Machine Learning & Recommender Systems @ Netflix Scale

November, 2013

Xavier Amatriain
Director - Algorithms Engineering @ Netflix

@xamat

Netflix Prize

COMPLETED

What we were interested in:

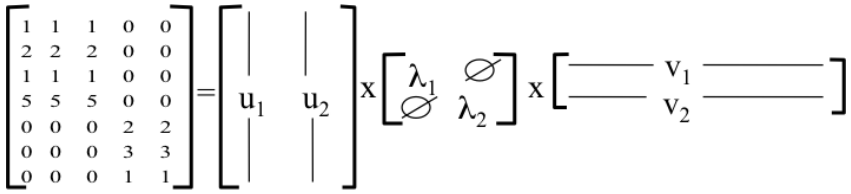
- High quality *recommendations*

Proxy question:

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

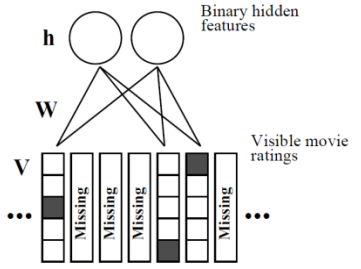
$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

SVD



Results

- Top 2 algorithms still in production



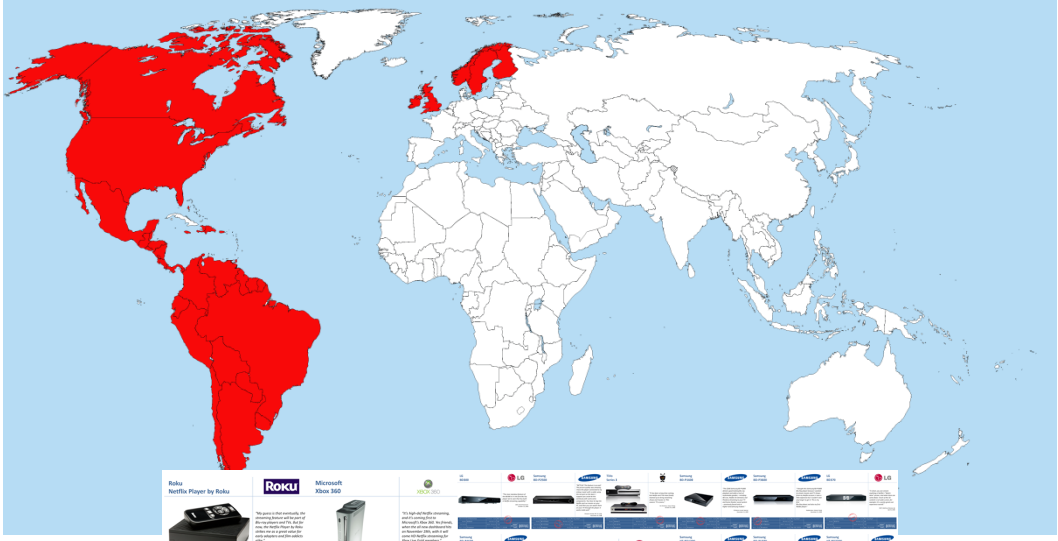
RBM

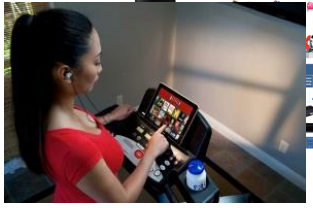
NETFLIX

From the Netflix Prize to today

NETFLIX

2006





2013

amazon web services™

Everything is

ARRESTED DEVELOPMENT™

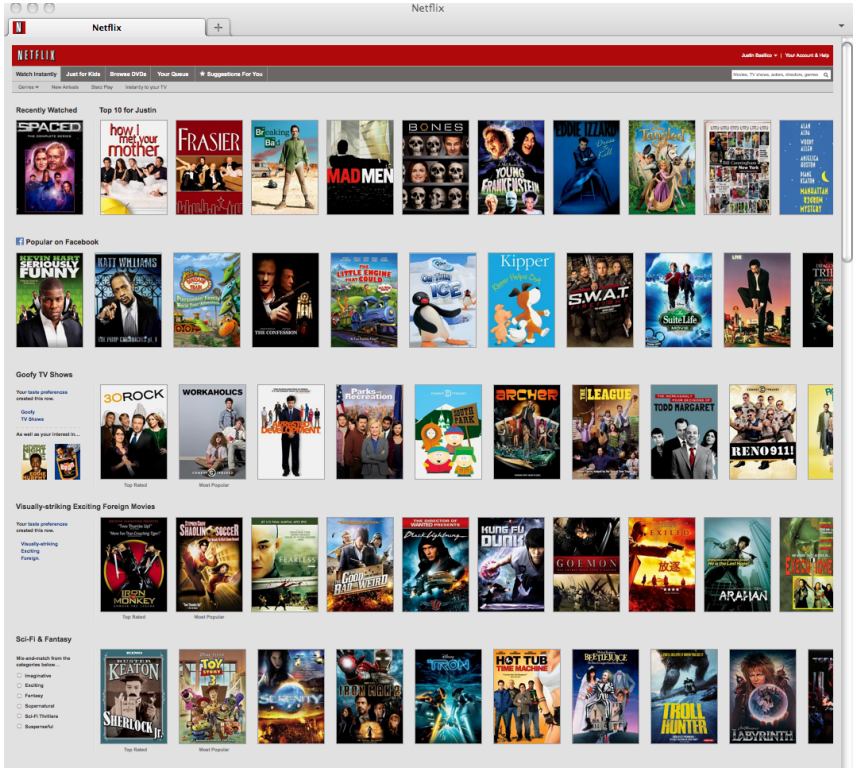
Personalized

NETFLIX™

Everything is personalized

Ranking

Rows

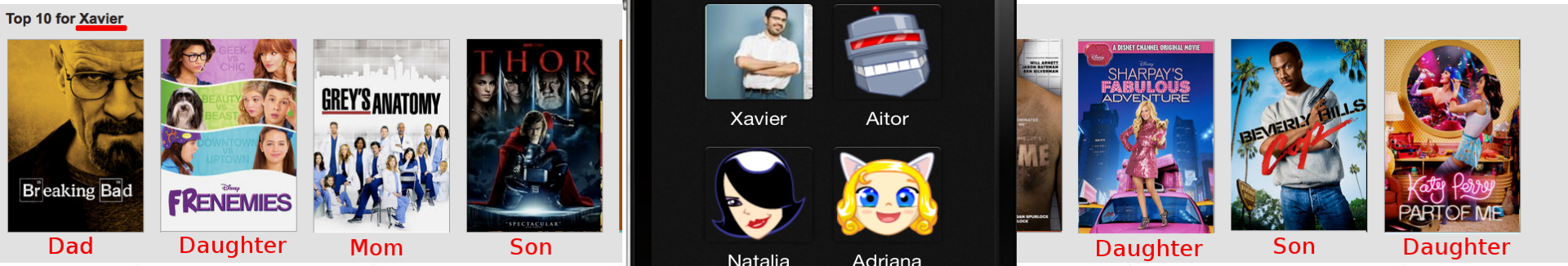


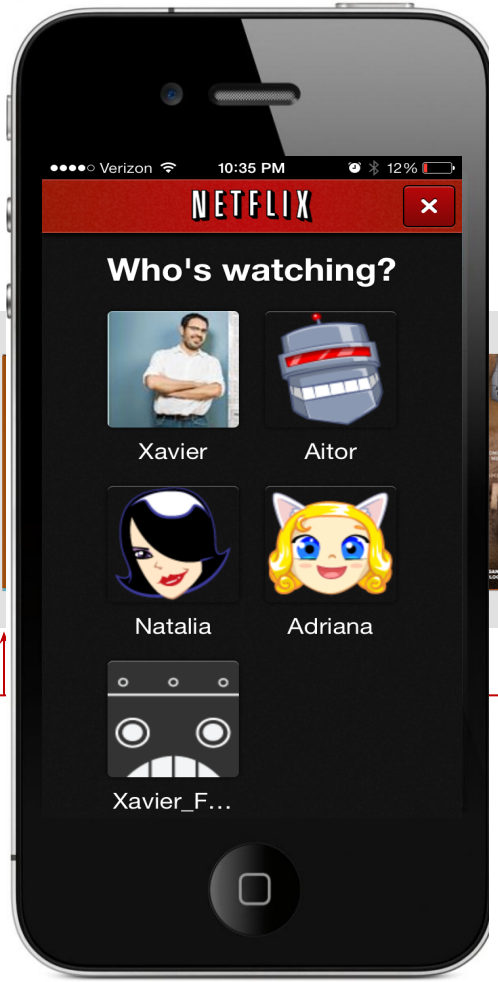
Over **75%** of what people watch comes from a recommendation

NETFLIX

Top 10

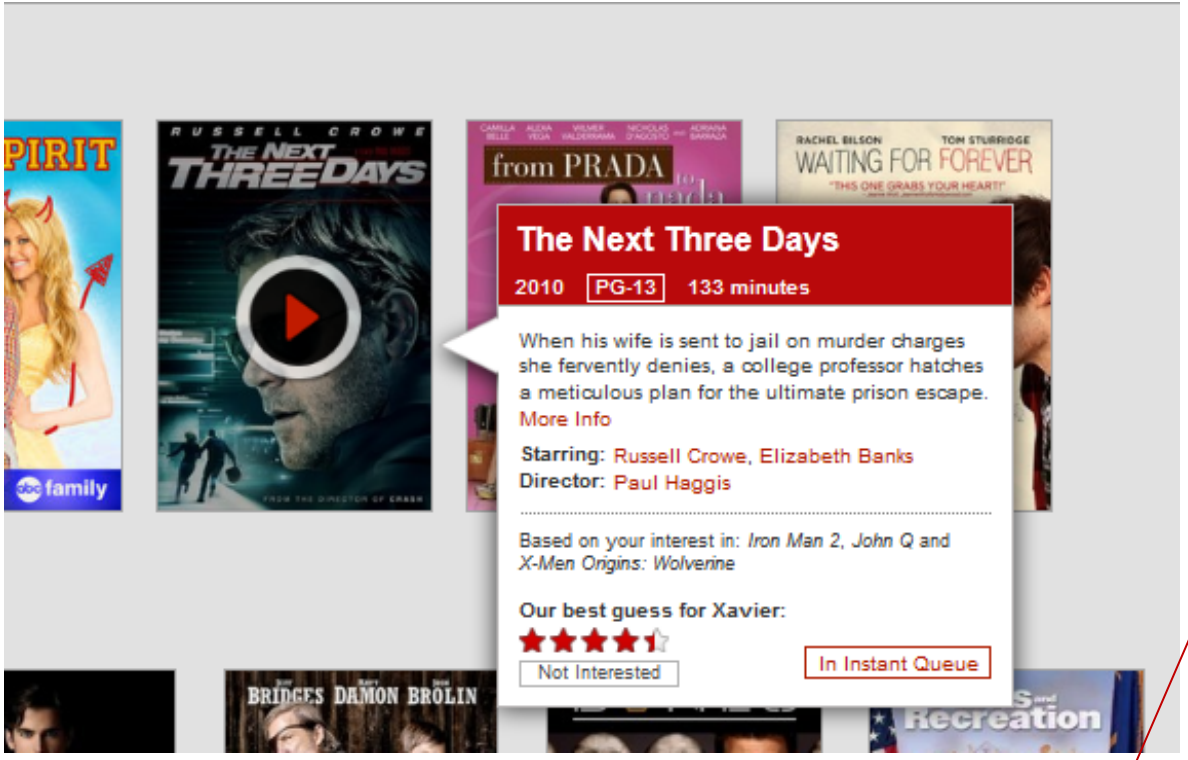
Personalization awareness

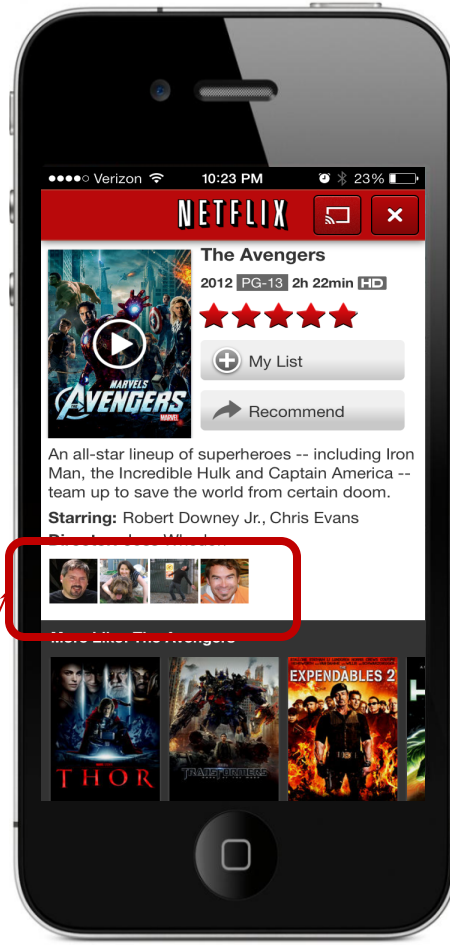




NETFLIX

Support for Recommendations





Social Support

NETFLIX

Genre Rows

NETFLIX

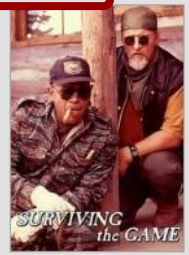
Watch Instantly | Just for Kids | Browse DVDs | Your Queue | ★ Suggestions For You

Genres ▾ | New Arrivals | Starz Play | Instantly to your TV

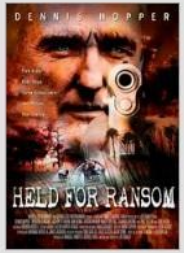
Suspenseful Wilderness-survival Action & Adventure

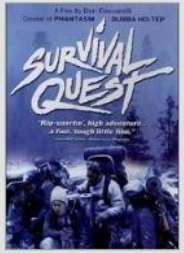
Based on your interest in...

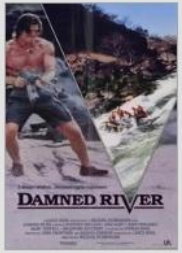

Top Rated

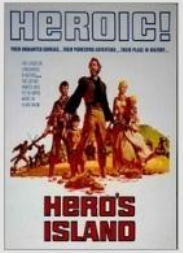


Most Popular







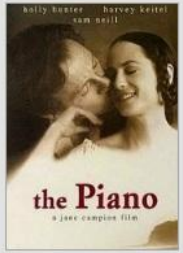


Independent Dramas Featuring a Strong Female Lead

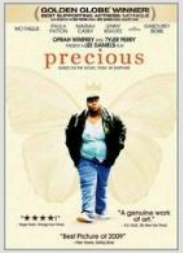
Your taste preferences created this row.

Independent

As well as your interest in...

Top Rated

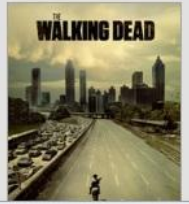


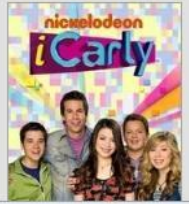
Most Popular

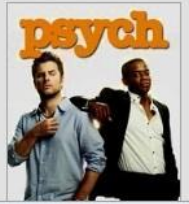
TV Shows

Mix-and-match from the categories below...

- [ ] Family-friendly
- [ ] TV Comedies
- [ ] Cartoons
- [ ] Kids' TV Shows
- [ ] TV Docs

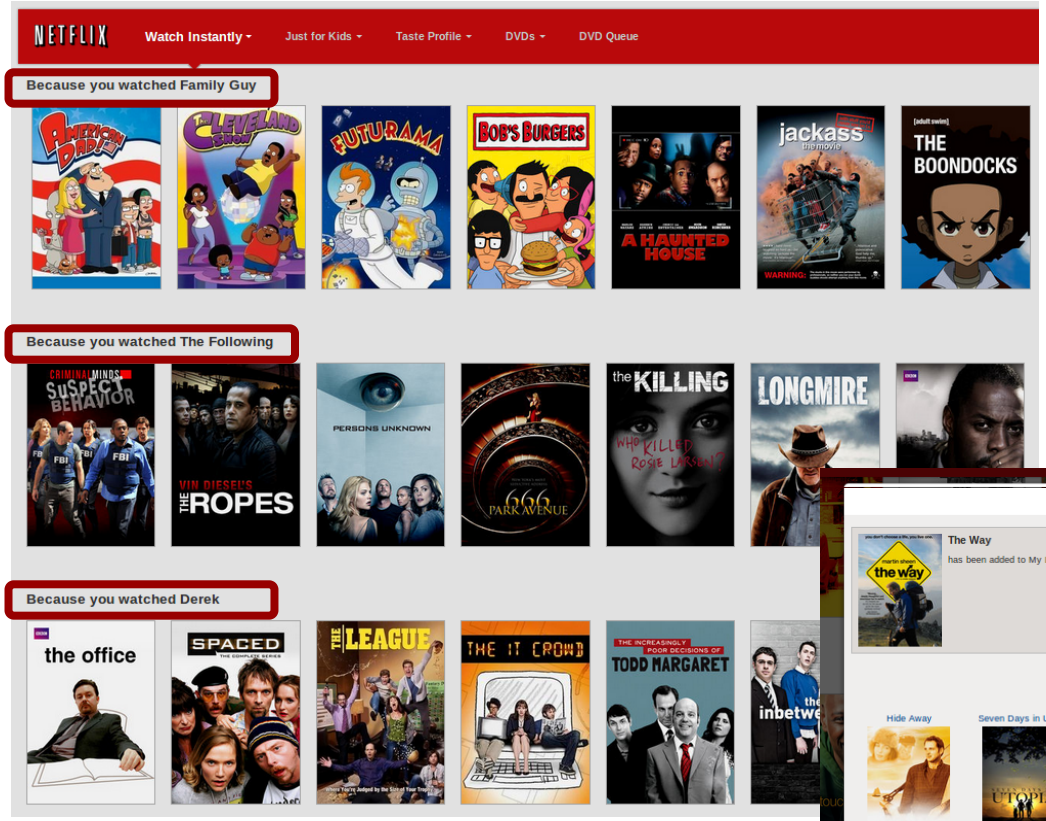


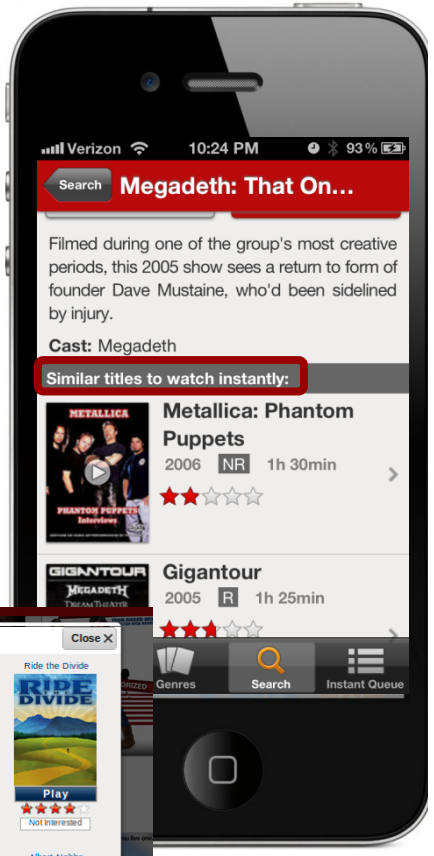


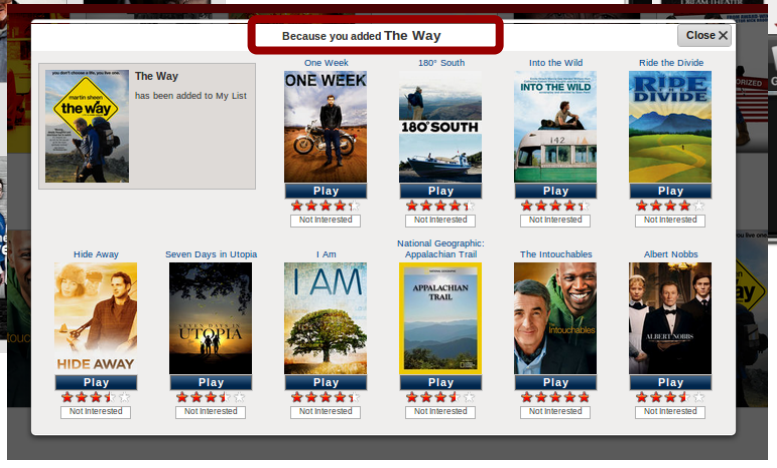


NETFLIX

Similars

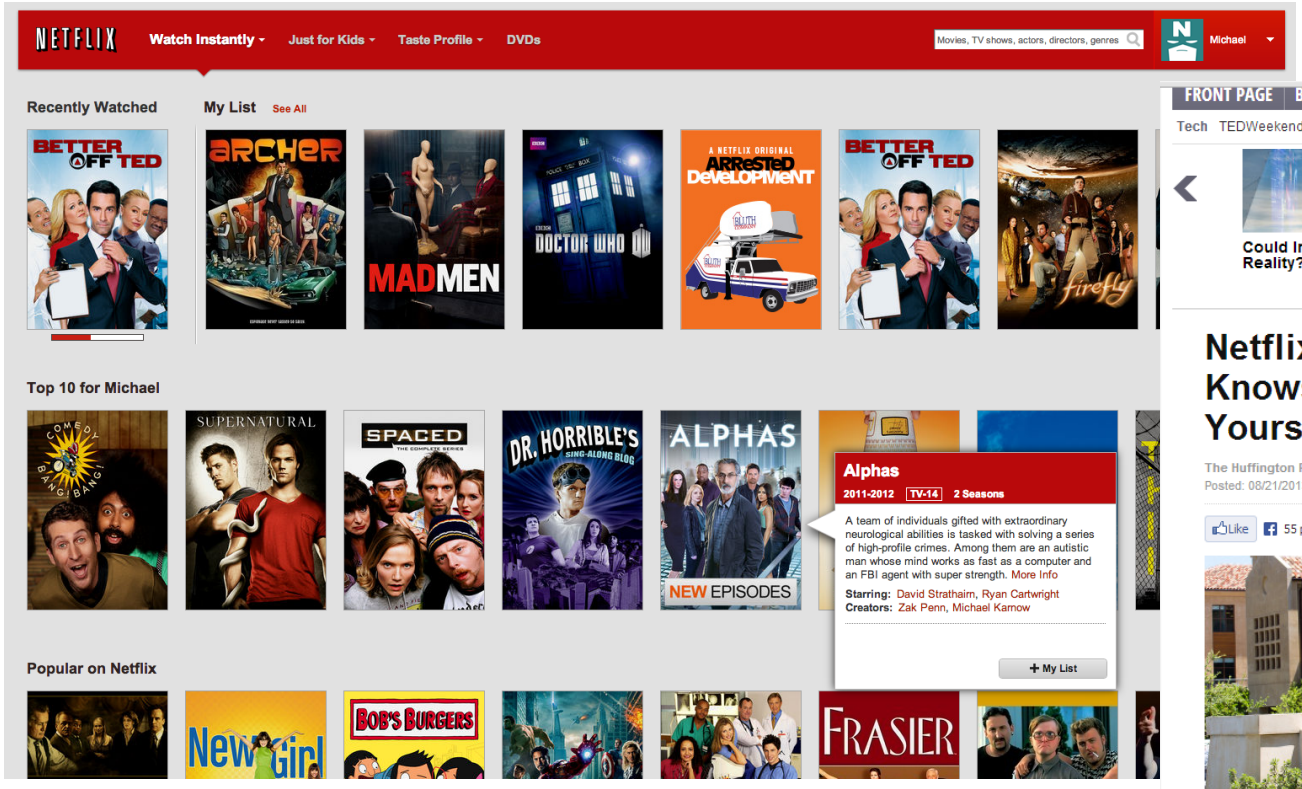


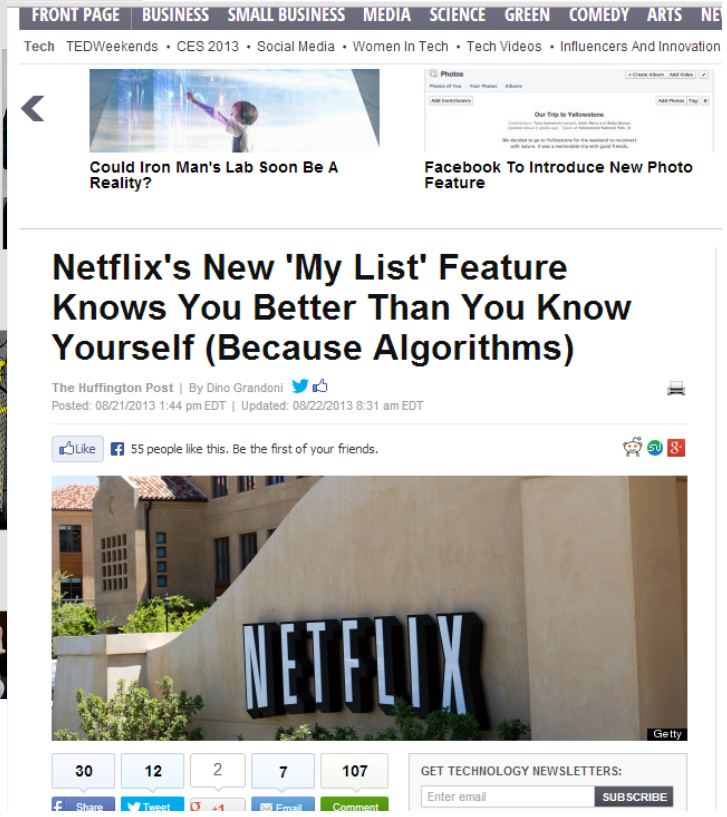




NETFLIX

EVERYTHING is a Recommendation





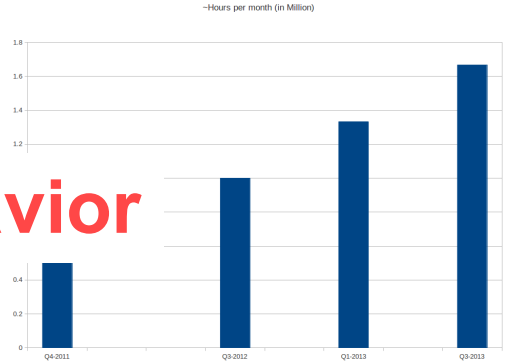
NETFLIX

Data & Models

NETFLIX

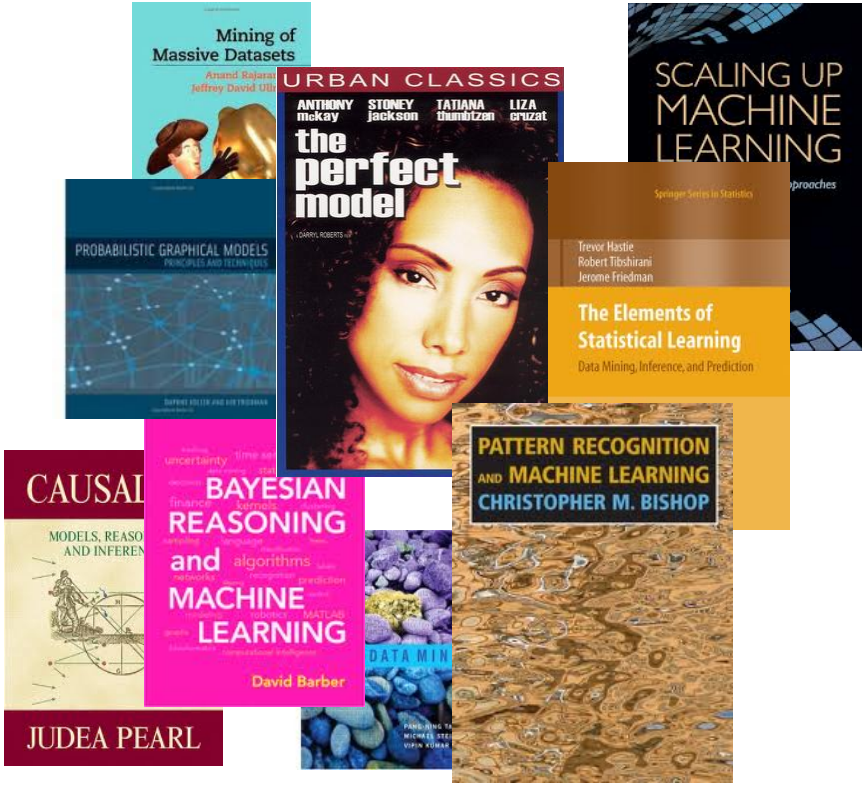
Big Data @Netflix

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



NETFLIX

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
- …

NETFLIX

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 $\hat{r}_{uv} = b_{uv} + p_u^T q_v$
- SVD++ (Koren et. Al) asymmetric variation w. implicit feedback

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

- 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)

NETFLIX

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) – 4th 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

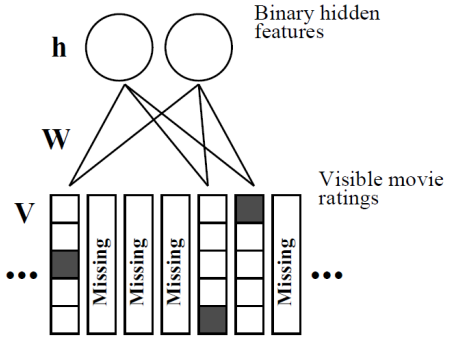


Figure 1. A restricted Boltzmann machine with binary hidden units and softmax visible units. For each user, the RBM only includes softmax units for the movies that user has rated. In addition to the symmetric weights between each hidden unit and each of the $K = 5$ values of a softmax unit, there are 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.

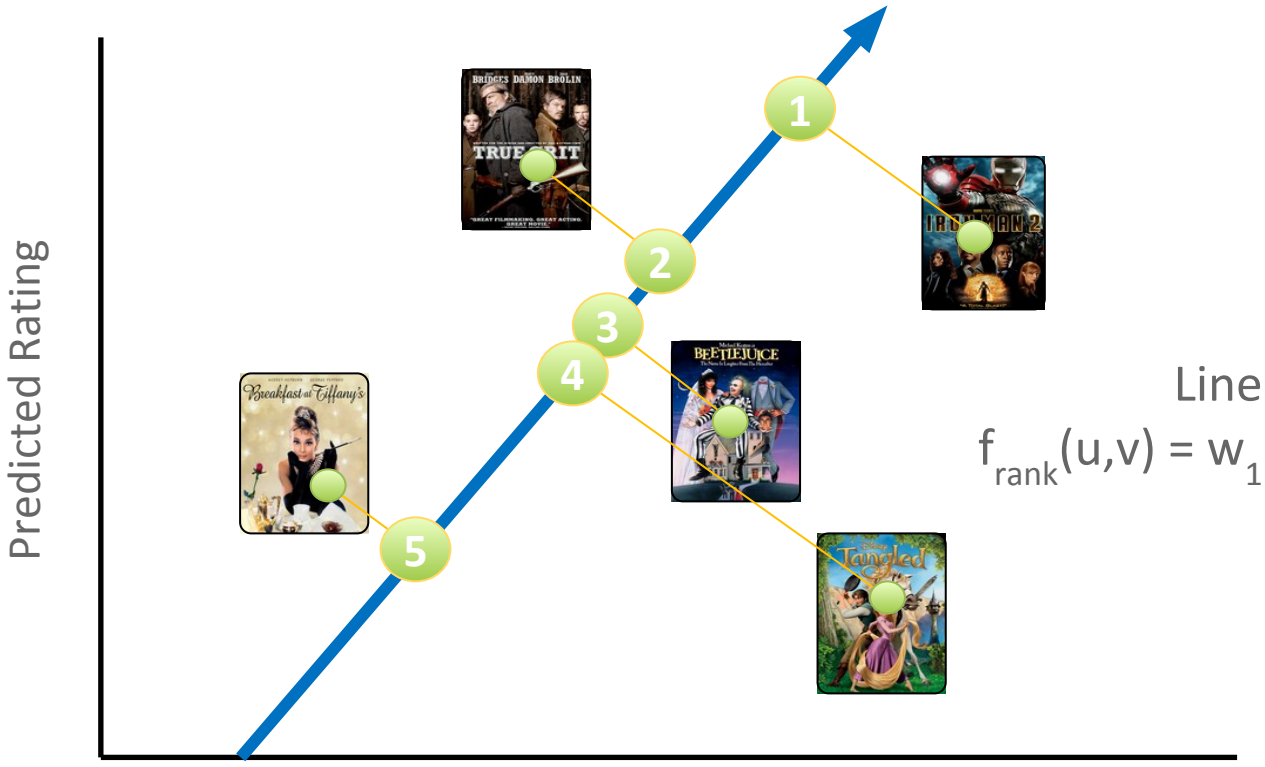
Ranking

- 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



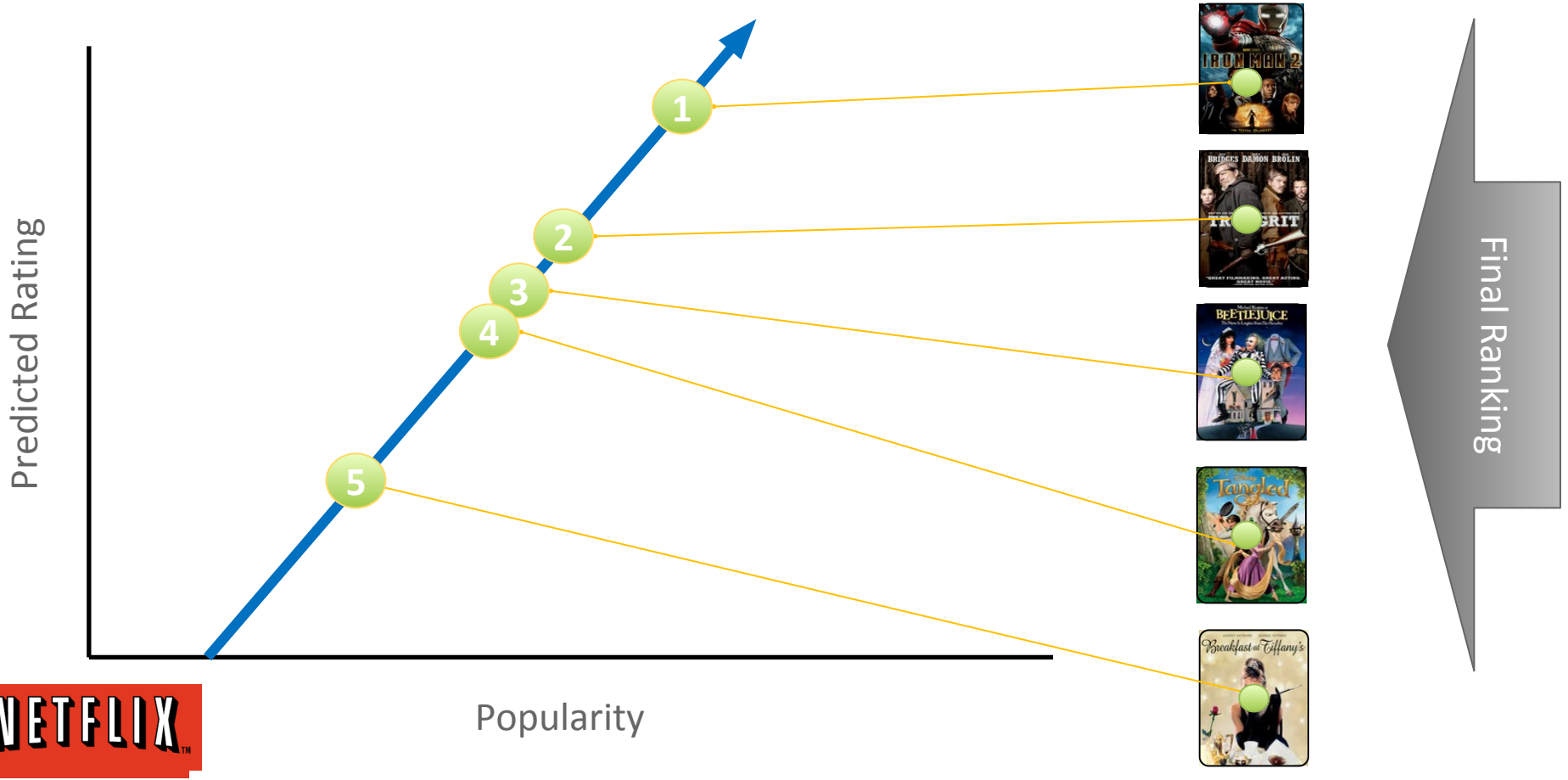
Linear Model:

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

Final Ranking

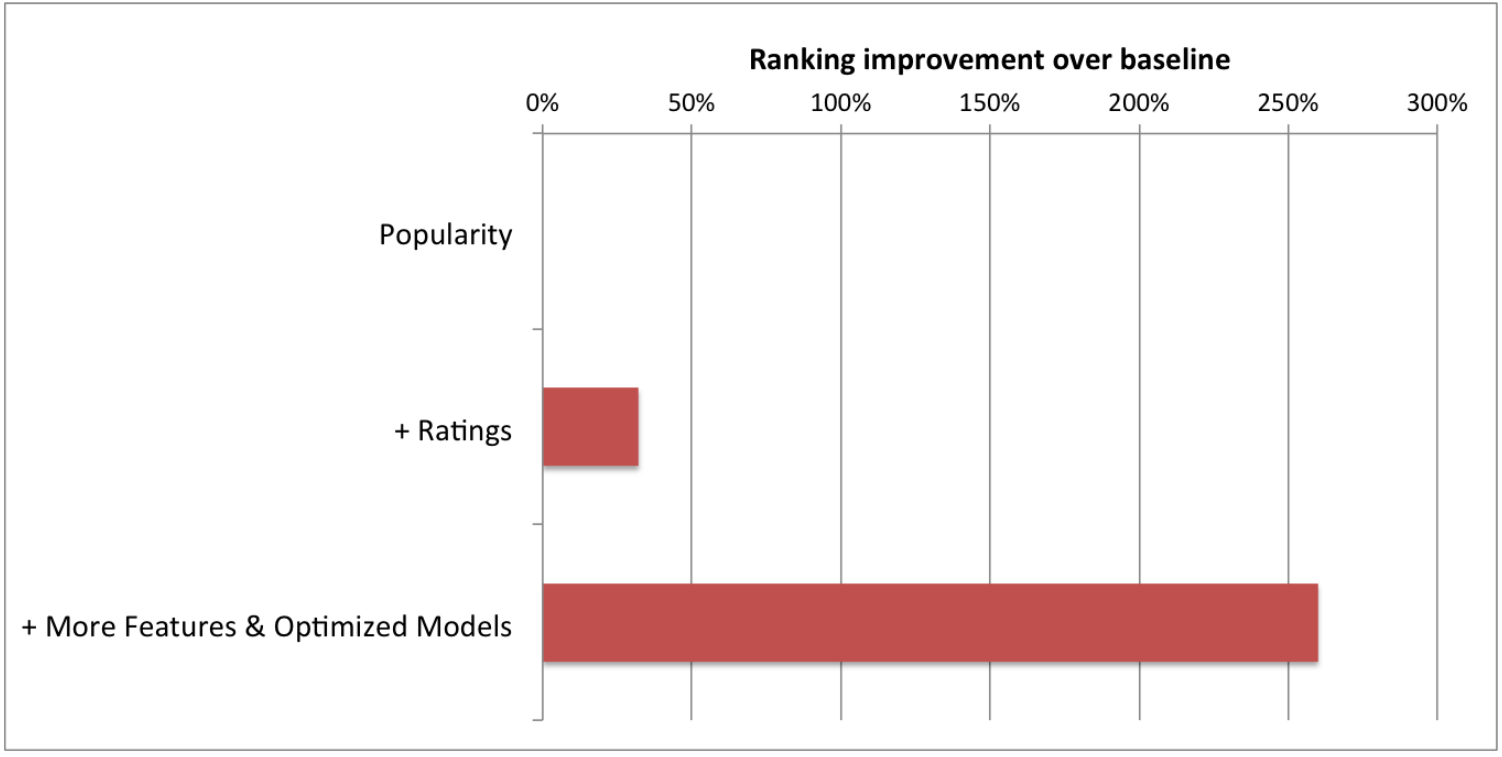
NETFLIX

Example: Two features, linear model



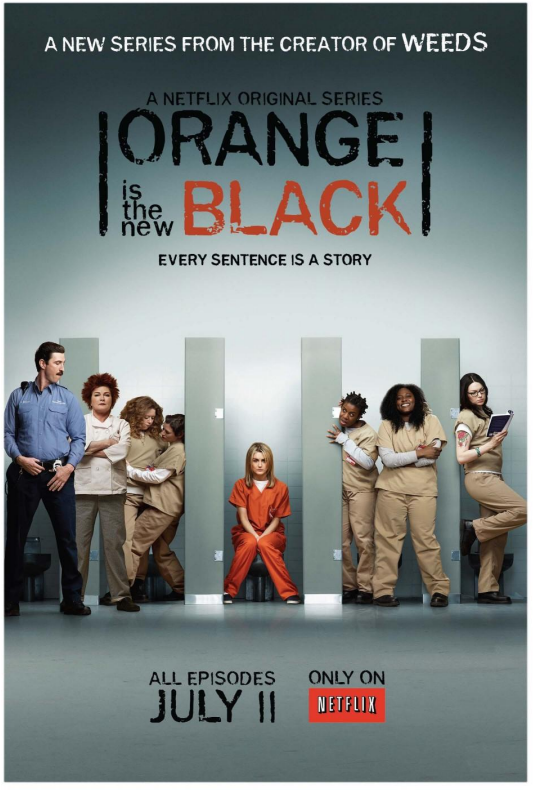
NETFLIX

Ranking



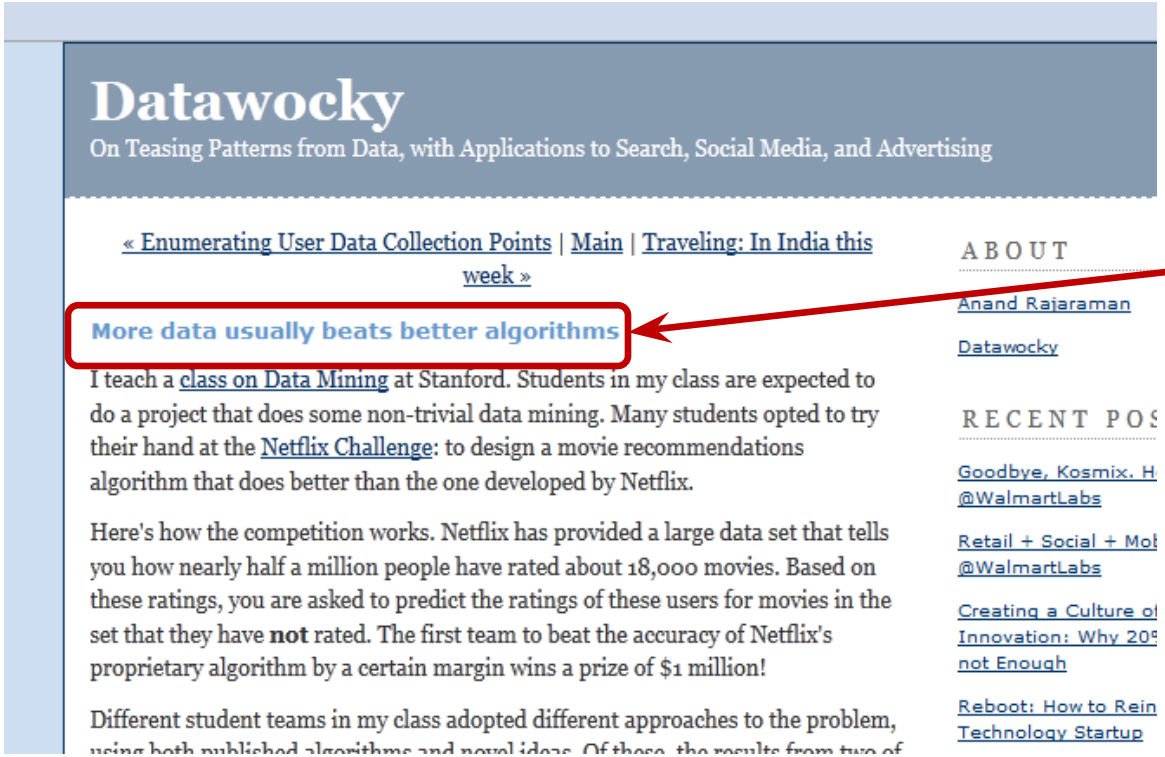
NETFLIX

More data or better models?



NETFLIX

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 + Mob @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

NETFLIX

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

NETFLIX

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]

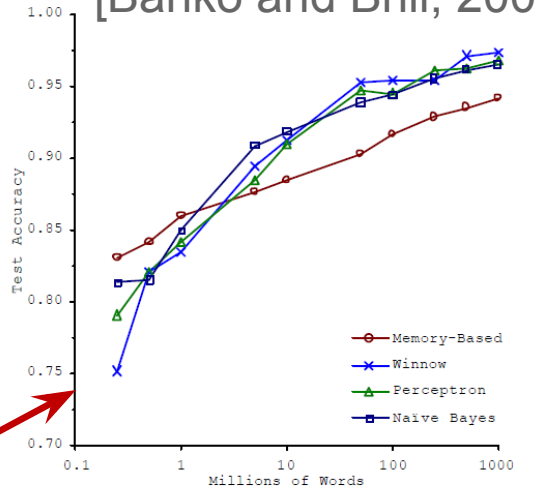


Figure 1. Learning Curves for Confusion Set Disambiguation

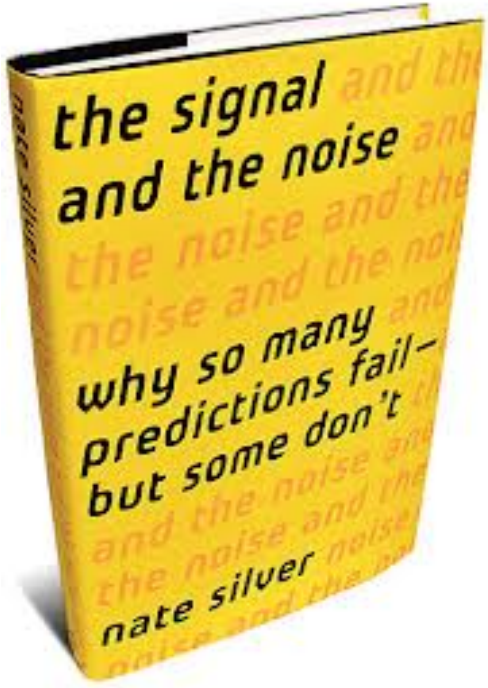
NETFLIX

More data or better models?

Sometimes, it's not about more data

NETFLIX

“Data without a sound approach = noise”

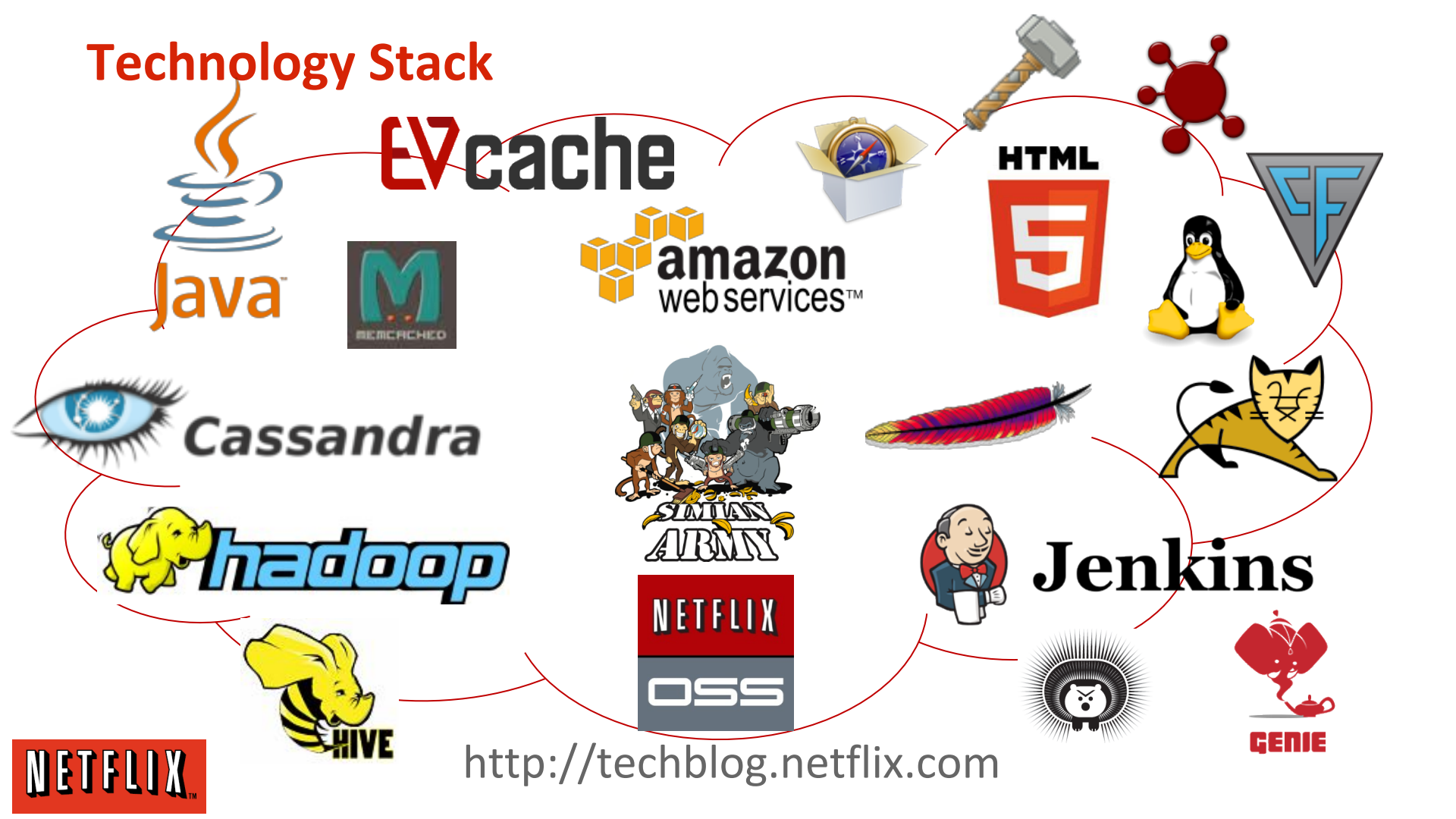


NETFLIX

Smart Architectures

NETFLIX

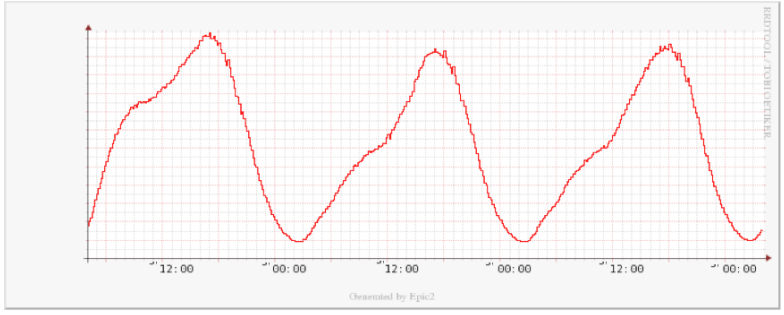
Technology Stack



http://techblog.netflix.com

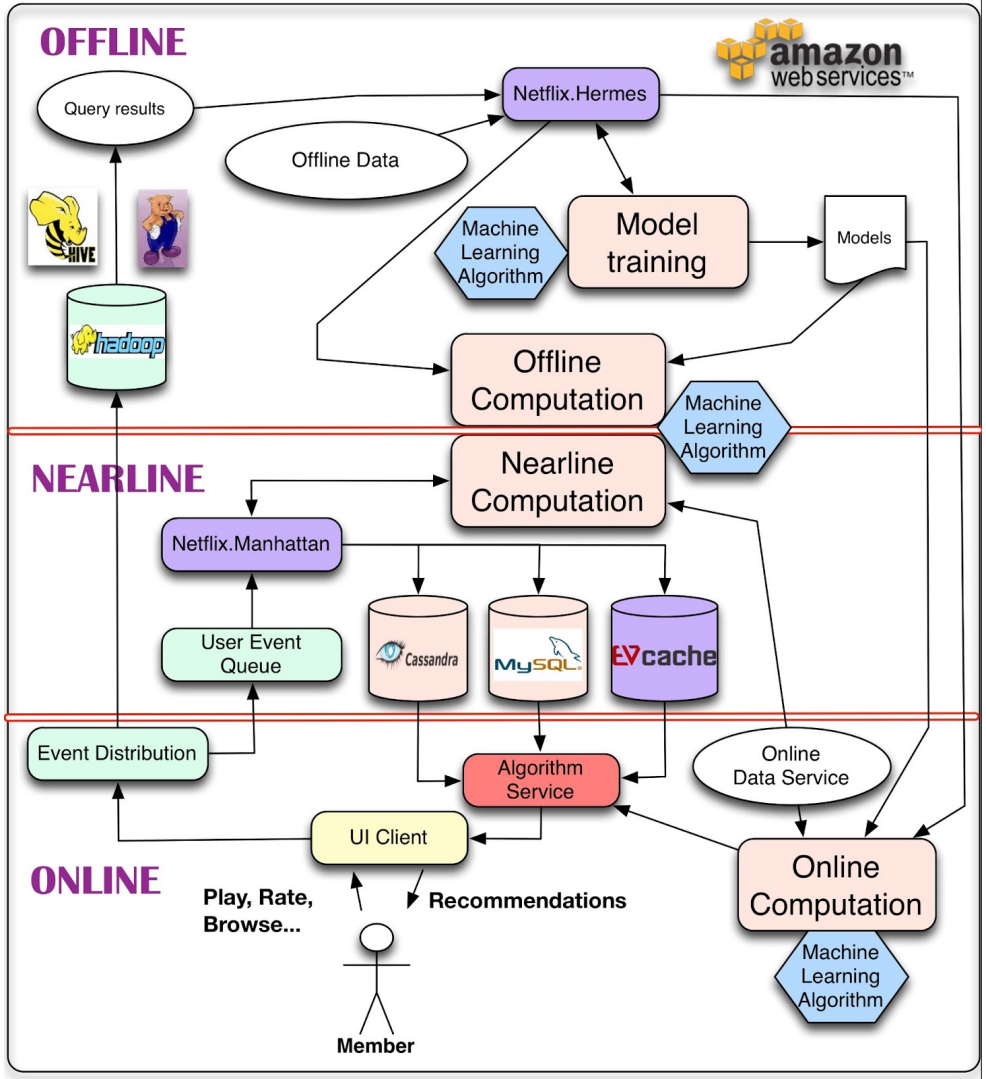
Cloud Computing at Netflix

- Layered services
  - 100s of services and applications
- Clusters: Horizontal scaling
  - 10,000s of EC2 instances
- Auto-scale with demand
- Plan for failure
  - Replication
  - Fail fast
  - State is bad
- Simian Army: Induce failures to ensure resiliency



System Overview

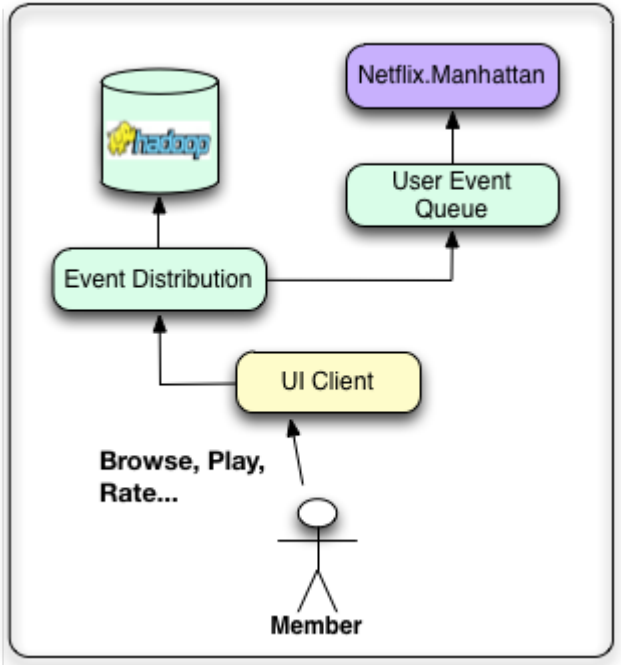
- Blueprint for multiple personalization algorithm services
  - Ranking
  - Row selection
  - Ratings
  - …
- Recommendation involving multi-layered Machine Learning

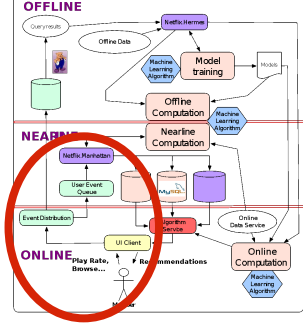


NETFLIX

Event & Data Distribution

- Collect actions
  - Plays, browsing, searches, ratings, etc.
- Events
  - Small units
  - Time sensitive
- Data
  - Dense information
  - Processed for further use
  - Saved



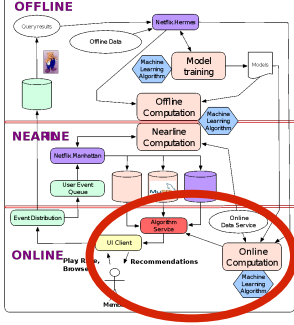


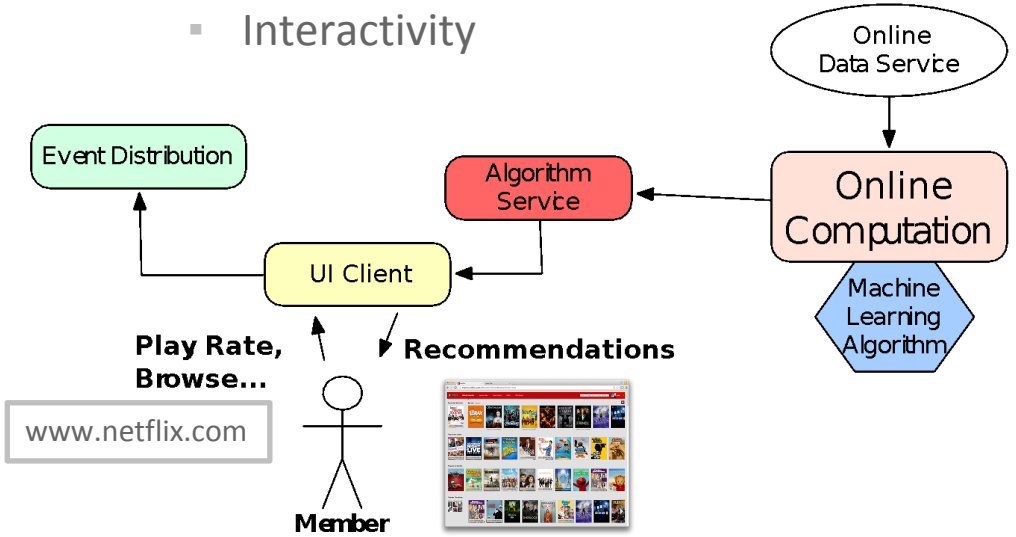
NETFLIX

Online Computation

- Synchronous computation in response to a member request
- Pros:
  - Access to most **fresh** data
  - Knowledge of full request **context**
  - Compute only what is **necessary**
- Cons:
  - **Strict** Service Level Agreements
    - Must respond **quickly** … in all cases
    - Requires high **availability**
  - **Limited** view of data

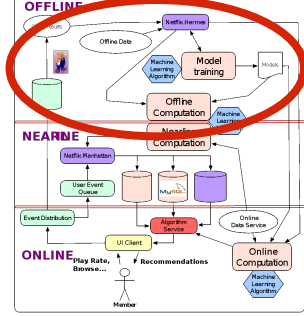
- Good for:
  - Simple algorithms
  - Model application
  - Business logic
  - Context-dependence
  - Interactivity



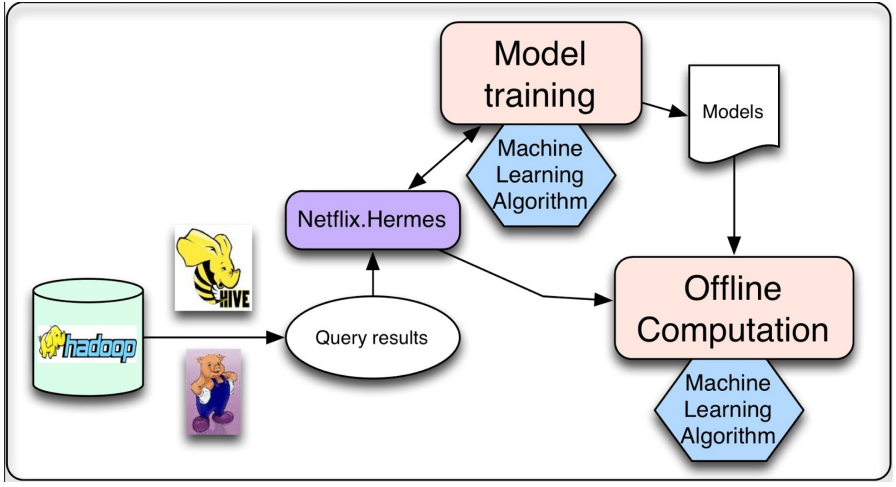


Offline Computation

- Asynchronous computation done on a regular schedule
- Pros:
  - Can handle **large data**
  - Can do **bulk** processing
  - **Relaxed** time constraints
- Cons:
  - Cannot **react** quickly
  - Results can become **stale**



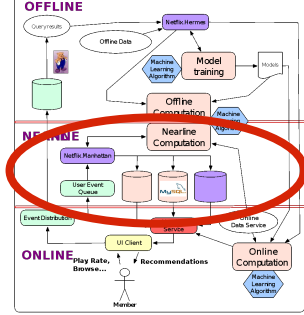
- Good for:
  - Batch learning
  - Model training
  - Complex algorithms
  - Precomputing



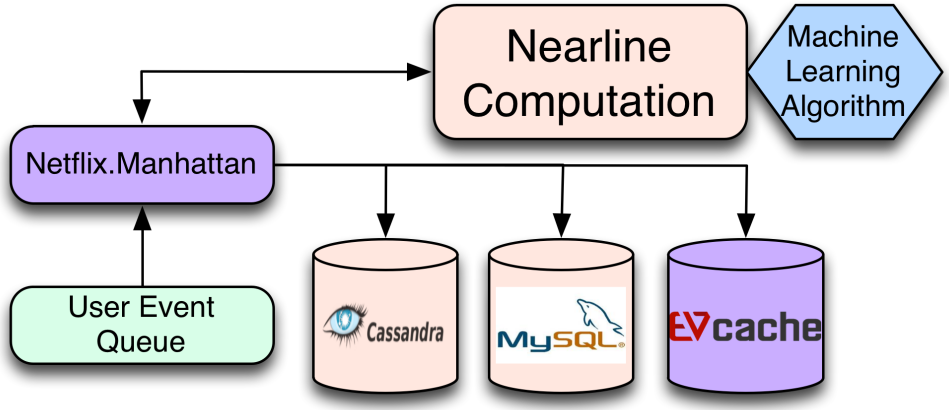
NETFLIX

Nearline Computation

- Asynchronous computation in response to a member event
- Pros:
  - Can keep data **fresh**
  - Can run **moderate** complexity algorithms
  - Can **average** computational cost across users
  - **Change** from actions
- Cons:
  - Has some **delay**
  - Done in **event context**



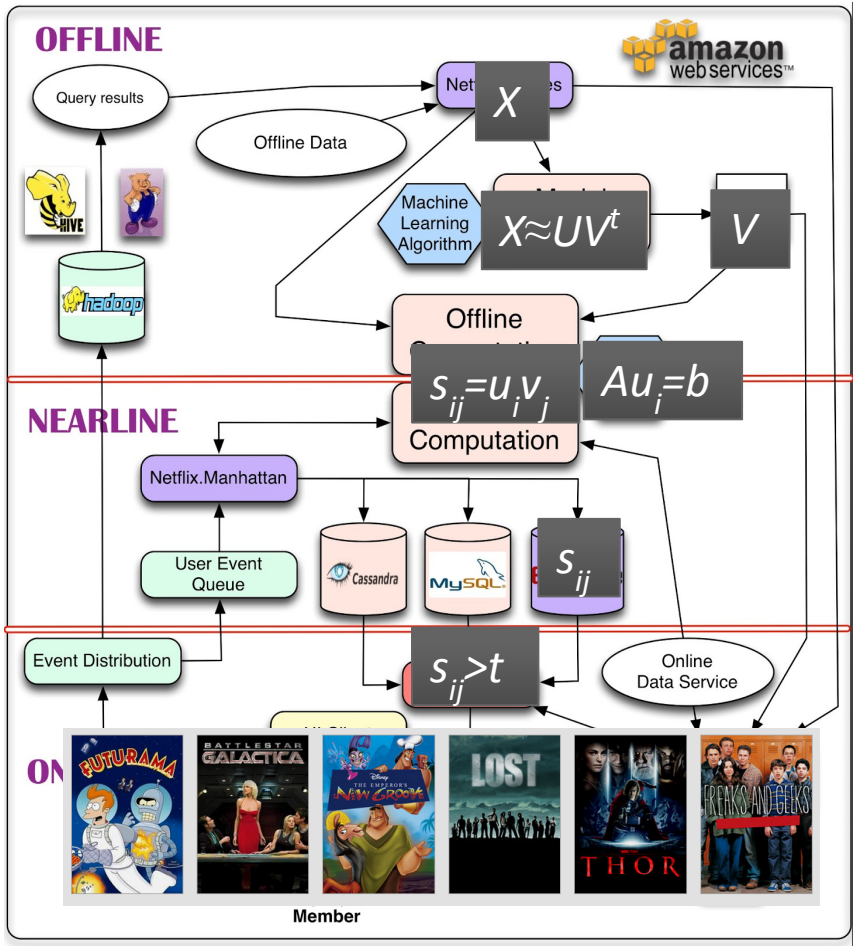
- Good for:
  - Incremental learning
  - User-oriented algorithms
  - Moderate complexity algorithms
  - Keeping precomputed results fresh



NETFLIX

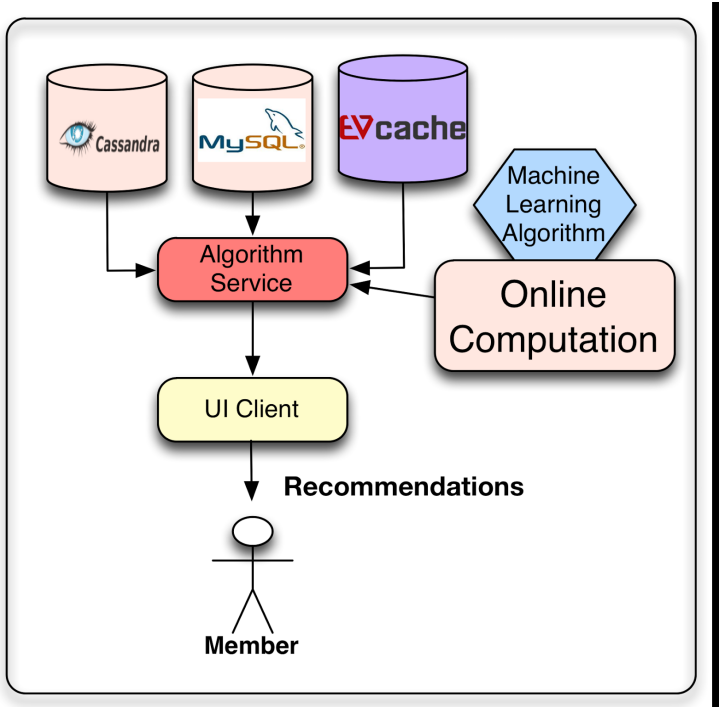
Where to place components?

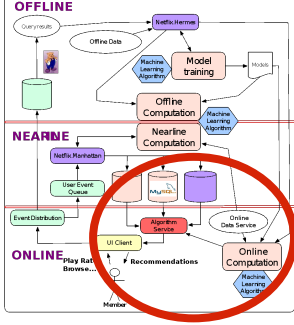
- Example: Matrix Factorization
- Offline:
  - Collect sample of play data
  - Run batch learning algorithm to produce factorization
  - Publish item factors
- Nearline:
  - Solve user factors
  - Compute user-item products
  - Combine
- Online:
  - Presentation-context filtering
  - Serve recommendations



Recommendation Results

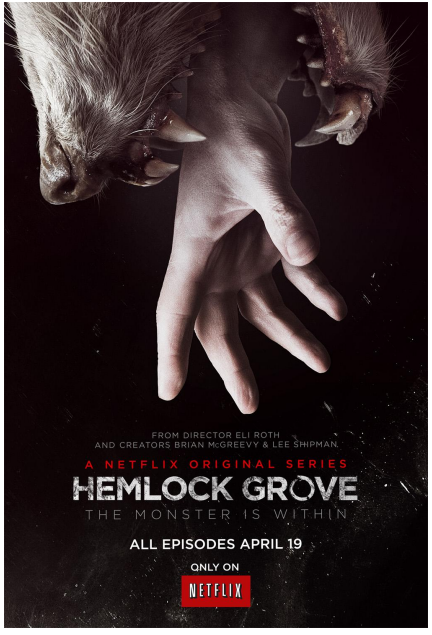
- Precomputed results
  - Fetch from data store
  - Post-process in context
- Generated on the fly
  - Collect signals, apply model
- Combination
- Dynamically choose
  - Fallbacks





NETFLIX

Conclusion



NETFLIX™

More data +

Smarter models +

More accurate metrics +

Better system architectures

Lots of room for improvement!

Xavier Amatriain (@xamat)
xavier@netflix.com

Thanks!

We’re hiring!