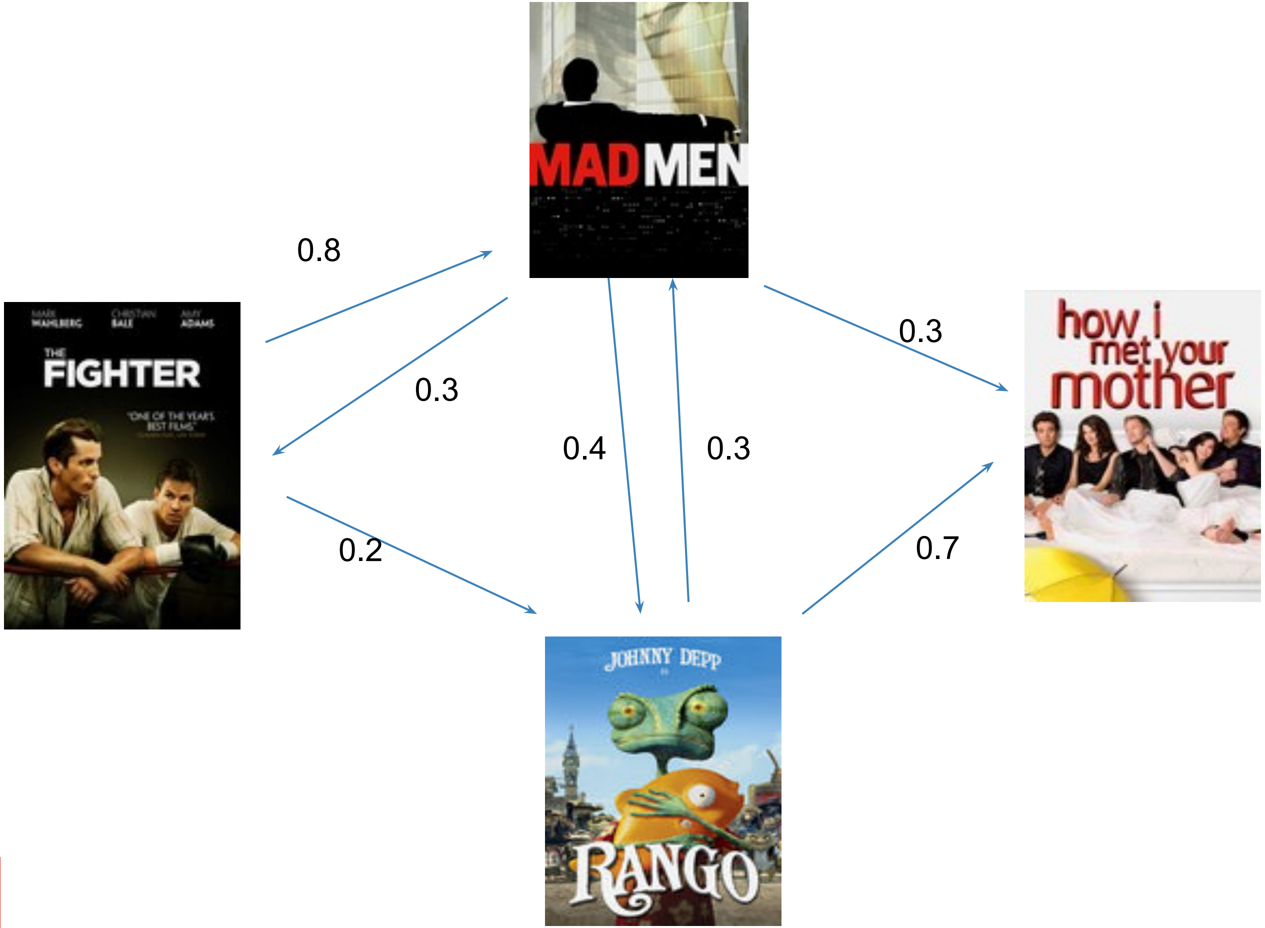

NETFLIX
Watch Instantly
Just for Kids
Taste Profile
DVDs
DVD Queue
Because you watched Family Guy
Because you watched The Following
Because you watched Derek


Because you added The Way
Close
The Way
has been added to My List
One Week
180° South
Into the Wild
Ride the Divide
Hide Away
Seven Days in Utopia
I Am
National Geographic: Appalachian Trail
The Intouchables
Albert Nobbs
Play
Not Interested

What is similarity?

- Similarity can refer to different dimensions
  - Similar in metadata/tags
  - Similar in user play behavior
  - Similar in user rating behavior
  - …
- You can learn a model for each of them and finally learn an ensemble

Graph-based similarities



Example of graph-based similarity: SimRank

- SimRank (Jeh & Widom, 02): “two objects are similar if they are referenced by similar objects.”

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

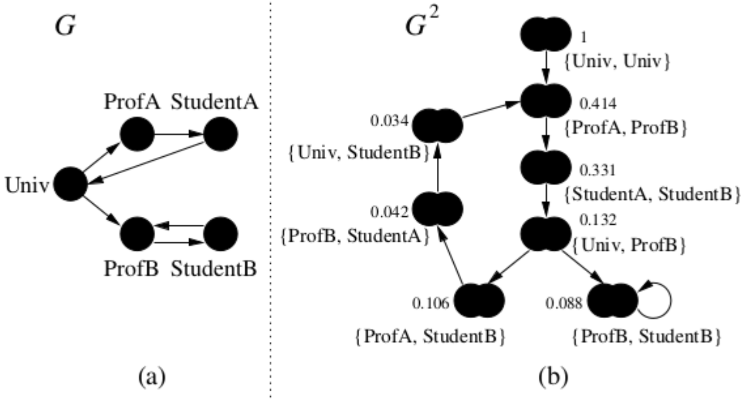
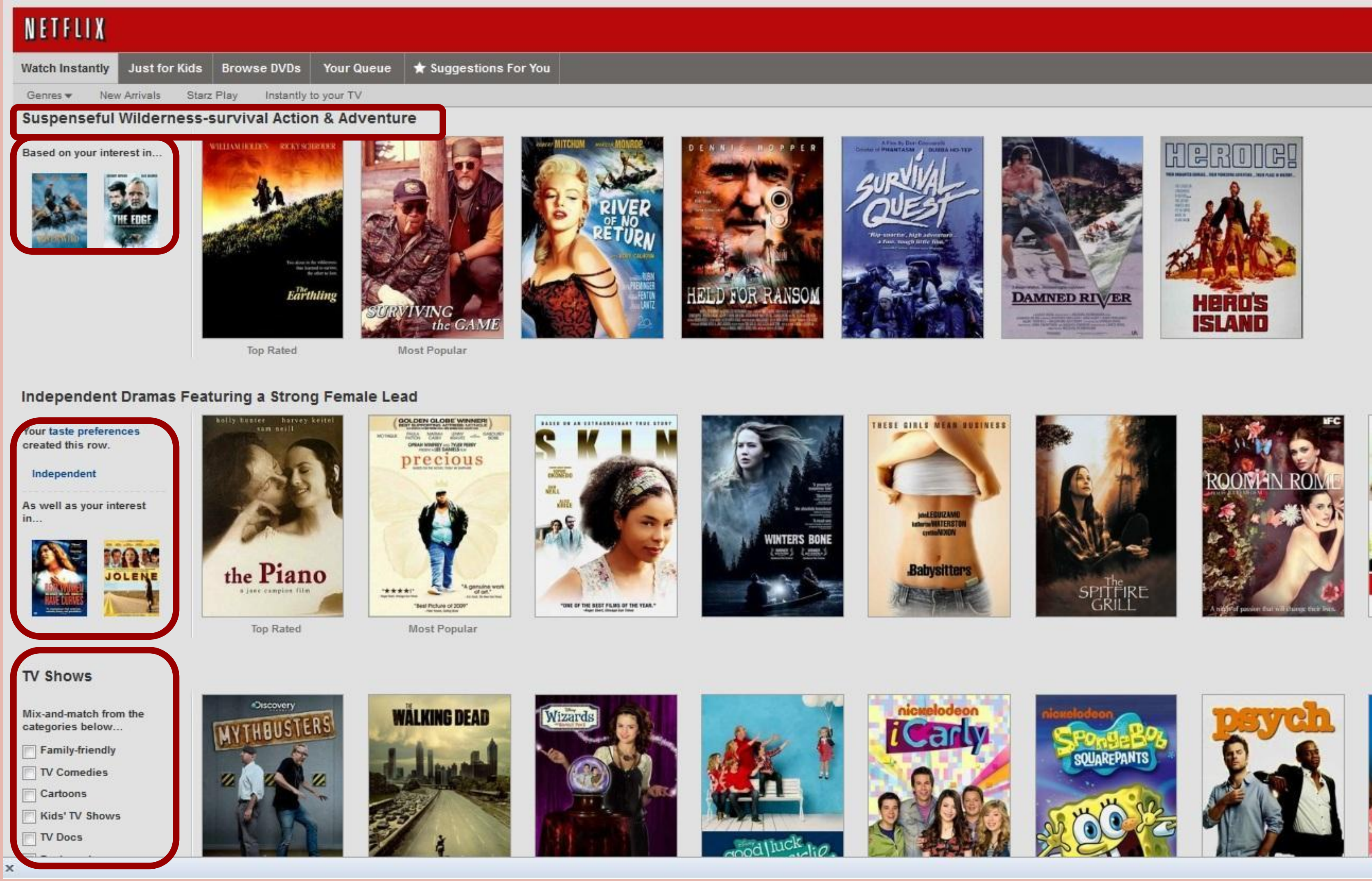


Figure 1: A small Web graph G and simplified node-pairs graph G^2. SimRank scores using parameter $C = 0.8$ are shown for nodes in G^2.

Final similarity ensemble

- Come up with a score of play similarity, rating similarity, tag-based similarity…
- Combine them using an ensemble
  - Weights are learned using regression over existing response
- The final concept of “similarity” responds to what users vote as similar

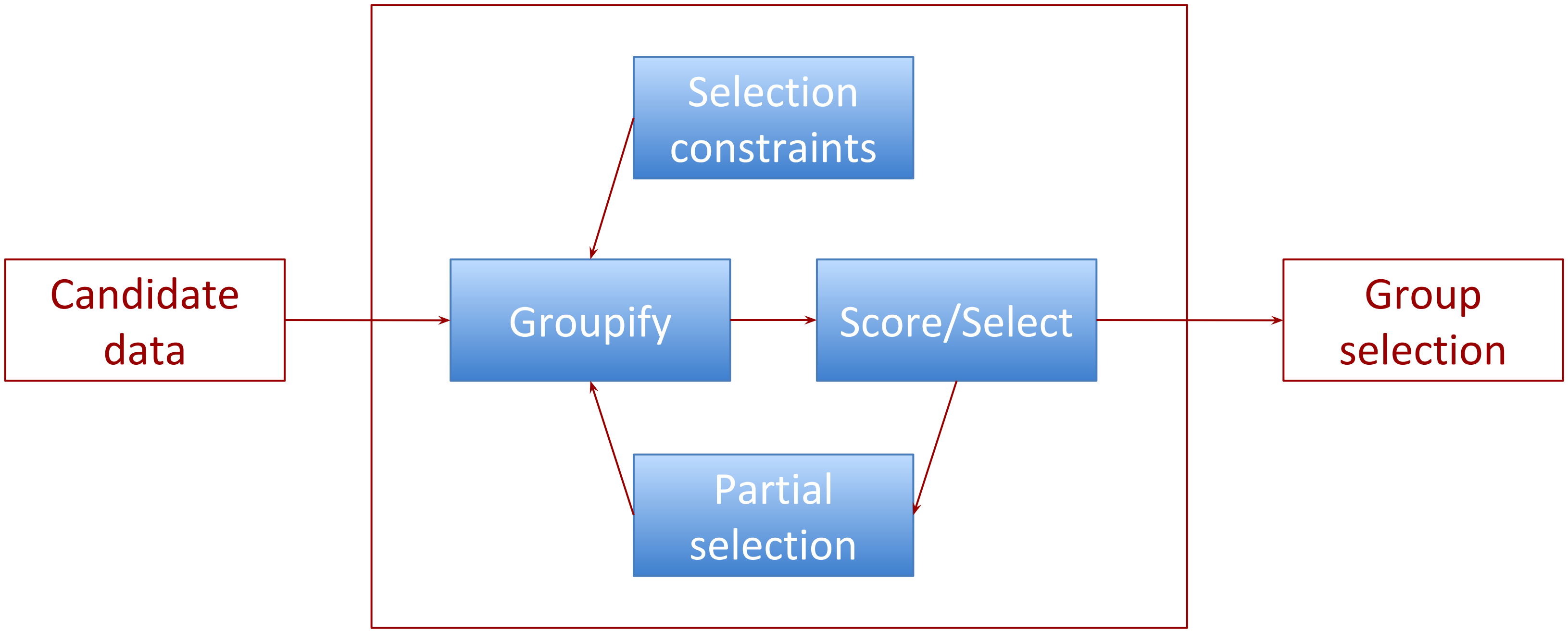
Row Selection



Row Selection Inputs

- Visitor data
  - Video history
  - Queue adds
  - Ratings
  - Taste preferences
  - Predicted ratings
  - PVR ranking
  - etc.
- Global data
  - Metadata
  - Popularity
  - etc.

Row Selection Core Algorithm

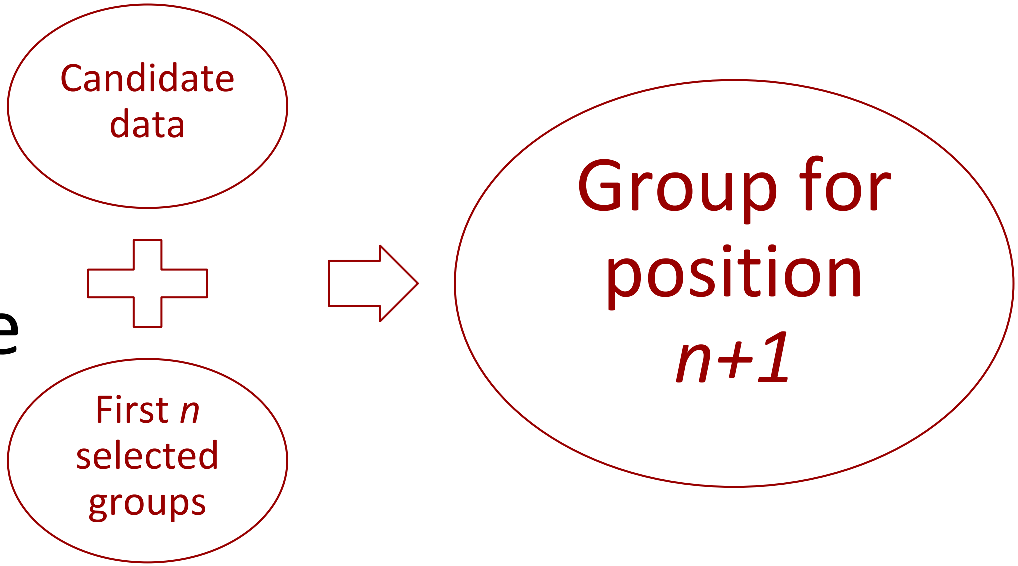


Selection Constraints

- Many business rules and heuristics
  - Relative position of groups
  - Presence/absence of groups
  - Deduping rules
  - Number of groups of a given type
- Determined by past A/B tests
- Evolved over time from a simple template

Groupification

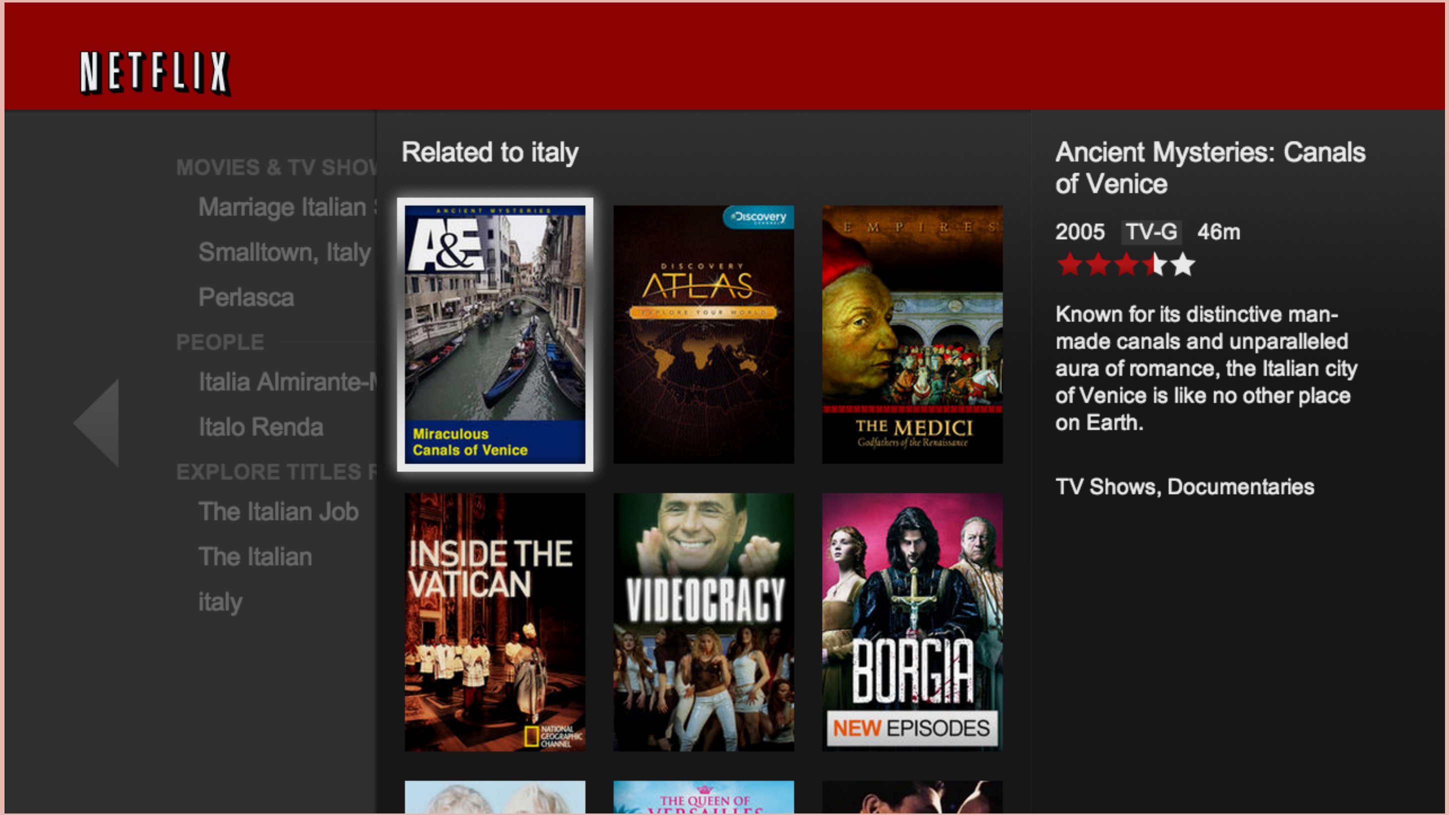
- Ranking of titles according to ranking algo
- Filtering
- Deduping
- Formatting
- Selection of evidence
- ...



Scoring and Selection

- Features considered
  - Quality of items
  - Diversity of tags
  - Quality of evidence
  - Freshness
  - Recency
  - …
  - ARO scorer
  - Greedy algorithm
  - Framework supports other methods
    - Backtracking
    - Approximate integer programming
    - etc.

Search Recommendations

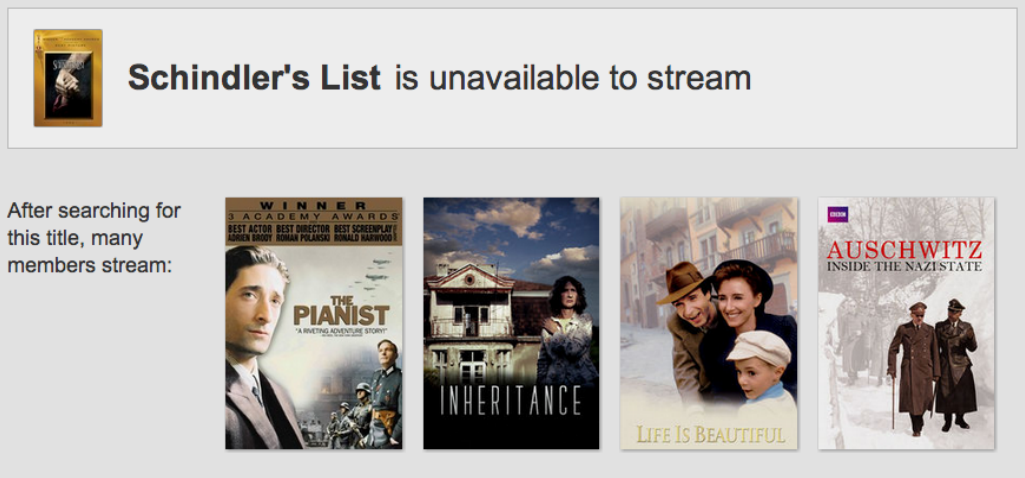


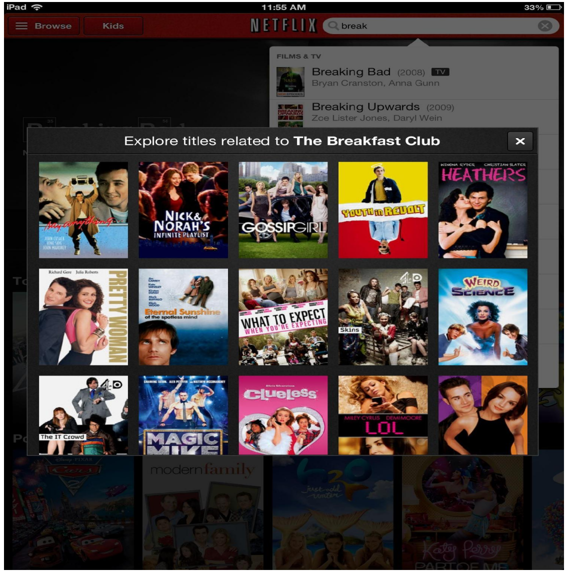
Search Recommendation

Combination of PAS+PRS model

- <u>**P**</u>lay-<u>**A**</u>fter-<u>**S**</u>earch and <u>**P**</u>lay-<u>**R**</u>elated-<u>**S**</u>earch
- PAS: Transition on Play after Query
- PRS: Similarity between user's (query, play)

Unavailable Title Recommendations





Gamification Algorithms

Rating Game

Mood Selection Algorithm

1. Editorially selected moods
2. Sort them according to users consumption
3. Take into account freshness, diversity...

NETFLIX

More data or better models?

WIRED MAGAZINE: 16.07

SCIENCE : DISCOVERIES

The End of Theory: The Data Deluge Makes the Scientific Method Obsolete

By Chris Anderson 06.23.08

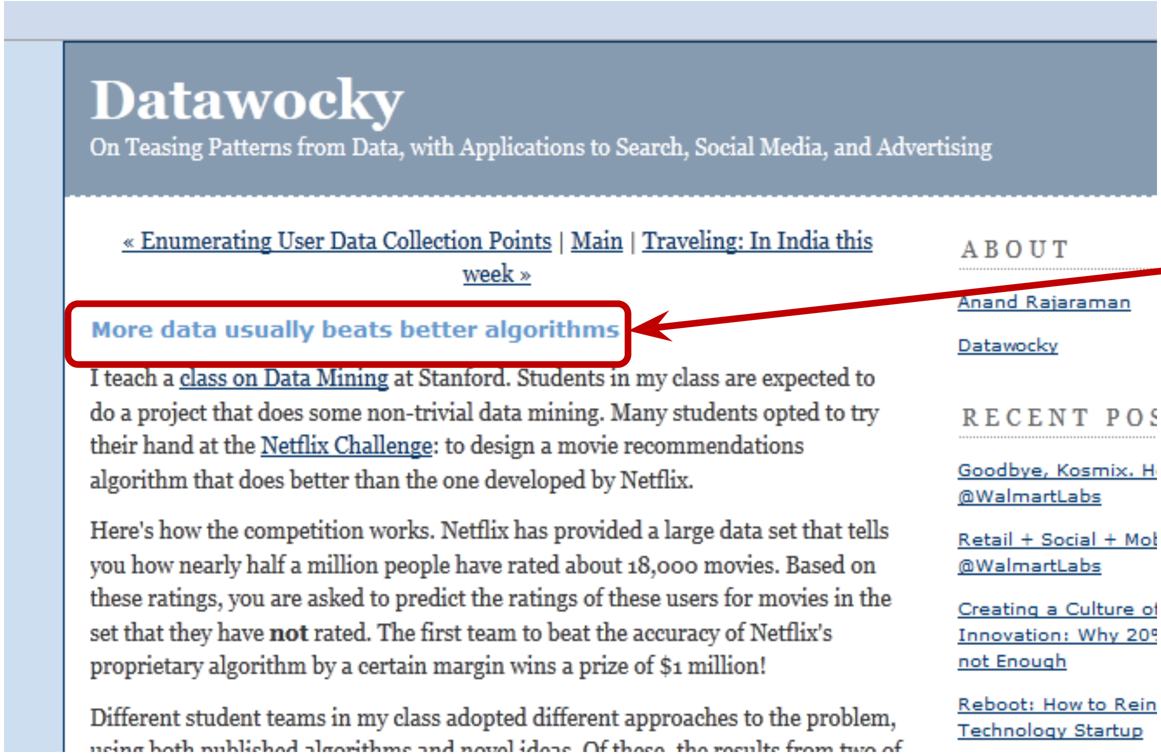
Illustration: Marian Bantjes

THE PETABYTE AGE:
Sensors everywhere. Infinite storage. Clouds of processors. Our ability to capture, warehouse,

"**All models are wrong**, but some are useful."

So proclaimed statistician George Box 30 years ago, and

More data or better models?



Datawocky

On Teasing Patterns from Data, with Applications to Search, Social Media, and Advertising

« Enumerating User Data Collection Points | Main | Traveling: In India this week »

More data usually beats better algorithms

I teach a class on Data Mining at Stanford. Students in my class are expected to do a project that does some non-trivial data mining. Many students opted to try their hand at the Netflix Challenge: to design a movie recommendations algorithm that does better than the one developed by Netflix.

Here's how the competition works. Netflix has provided a large data set that tells you how nearly half a million people have rated about 18,000 movies. Based on these ratings, you are asked to predict the ratings of these users for movies in the set that they have **not** rated. The first team to beat the accuracy of Netflix's proprietary algorithm by a certain margin wins a prize of $1 million!

Different student teams in my class adopted different approaches to the problem,

ABOUT

Anand Rajaraman

Datawocky

RECENT POS

Goodbye, Kosmix. H @WalmartLabs

Retail + Social + Mol @WalmartLabs

Creating a Culture of Innovation: Why 20% not Enough

Reboot: How to Rein Technology Startup

Really?

Anand Rajaraman: Former Stanford Prof. & Senior VP at Walmart

More data or better models?

Sometimes, it's not about more data

Recommending New Movies: Even a Few Ratings Are More Valuable Than Metadata

István Pilászy *
Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
pila@mit.bme.hu

Domonkos Tikk *,†
Dept. of Telecom. and Media Informatics
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
tikk@tmit.bme.hu

ABSTRACT

The Netflix Prize (NP) competition gave much attention to collaborative filtering (CF) approaches. Matrix factorization (MF) based CF approaches assign low dimensional feature vectors to users and items. We link CF and content-based filtering (CBF) by finding a linear transformation that transforms user or item descriptions so that they are as close as possible to the feature vectors generated by MF for CF.

We propose methods for explicit feedback that are able to handle 140 000 features when feature vectors are very sparse. With movie metadata collected for the NP movies we show that the prediction performance of the methods is comparable to that of CF, and can be used to predict user preferences on new movies.

We also investigate the value of movie metadata compared to movie ratings in regards of predictive power. We compare

1. INTRODUCTION

The goal of recommender systems is to give personalized recommendation on items to users. Typically the recommendation is based on the former and current activity of the users, and metadata about users and items, if available.

There are two basic strategies that can be applied when generating recommendations. Collaborative filtering (CF) methods are based only on the activity of users, while content-based filtering (CBF) methods use only metadata. In this paper we propose hybrid methods, which try to benefit from both information sources.

The two most important families of CF methods are matrix factorization (MF) and neighbor-based approaches. Usually, the goal of MF is to find a low dimensional representation for both users and movies, i.e. each user and movie is associated with a feature vector. Movie metadata (which

More data or better models?

Norvig: “Google does not have better Algorithms, only more Data”

The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, *Google*

Many features/ low-bias models

[Banko and Brill, 2001]

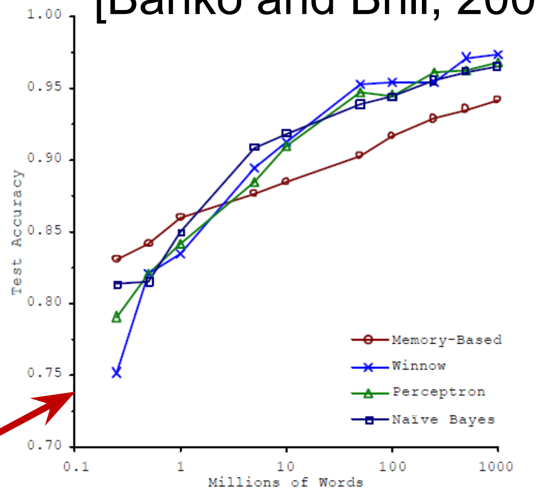
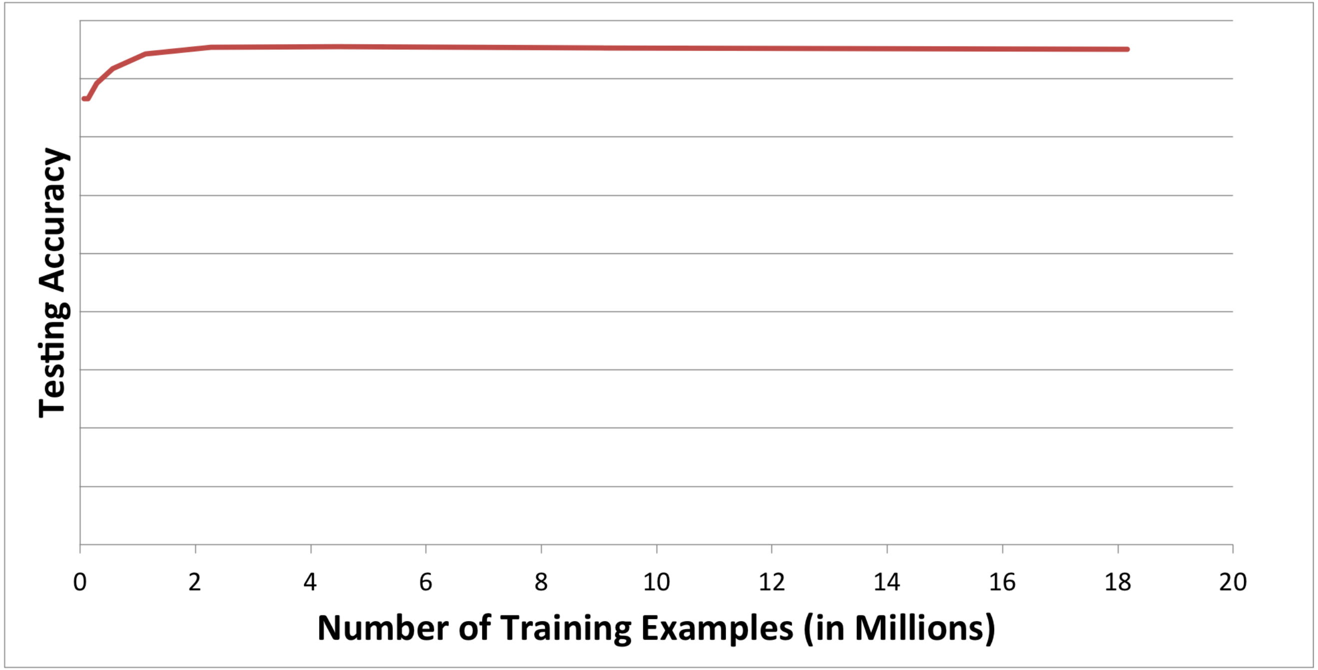


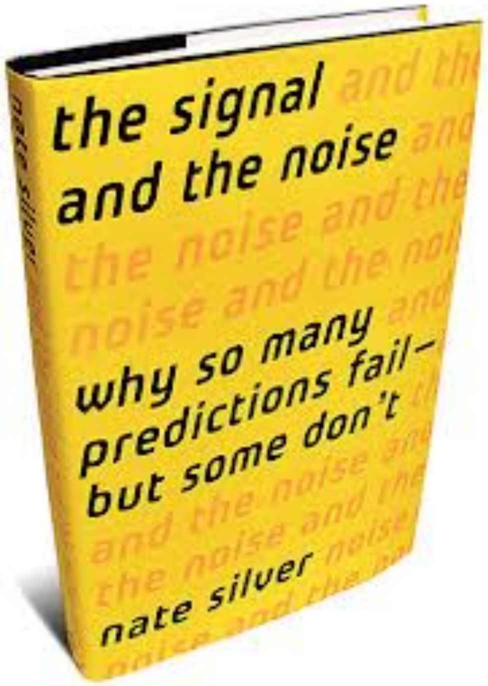
Figure 1. Learning Curves for Confusion Set Disambiguation

More data or better models?



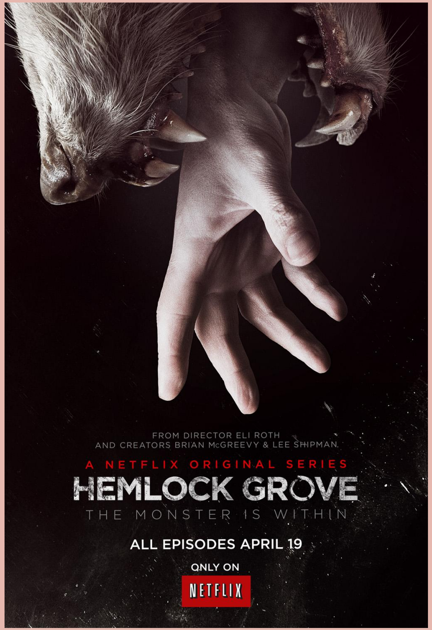
Sometimes, it's not about more data

"Data without a sound approach = noise"



NETFLIX

Conclusion


FROM DIRECTOR ELI ROTH
AND CREATORS BRIAN McGREEVY & LEE SHIPMAN
A NETFLIX ORIGINAL SERIES
HEMLOCK GROVE
THE MONSTER IS WITHIN
ALL EPISODES APRIL 19
ONLY ON
NETFLIX

More data +

Smarter models +

More accurate metrics +

Better system architectures

Lots of room for improvement!

Xavier Amatriain (@xamat)
xavier@netflix.com

Thanks!

NETFLIX

We’re hiring!