Recommendation Systems: A Whirlwind Tour

Kevin Li
UC Berkeley Tutorial
09/14/2017

Agenda

1. The Recommender Problem
2. The History of Recommender Systems
3. How do Traditional Recommender Systems work?
4. Beyond Traditional Methods
    - Music
        - Discovery in the Long Tail
        - Deep Learning
    - E-Commerce
        - Monetization: Margin Matters
        - Recommender Systems with Side Information
    - Advertisements/News Recommendation - Beyond Search
        - Exploration vs Exploitation
5. Conclusions

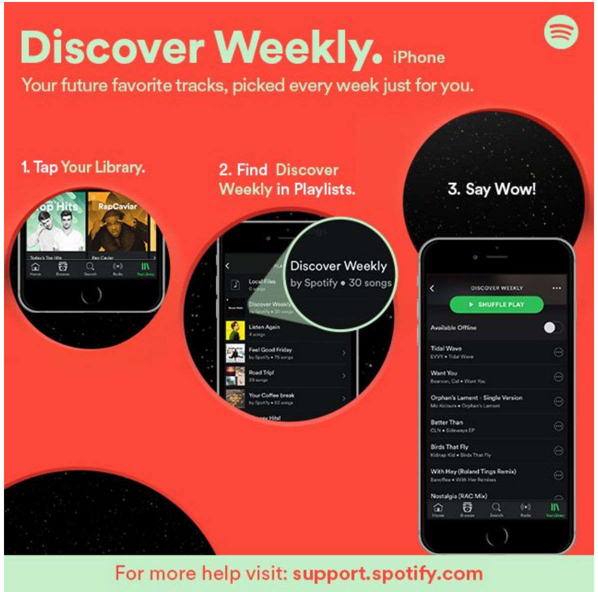
1. The Recommender Problem

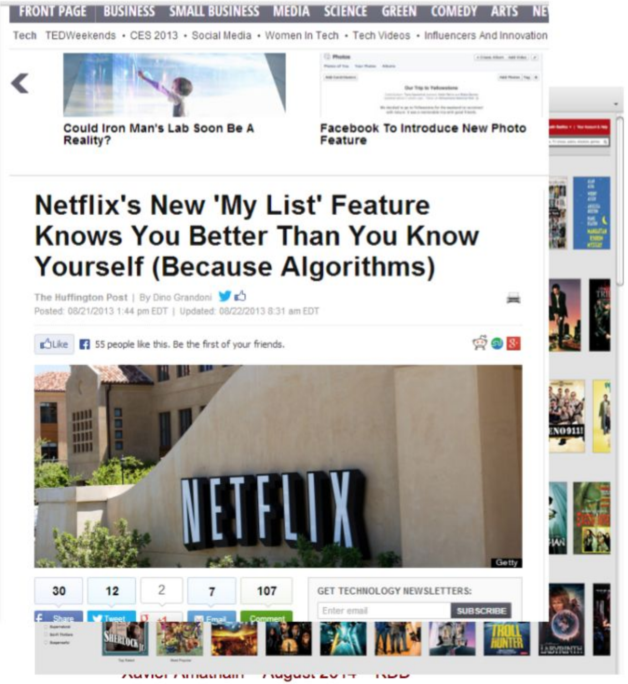
Beyond Search: The Age of Recommendations

- Xavier Amatriain
  - The Age of Search has come to an end
- Chris Anderson in “The Long Tail”
  - “We are leaving the age of information and entering the age of recommendation”
- CNN Money, “The race to create a ‘smart’ Google”:
  - “The Web, they say, is leaving the era of search and entering one of discovery. What’s the difference? Search is what you do when you’re looking for something. Discovery is when something wonderful that you didn’t know existed or didn’t know how to ask for, finds you.”

Personalizing Recommendations

- Everything is personalized





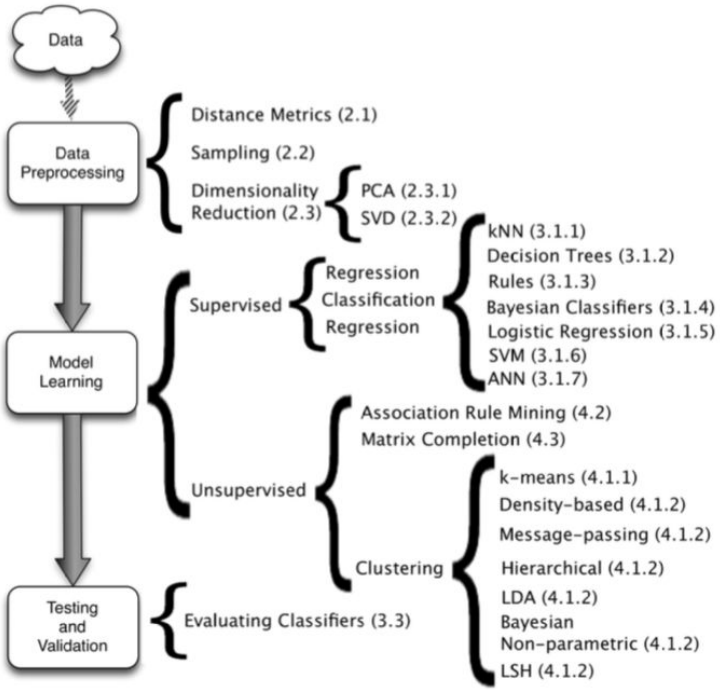
The Recommender Problem

- Traditional definition: Estimate a utility function that automatically predicts how a user will like an item.
- Based on:
  - Past behavior
  - Relations to other users
  - Item Similarity
  - Context
  - …

Recommendation in the lens of data mining

- The core of the Recommendation Engine can be assimilated to a general data mining problem

  (Amatriain et al. Data Mining Methods for Recommender Systems in Recommender Systems Handbook)

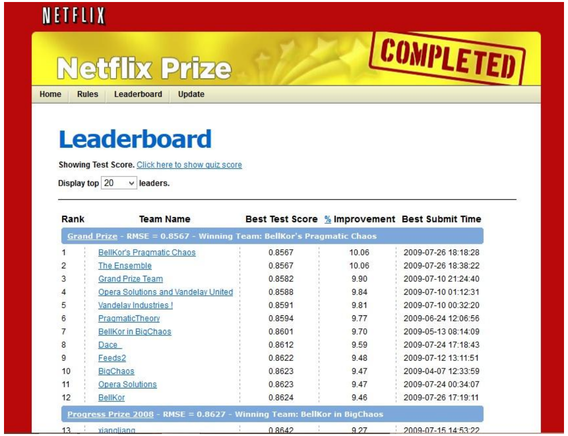


Recommendation: ML + all other things

- User Interface
- System Requirements (efficiency, scalability, privacy)
- Serendipity
- Diversity
- Awareness
- Explanations
- …

2. The History of Recommender Systems

The Netflix Challenge


NETFLIX

Netflix Prize COMPLETED

Home | Rules | Leaderboard | Update

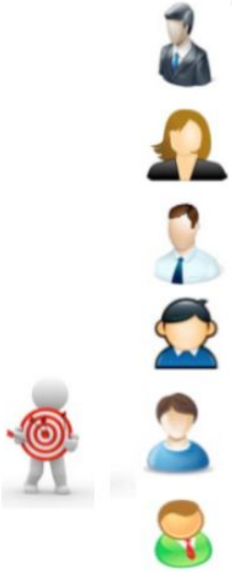
Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22

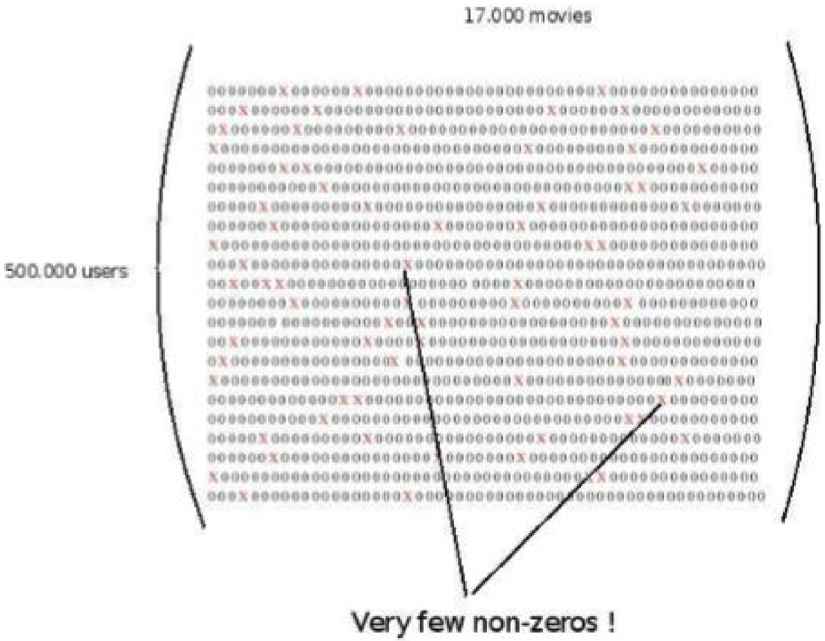
The Netflix Challenge (Cont.)



2		2	4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

The Sparsity Problem

- Typically:
  - large product sets, user ratings for a small percentage of them
    - Amazon: millions of books and a user may have bought hundreds of books)
    - Netflix: 8.5 Billion positions in the User/Movie matrix and 100 Million filled positions
    - Spotify: 20 Million Songs
  - number of users needs to be ~0.1x size of the catalog
  - each user on average must have r interactions if we want to learn r factors



3. How do Recommender Systems work?

“Traditional” Methods

- Collaborative Filtering
- Content-Based Recommendations
- Hybrid Approaches

Collaborative Filtering: Ingredients

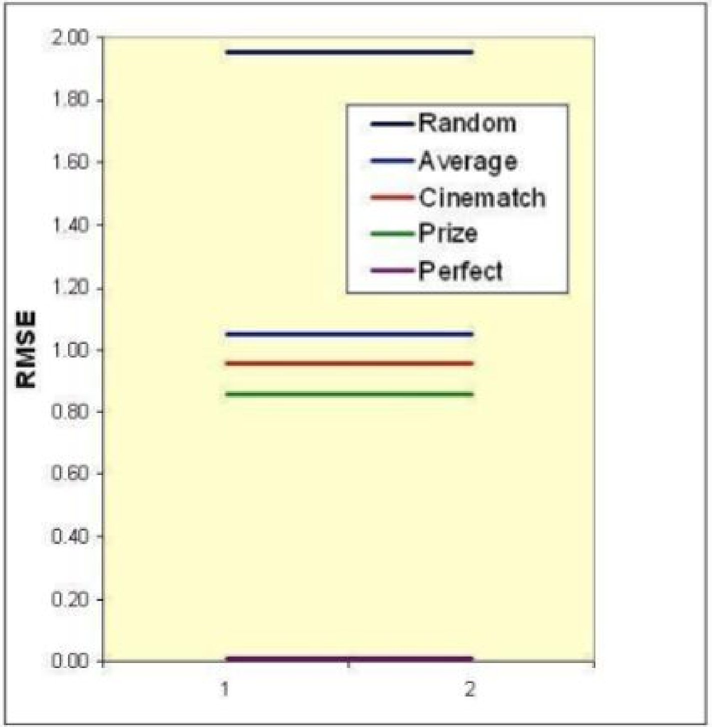
- List of m Users
- List of n Items
- Each user has a list of items with associated opinions
  - Explicit: a rating score
  - Implicit: purchase records or music listening records
- Active users for whom the CF prediction task is performed
- Metric for measuring similarity between users
- Method for selecting a subset of neighbors
- Method for predicting a rating for items not currently rated by the active user

Collaborative Filtering

1. **Memory-Based Collaborative Filtering**
    a. User-Based
    b. Item-Based
    c. Memorization works well with good data
    d. Generalization is challenging

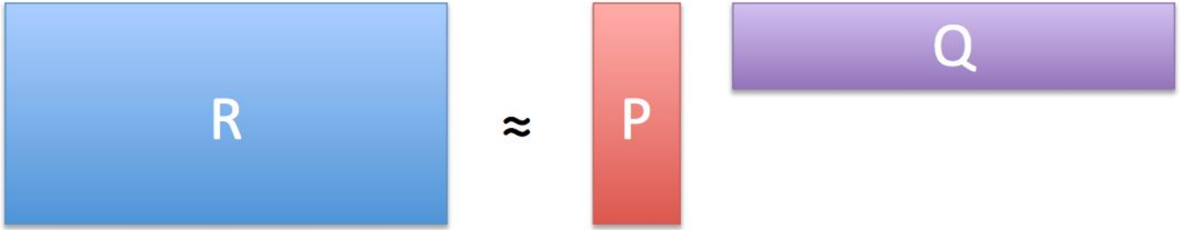
2. **Model-Based Collaborative Filtering**
    a. Model-Based
    b. Requires careful modeling
    c. Typically involves large computational efforts
    d. Generalization + Memorization

Personalized vs. Not Personalized



The Three Pillars: Matrix Factorization

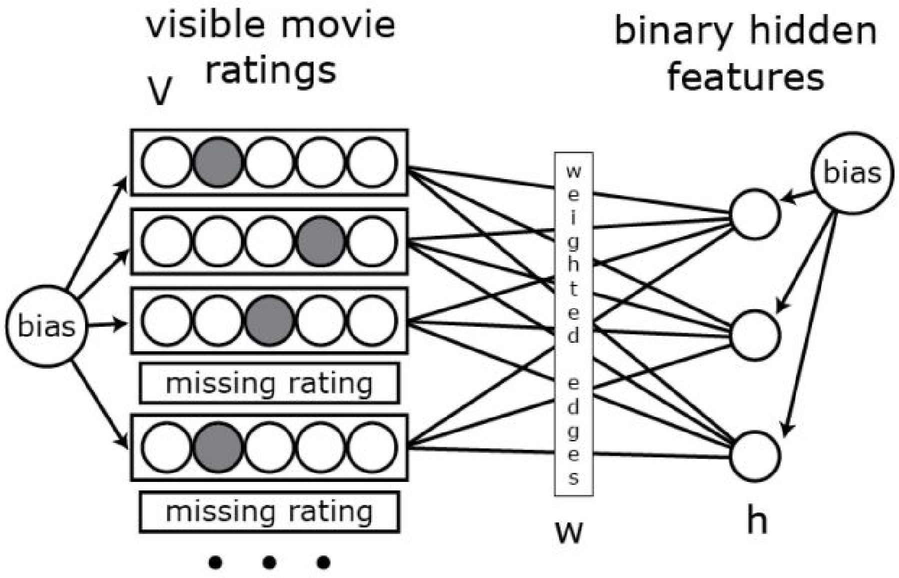
- Matrix Factorization



- Alternating Least Squares
- Online/Stochastic Gradient Descent
  - Dec. 2, 2006: Simon Funk (Brandyn Webb)
- Typical Improvement: 4%

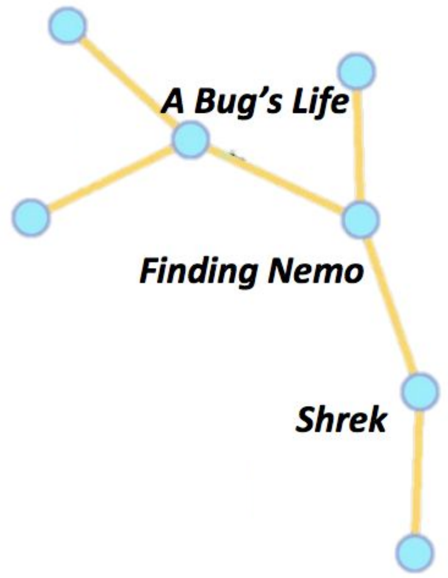
The Three Pillars: Restricted Boltzmann Machines

- **Deep Learning Strikes Back!**
- May 2007, Salakhutdinov and Mnih
- Typical Improvement: 5%



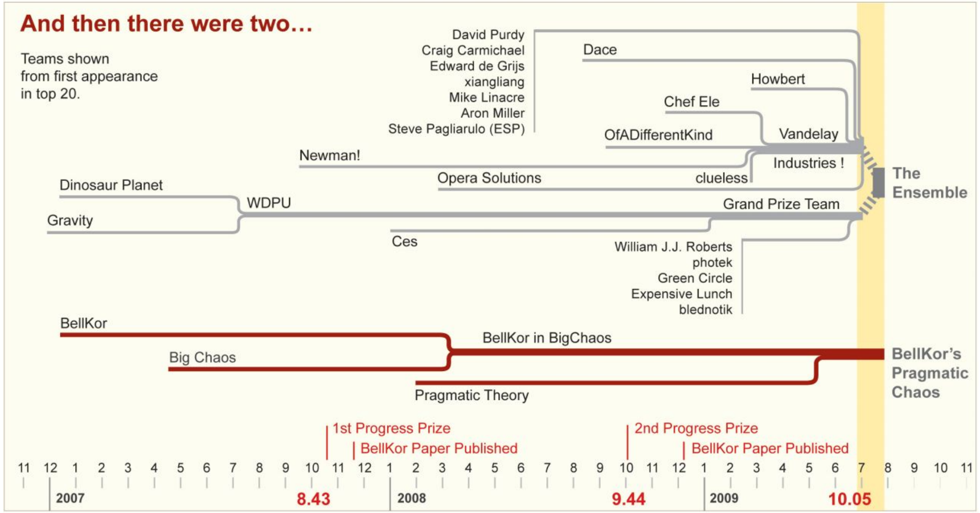
The Three Pillars: Nearest Neighbor Methods

- Nearest Neighbor Methods:
  - Classical KNN: 0.5% improvement
  - KNN with learned weights:
    - Aug. 2007, Bell, Koren, Volinsky
- Typical Improvement: 4.6%
- Often trained on errors of MF or RBM



Model Ensembling, Variations, and Side Information

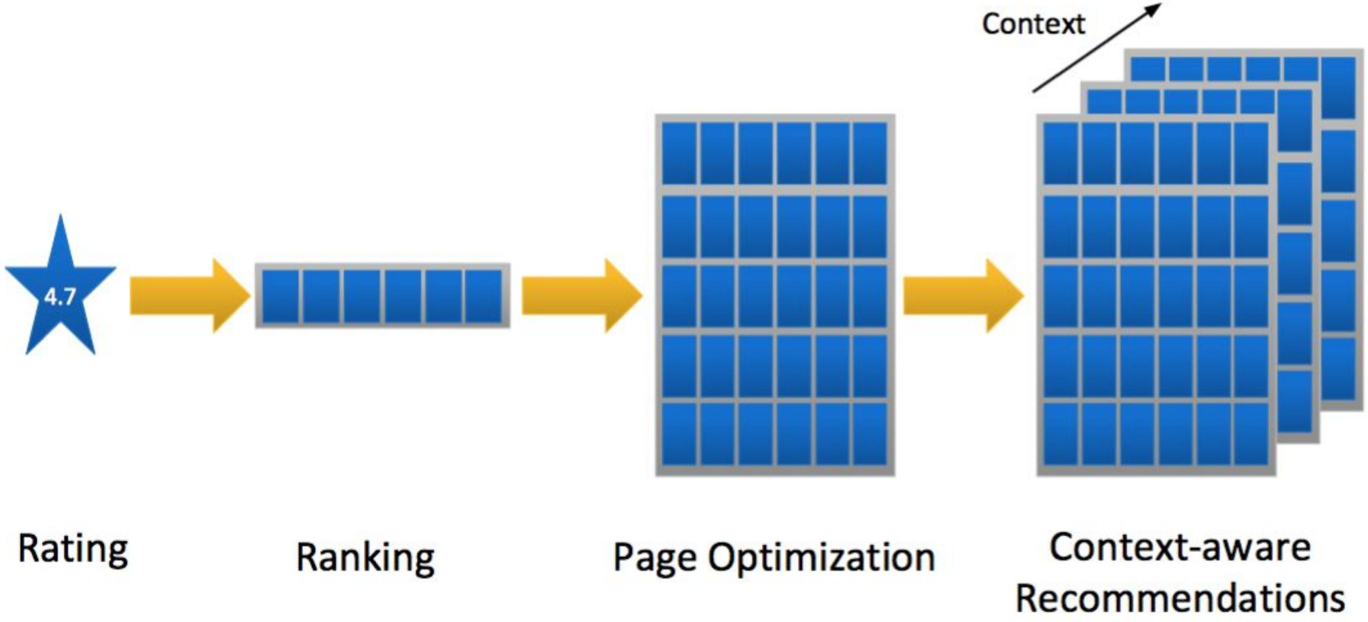
- In the final 2 years, model ensembling was the key
  - Training on Errors
  - Clustering on Errors
  - Stacked Linear Regressions
  - Model Variations/Side Information:
    - Add user, item, data features as covariates
    - Date of Rating
    - Gaussian Missing Data Model
  - Team Mergers!



What Netflix learned

- Netflix's current production model includes the top 2 algorithms in the 2007 Progress Prize
  - SVD ++ (RMSE: 0.8914)
  - RBMs (RMSE: 0.8990)
- Linear Blend of the 2 top algorithms
  - RMSE: 0.88
- Currently used as part of Netflix's rating prediction component
- Limitations:
  - Designed for 100M ratings, Netflix has 5B ratings
  - Not adaptable as users add ratings
  - Performance Issues

Evolution of the Recommender Problem



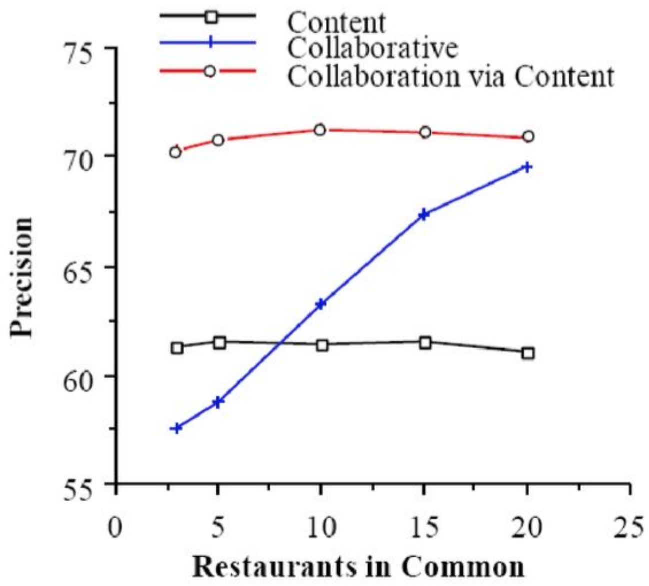
Content-Based Recommendations

- Recommendations based on content of items rather than on other users' opinions/interactions
- Goal: recommend items similar to those the user liked
- Common for recommending text-based products
- Items to recommend are "described" by their associated features (e.g. keywords)
- Pros:
  - No need for data on other users - No cold-start or sparsity
  - Handles users with unique tastes
- Cons:
  - Requires content that is meaningful
  - Difficult to implement serendipity
  - Easy to overfit

Hybrid Approaches

Methods:

- Weighted
- Switching
- Mixed
- Feature Combination
- Cascade
- Feature Augmentation
- Meta-Level



Averaged on 44 users

Precision computed in top 3 recommendations

Beyond “Traditional” Recommendations (Netflix)

Music Recommendation

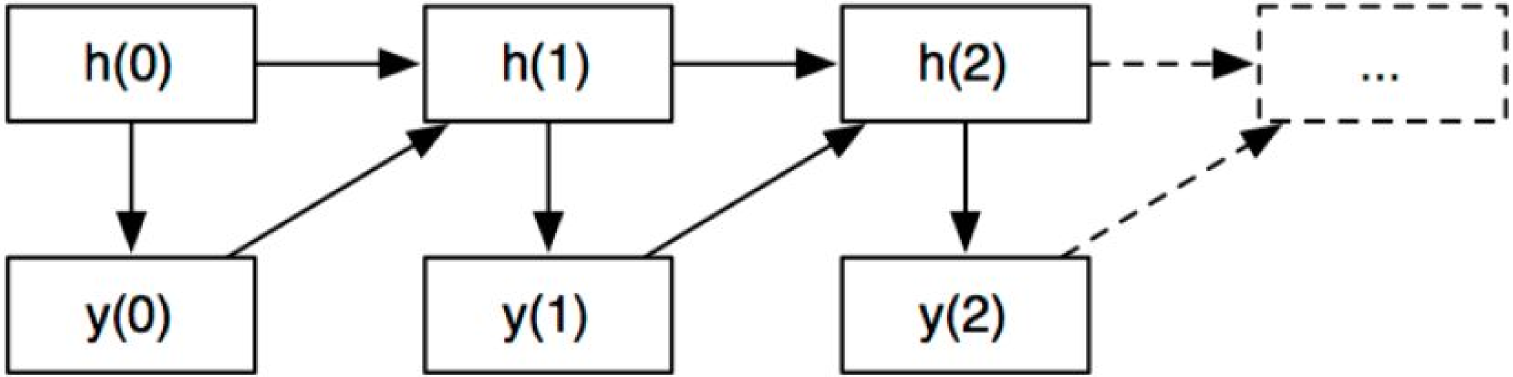
- Music consumption is biased towards a few popular artists
- In 2007
  - 1% of all digital tracks accounted for 80% of all sales
  - 1k albums accounted for 50% of all album sales
  - 80% of all albums sold were purchased less than 100 times
- There is a need to assist people to filter, discover, personalise, and recommend from the huge amount of music content available along the “Long Tail of Discovery” (Oscar Celma, Director of Research, Pandora)

Music Recommendation

- Quite often algorithms tend to simply recommend popular songs
  - Decreasing the effectiveness of recommendations
- Simply focusing on accuracy is an insufficient approach
- There is value in recognizing how the user perceives the recommendations
  - Promote both novel and relevant material
- Advancements in the past 2 decades
  - Network-Based Approach for recommender systems (based on user/item similarity graph)
  - User-Centric Evaluation that measures the user's relevance and novelty of the recommendations
  - Deep Learning Based Approaches

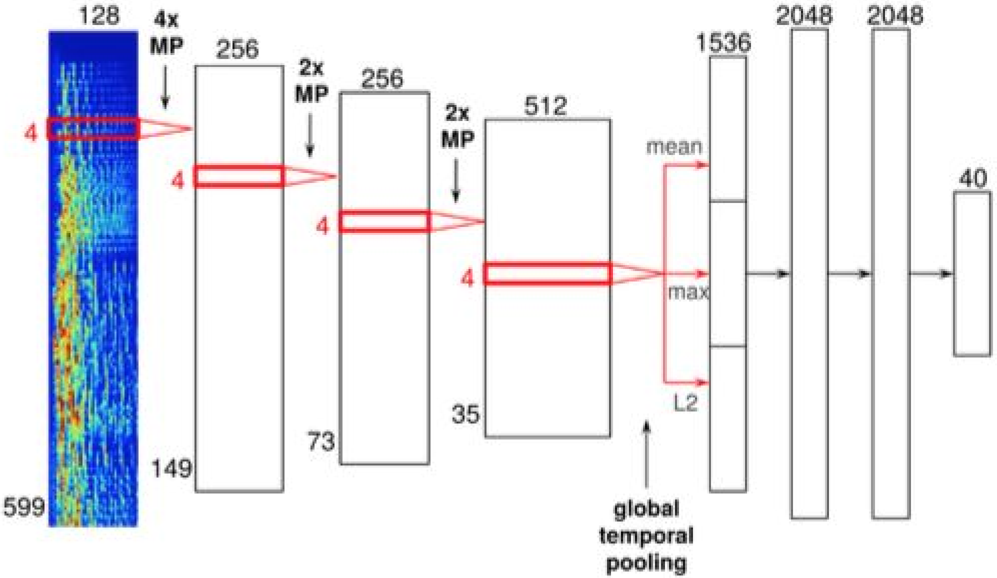
Deep Learning for Collaborative Filtering

- Spotify uses Recurrent Neural Networks for Playlist Prediction

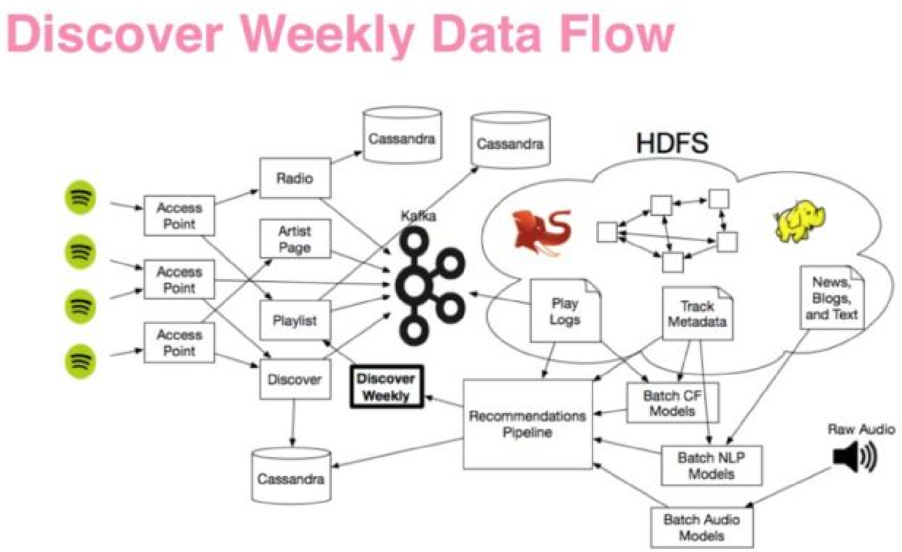


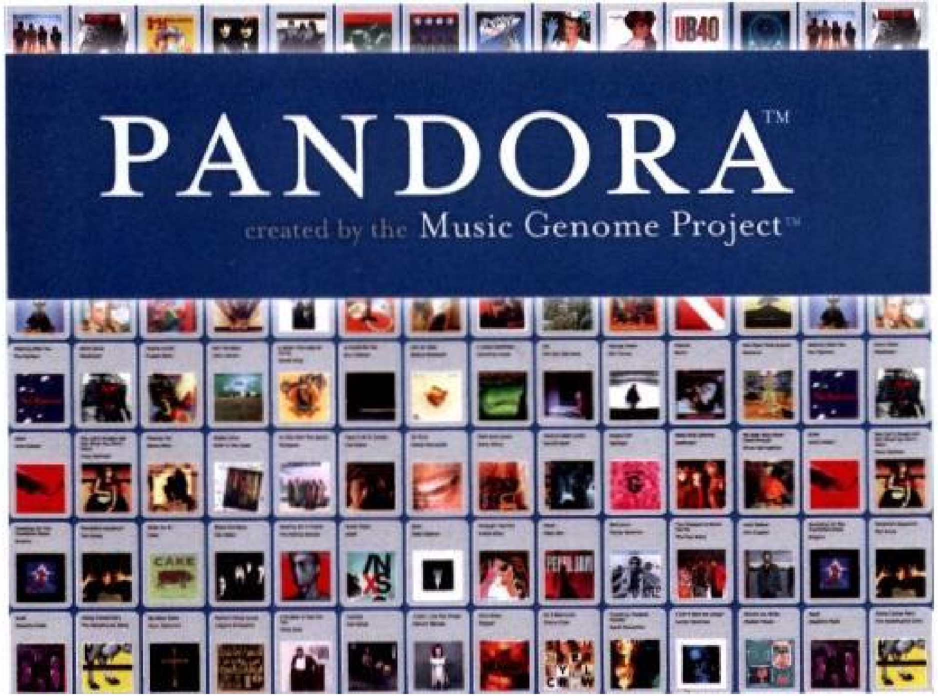
Deep Learning for Content-Based Recommendations

- Spotify also used deep learning to understand music composition to create content/features for its recommender engine



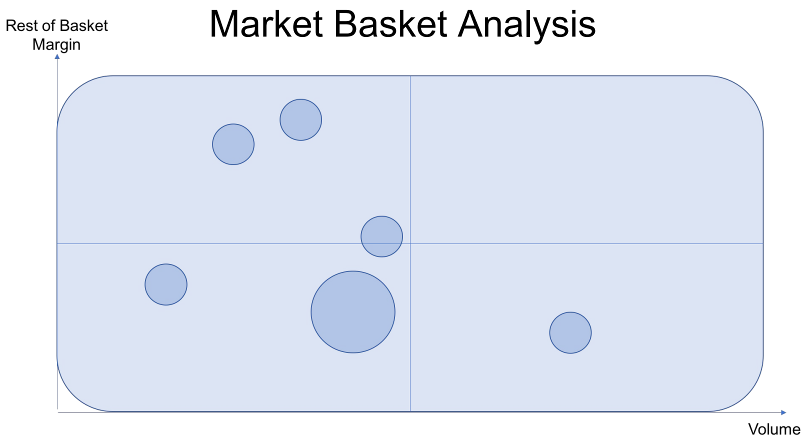
Music: Spotify and Pandora





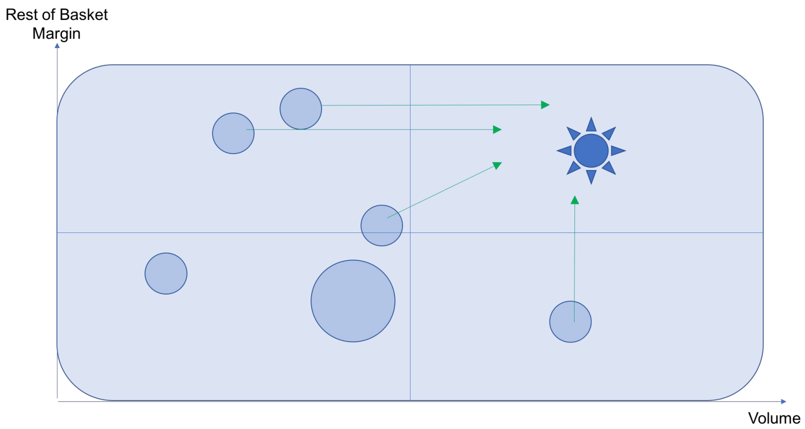
E-Commerce: Amazon/Alibaba/JD.com

- Amazon: the original inventor of item-to-item collaborative filtering
- Common Algorithms:
  - Association Rules
  - Item-based CF
- Miscellaneous Constraints
  - Inventory Constraints
  - Pricing effects/Demand Response
  - Bundling
  - Margin Calculation
  - Assortment Optimization

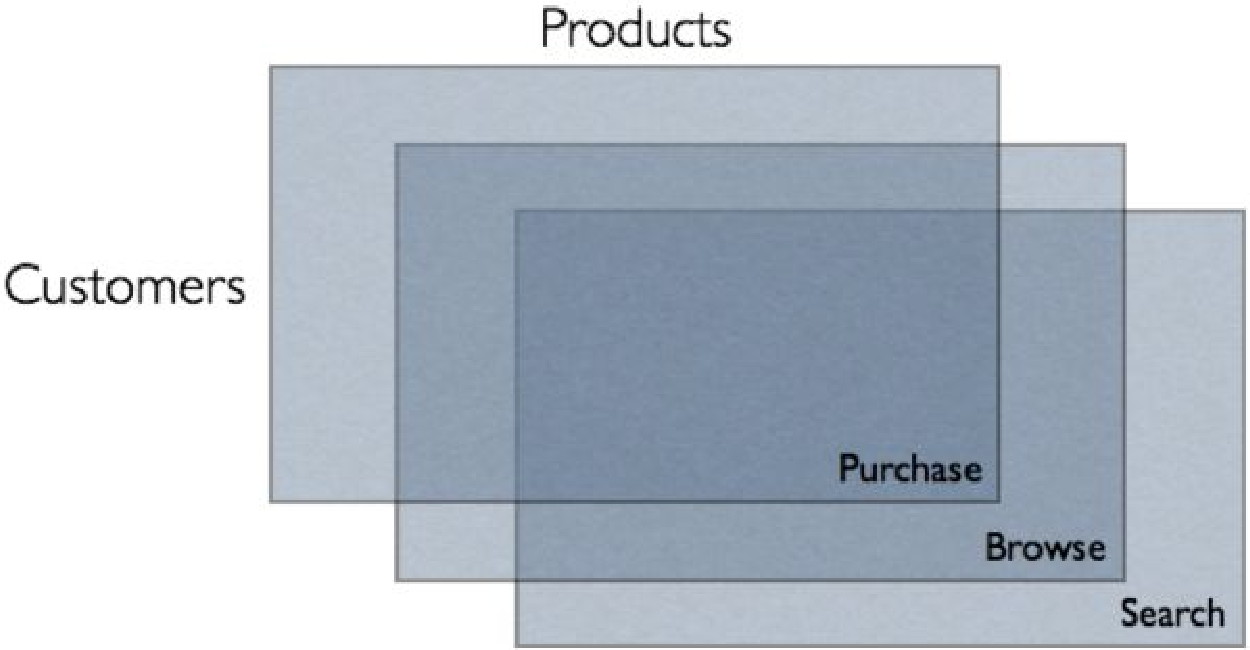


E-Commerce: Amazon/Alibaba/JD.com

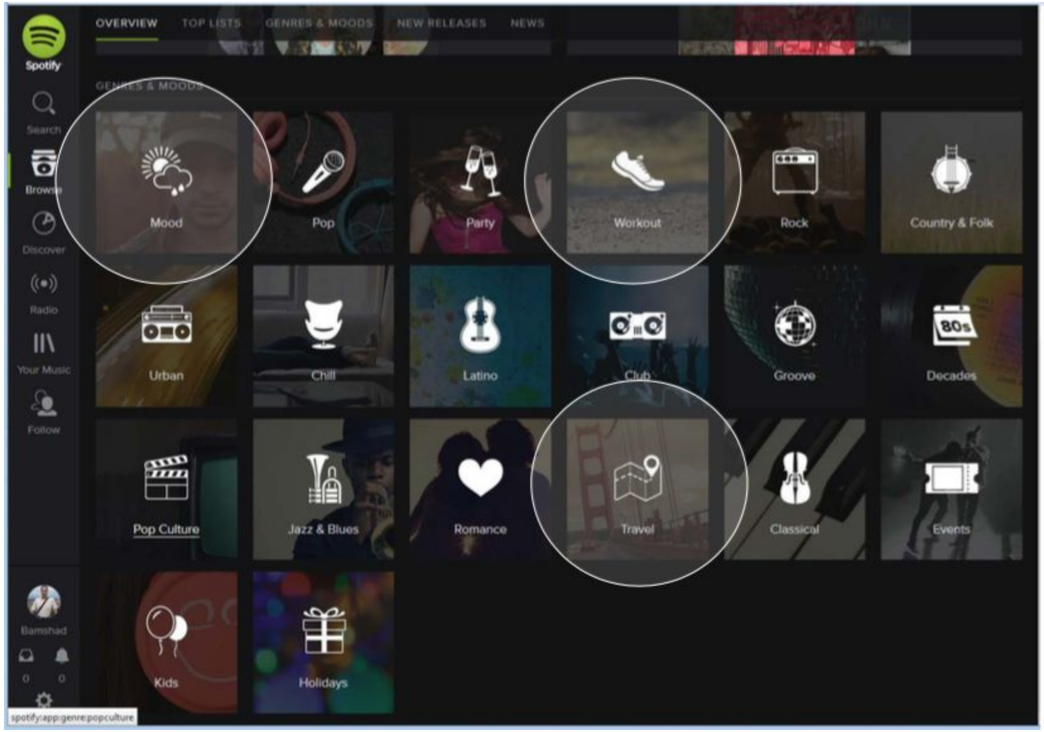
- Amazon: the original inventor of item-to-item collaborative filtering
- Common Algorithms:
  - Association Rules
  - Item-based CF
- Miscellaneous Constraints
  - Inventory Constraints
  - Pricing effects/Demand Response
  - Bundling
  - Margin Calculation



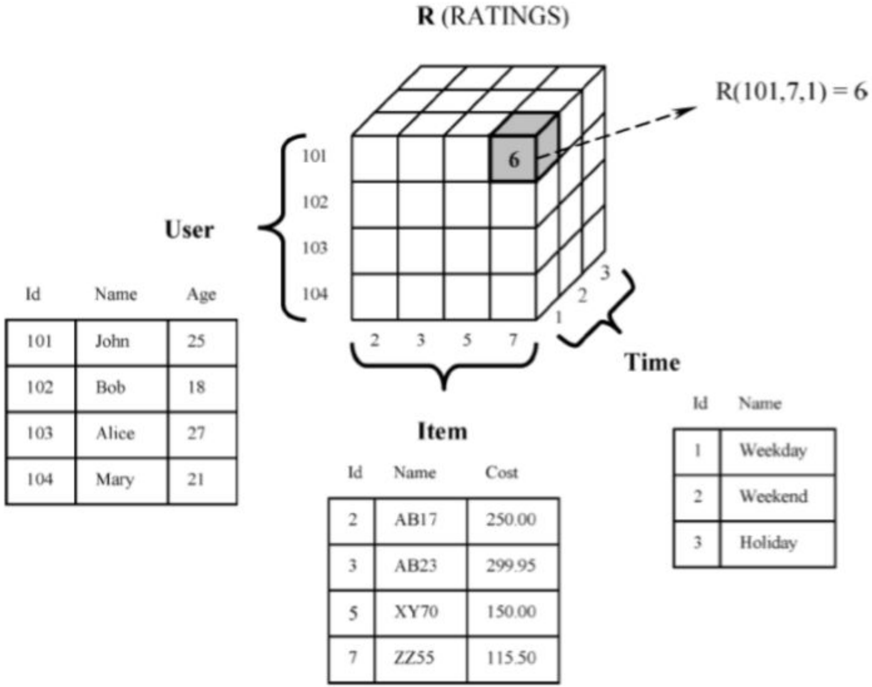
Recommender Systems with Side Information

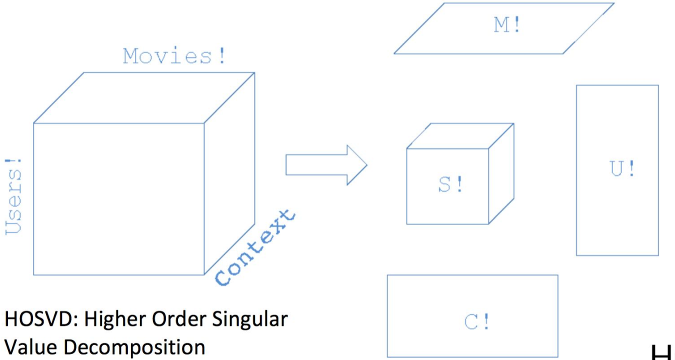


Context in Recommendation



CF for a N-dimensional Model





HOSVD: Higher Order Singular Value Decomposition

$U \in \mathbb{R}^{n \times d_U}$. $M \in \mathbb{R}^{m \times d_M}$ and $C \in \mathbb{R}^{c \times d_C}$

$S \in \mathbb{R}^{d_U \times d_M \times d_C}$

HOSVD Model

Advertisements/News Recommendation

- Consider the following scenario:
  - Loop:
    1. User visits website with profile, browsing history, ….
    2. Website operator chooses content/ads/news articles to display
    3. User reacts to content/ads (e.g. click, “like”)
- Goal: choose contents/ads that yield desired user behavior

A Concrete Example

- Consider our team’s newest member:
  Albert Lee

A Concrete Example

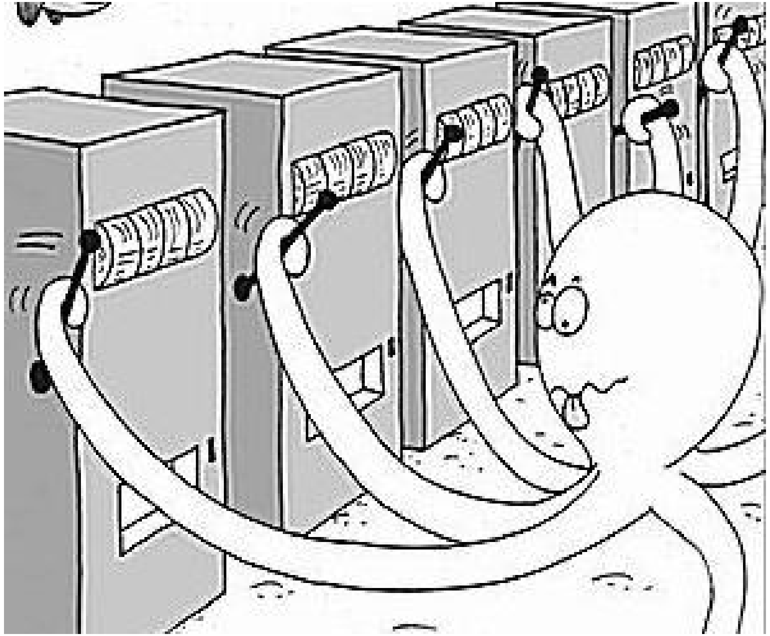
- Consider our team's newest member: Albert Lee
- What we already know Albert loves:
    - Chelsea
    - 3-Way Coffee

A Concrete Example

- Consider our team's newest member: Albert Lee
- What we already know Albert loves:
  - Chelsea
  - Coffee
- What else does Albert like?
  - Fast Cars?
  - Marvel Movies?
  - Wine?

A Concrete Example

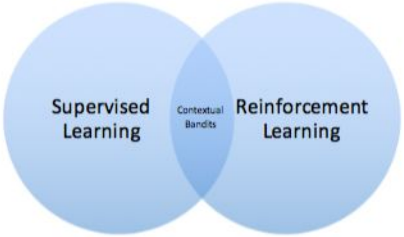
- Consider our team's newest member: Albert Lee
- What we already know Albert loves:
  - Chelsea
  - 3-Way Coffee
- What else does Albert like?
  - Fast Cars?
  - Marvel Movies?
  - Wine?
- Only One Way to Find Out - Run Experiments



The Exploration-And-Exploitation Tradeoff

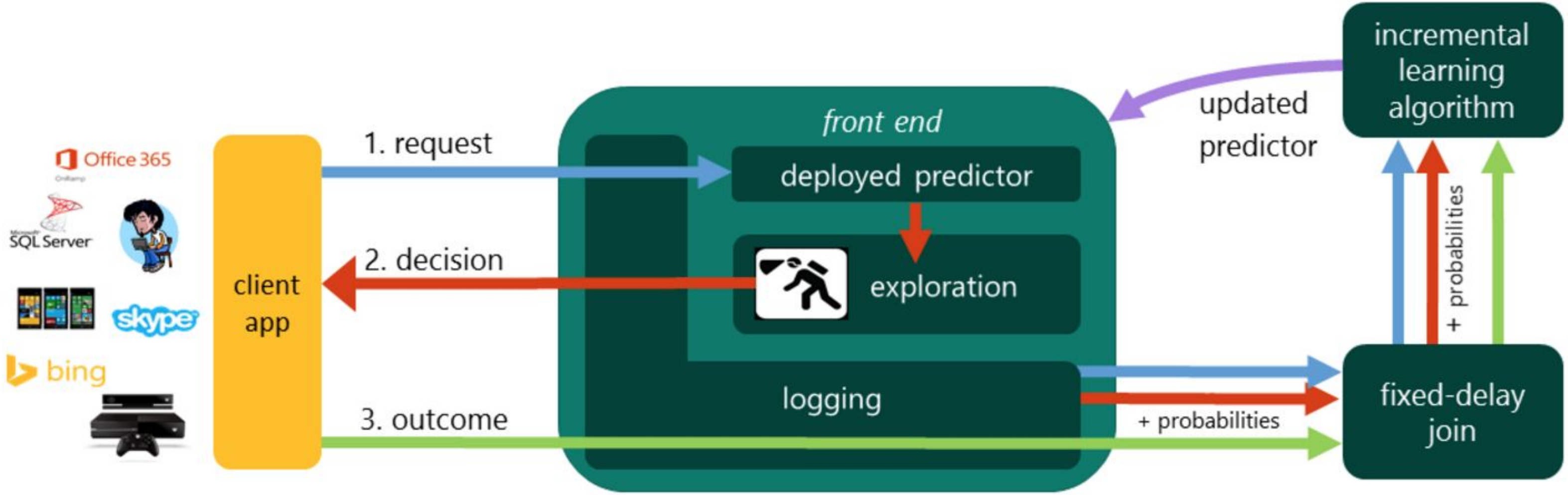
- Exploitation
  - Benefit immediately from current knowledge
- Exploration
  - There exists a need to learn
  - Experiments are expensive
  - Most likely, unknown unknowns are even pricier

- Multi-arm Bandits
  - Stochastic Bandits
  - Adversarial Bandits
  - Contextual Bandits
- Only the reward associated with the selected action is revealed.



Microsoft/LinkedIn

- In 2016, Microsoft released its implementation of contextual bandits and branded it as Multiworld Testing:



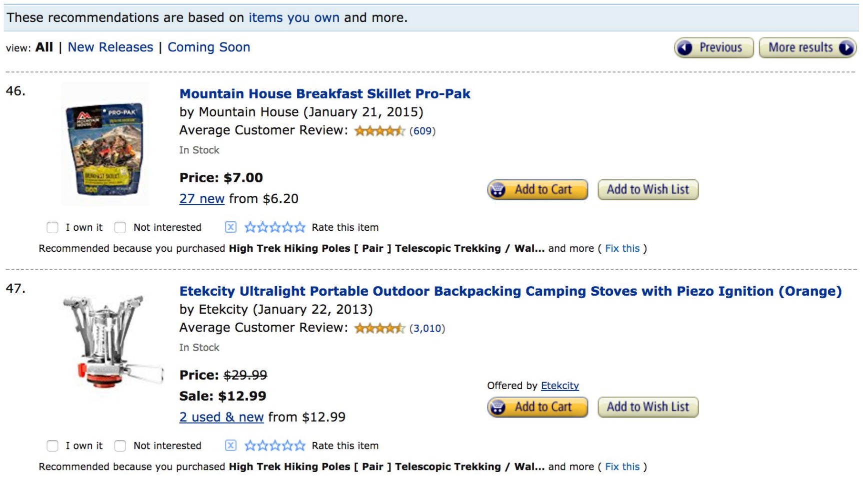
Conclusions & Takeaways

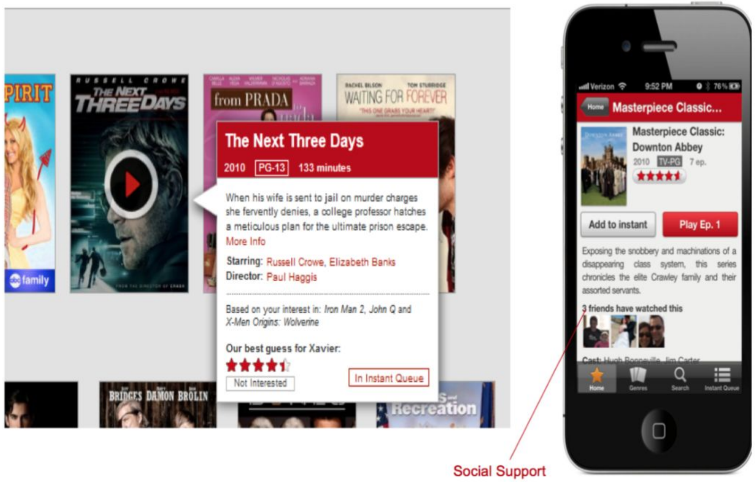
- Recommendation is just the beginning
  - Ranking, Explainability, Serendipity, Page Optimization, Exploration & Exploitation
- What works?
  - Depends on the domain and the particular problem
- What matters?
  - Data Preprocessing
  - Smart Dimensionality Reduction
  - Combining methods/ensembles
- For many companies, it means:
  - Ranking by rating may work (depending on product catalog size)
  - Ensemble various sources of information and similarity measures
  - Personalized Recommendations is the future

References

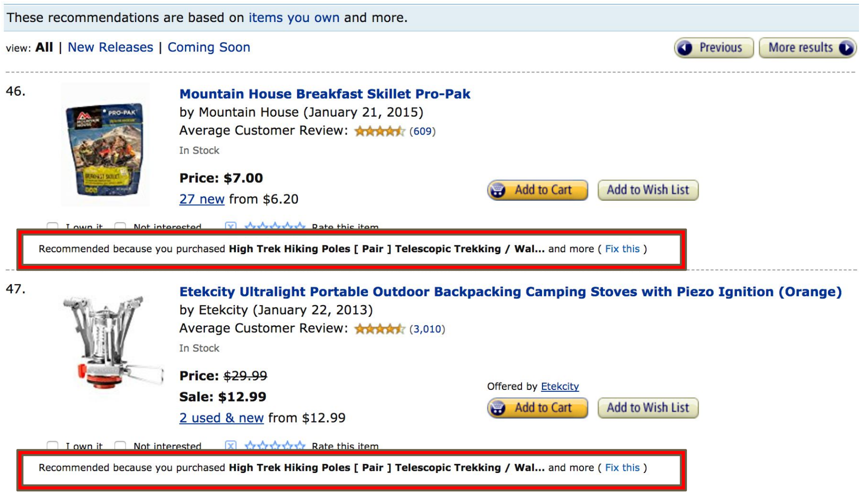
- The Recommender Problem Revisited. Xavier Amatriain. KDD 2014.
- Context-Aware Recommendations. Bamshad Mobasher. KDD 2014.
- Learning Preferences with Side Information. Viviek F. Farias and Andrew A. Li. MIT Working Paper
- Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. ICML 2014.
- The Story of the Netflix Prize: An Ensembler's Tale. Lester Mackey. National Academies' Seminar, Washington, DC, Nov. 2011.

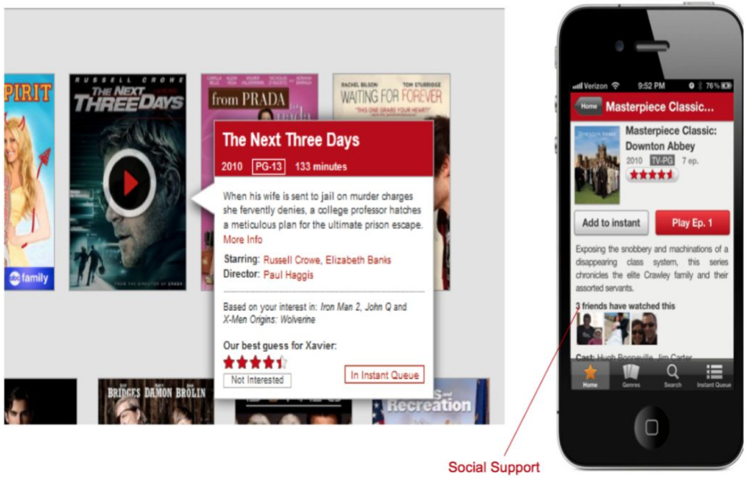
Explanation/Support for Recommendations





Explanation/Support for Recommendations





Diversity & Awareness