

# Approaches to Recommendation in Industry

Xavier Amatriain

[xavier@amatriain.net](mailto:xavier@amatriain.net)

@xamat

Quora/Xavier-Amatriain

Recsys 2017 Summer School

Bolzano, IT

# Outline

---

1. The Traditional Recommender Problem
2. The Netflix Prize
3. Beyond Rating Prediction
4. Lessons Learned
5. A Recsys Architectural Blueprint
6. Building a state-of-the-art recommender system in practice
7. Hands-on tutorial
8. Future research Directions
9. Conclusions
10. Some references

# 1. The Recommender Problem

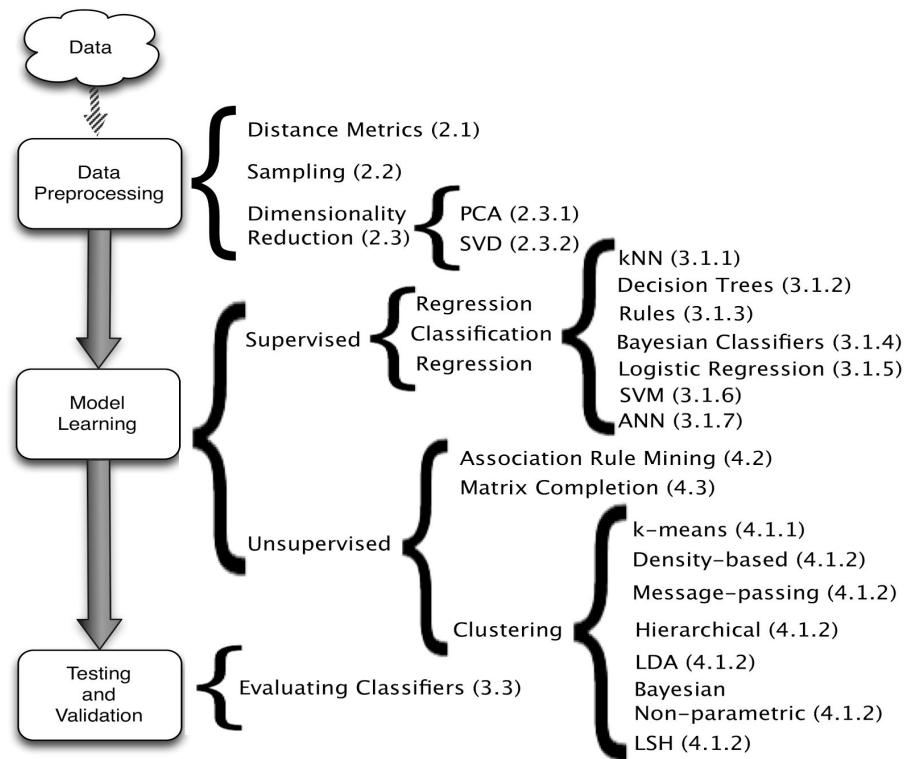
# The “Recommender problem”

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

# Recommendation as 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*)



# Data Mining + all those other things

---

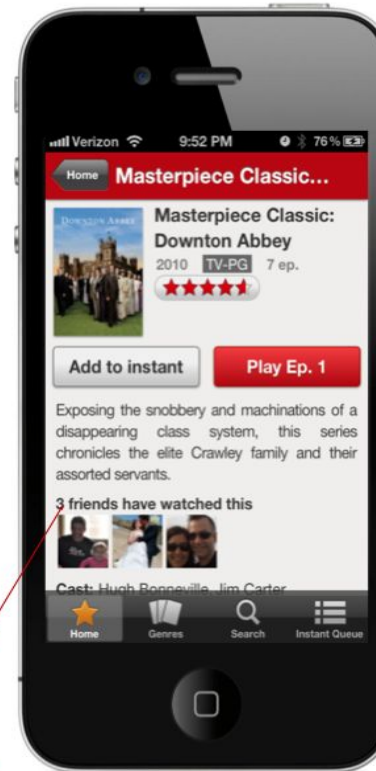
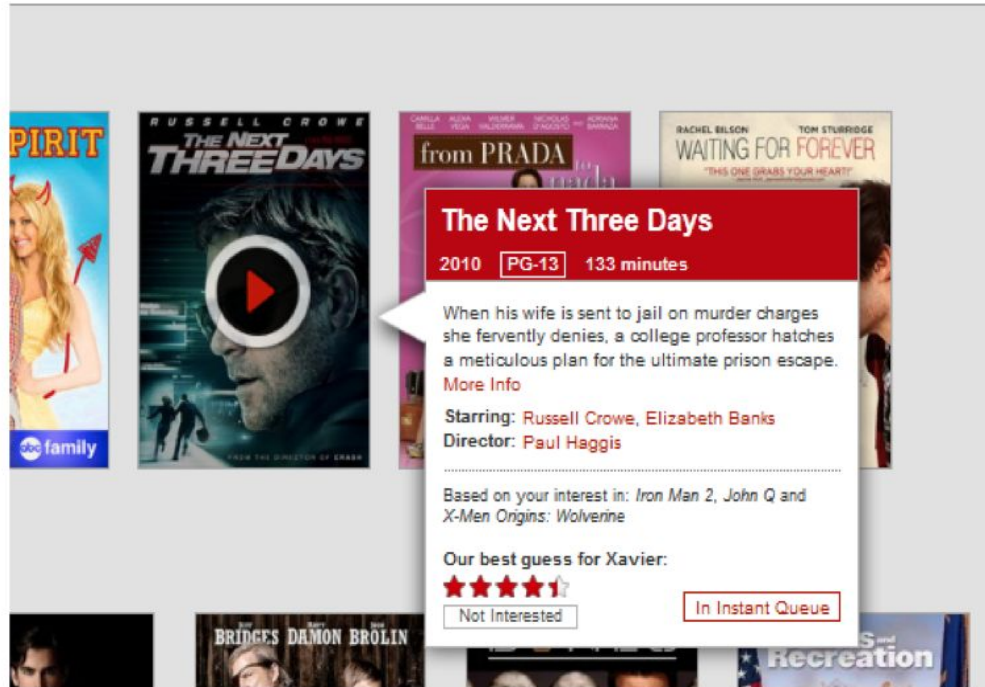
- User Interface
- System requirements (efficiency, scalability, privacy....)
- Serendipity
- Diversity
- Awareness
- Explanations
- ...

# Serendipity

---

- Unsought finding
- **Don't recommend** items the user already knows or **would have found anyway**.
- Expand the user's taste into neighboring areas by improving the obvious
- Serendipity ~ Explore/exploit tradeoff

# Explanation/Support for Recommendations

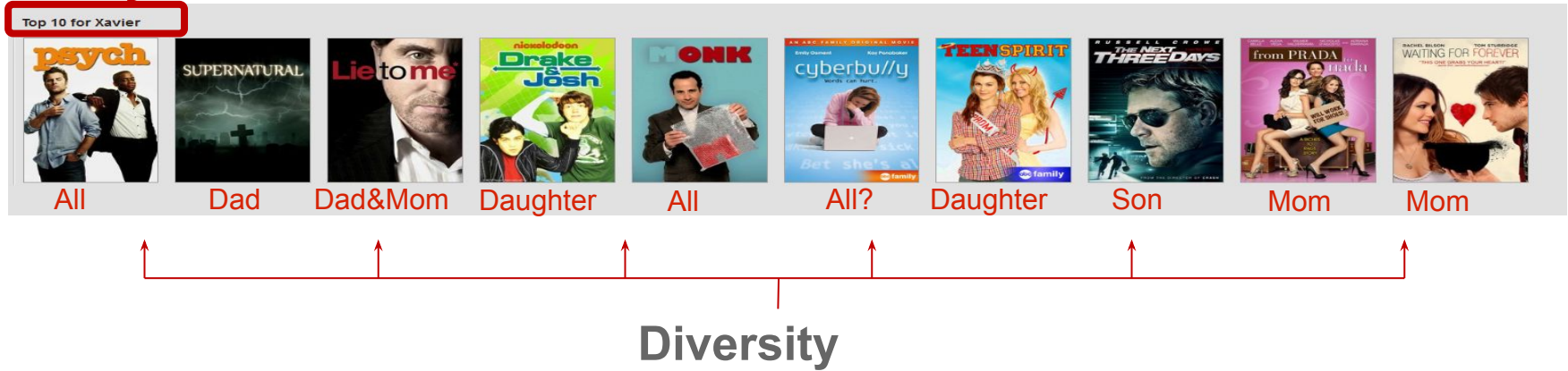


Social Support



# Diversity & Awareness

Personalization awareness



# What works

- Depends on the **domain** and particular **problem**
- However, in the general case it has been demonstrated that the best isolated approach is CF.
  - Other approaches can be hybridized to improve results in specific cases (cold-start problem...)
- What matters:
  - **Data preprocessing**: outlier removal, denoising, removal of global effects (e.g. individual user's average)
  - “Smart” **dimensionality reduction** using MF
  - **Combining methods** through ensembles

## 2. The Netflix Prize

# Netflix Prize

COMPLETED

What we were interested in:

- High quality *recommendations*

Proxy question:

- Accuracy in predicted rating

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



# 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, not XB ratings
  - Not adaptable as users add ratings
  - Performance issues

## What about the final prize ensembles?

---


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

# 3. Beyond Rating Prediction


# Everything is a recommendation

FRONT PAGEBUSINESSSMALL BUSINESSMEDIA SCIENCE GREEN COMEDY ARTS NE

Tech TEDWeekends • CES 2013 • Social Media • Women In Tech • Tech Videos • Influencers And Innovation





Could Iron Man's Lab Soon Be A Reality?





Facebook To Introduce New Photo Feature

## Netflix's New 'My List' Feature Knows You Better Than You Know Yourself (Because Algorithms)

The Huffington Post | By Dino Grandoni  





Posted: 08/21/2013 1:44 pm EDT | Updated: 08/22/2013 8:31 am EDT








 55 people like this. Be the first of your friends.



Getty

301227107













 Share  Tweet  Email  Comment



























GET TECHNOLOGY NEWSLETTERS:













Enter email












SUBSCRIBE





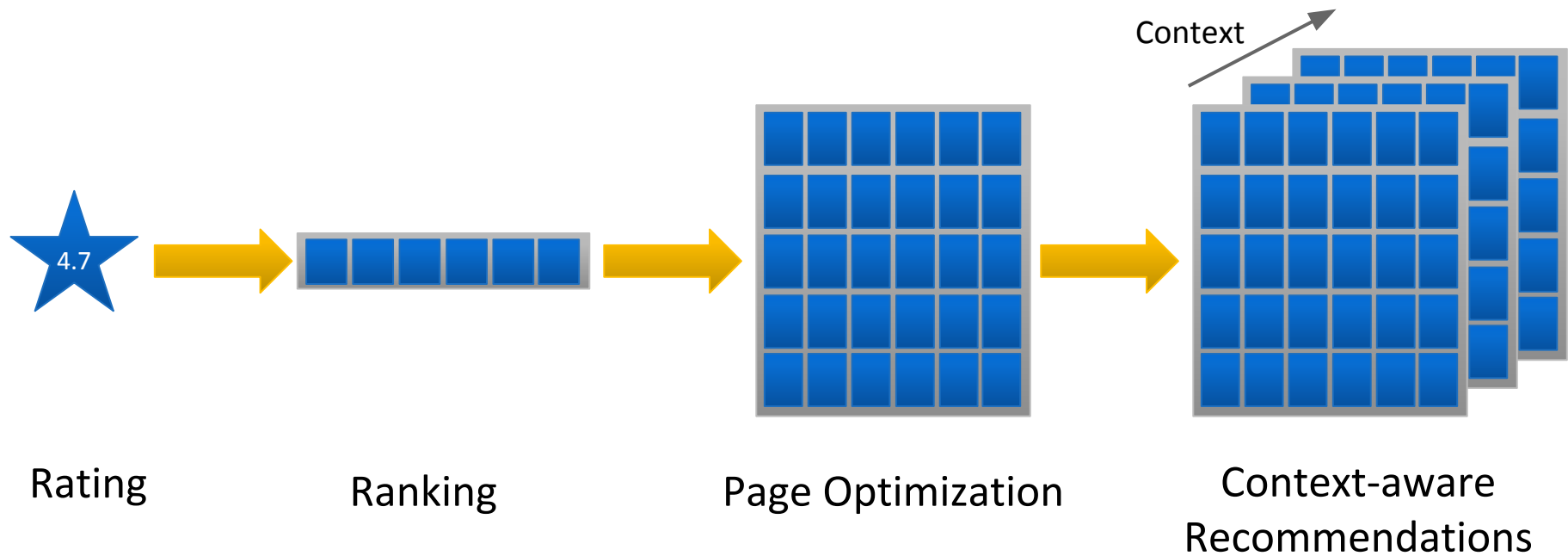








# Evolution of the Recommender Problem



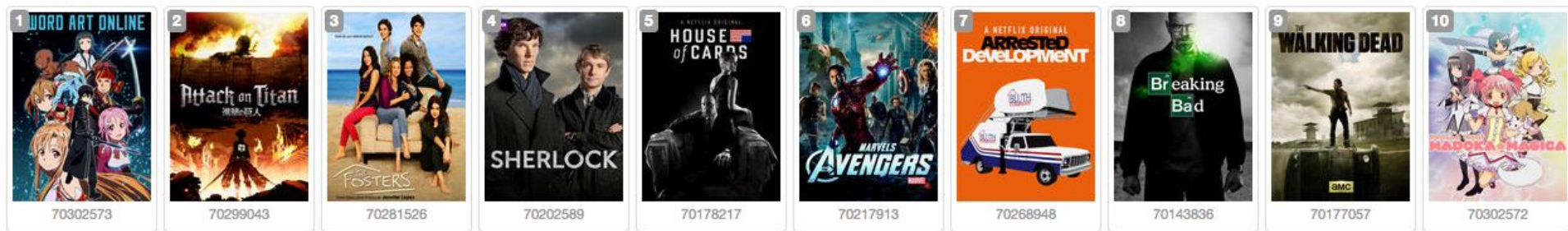
# 3.1 Ranking

# Ranking

---

- Most recommendations are presented in a sorted list
- Recommendation can be understood as a ranking problem
- Popularity is the obvious baseline
- What about rating predictions?

# Ranking by ratings



4.7

4.6

4.5

4.5

4.5

4.5

4.5

4.5

4.5

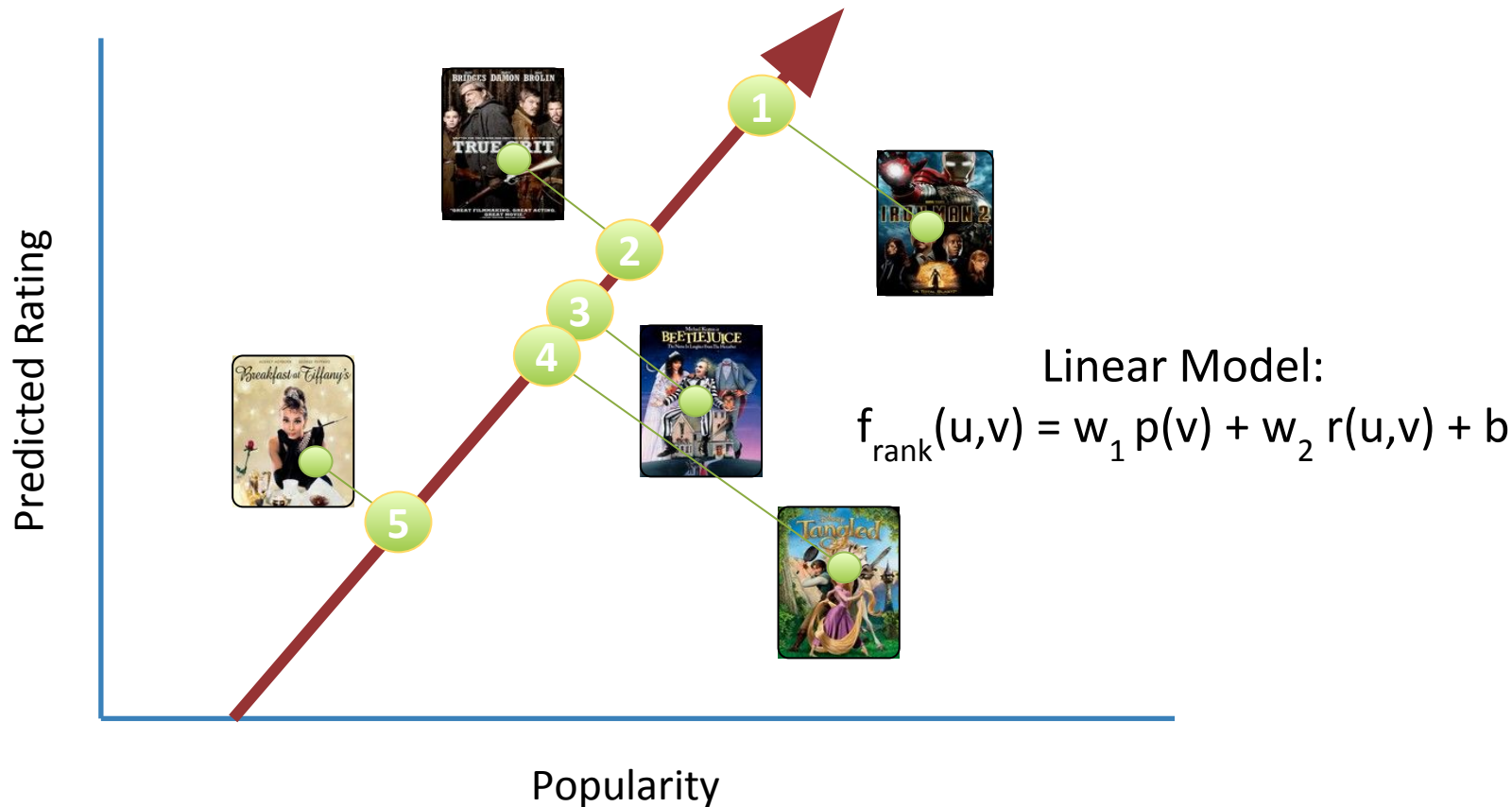
4.5

Niche titles

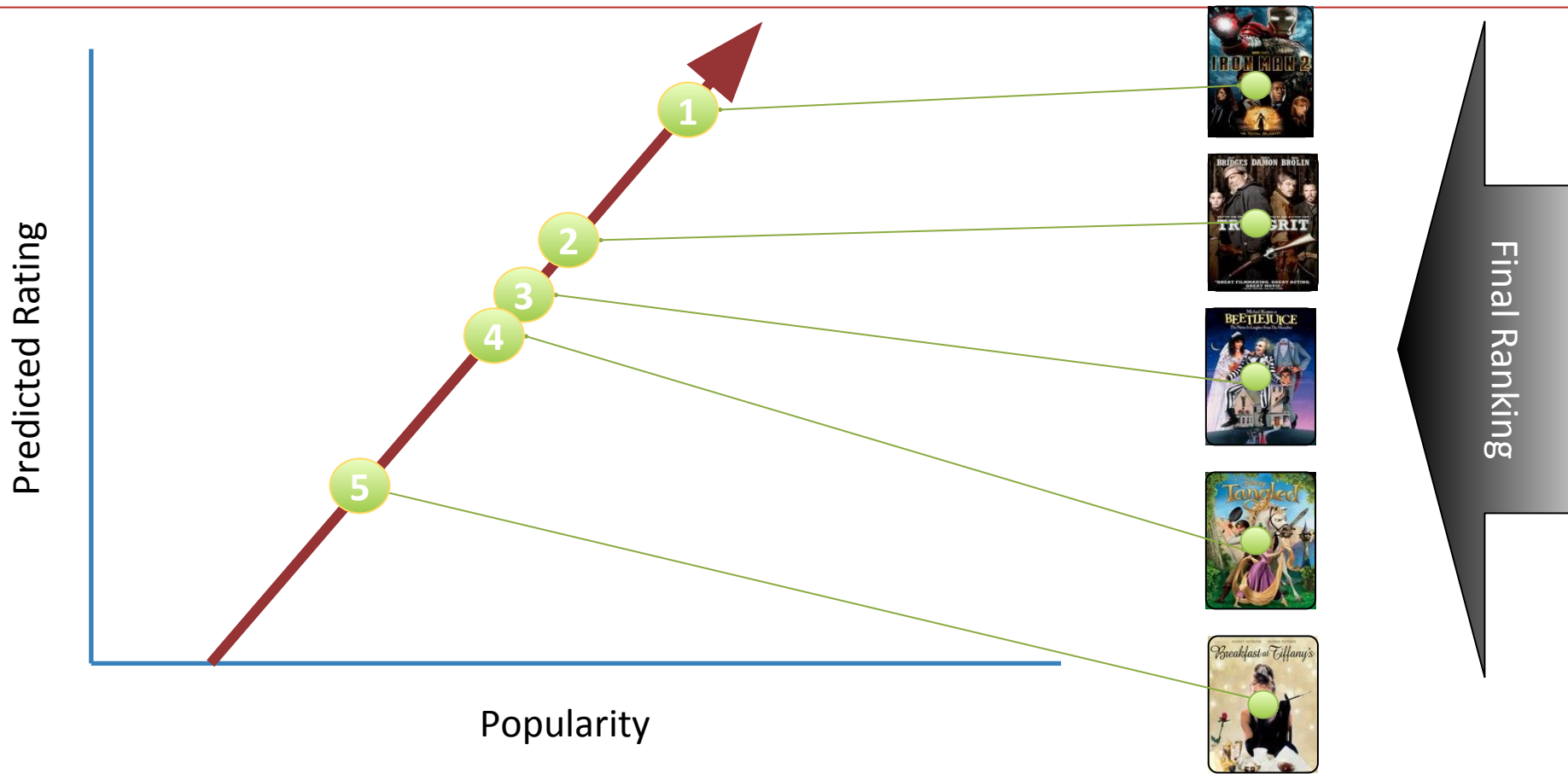
High average ratings... by those who would watch it



# Example: Two features, linear model



# Example: Two features, linear model



# Learning to rank

---

- Machine learning problem: goal is to construct ranking model from training data
- Training data can be a partial order or binary judgments (relevant/not relevant).
- Resulting order of the items typically induced from a numerical score
- Learning to rank is a key element for personalization
- You can treat the problem as a standard supervised classification problem



# Learning to rank - Metrics

- Quality of ranking measured using metrics as
  - Normalized Discounted Cumulative Gain
  - Mean Reciprocal Rank (MRR)
  - Fraction of Concordant Pairs (FCP)
  - Others...
- But, it is hard to optimize machine-learned models directly on these measures (e.g. non-differentiable)
- Recent research on models that directly optimize ranking measures

# Ranking - Quora Feed

**Goal:** Present most *interesting stories* for a

**Interesting** = topical relevance +  
social relevance + timeliness

**Stories** = questions + answers

ML: Personalized learning-to-rank approach

Relevance-ordered vs time-ordered = big g

Business Intelligence Answers wanted • 1m

**What were the steps and experiences that Quora went through when they started to build out their data science and intelligence team?**

Want Answers | 28

Write Answer

Share Downvote

...

Computer Programming

Tommy MacWilliam wrote this • 20 Dec

**How does a blind computer programmer do programming?**



Tommy MacWilliam, Quora Mobile Engineer

100 upvotes by Adrien Lucas Ecoffet, Eyob Fitwi Abraham, Charles Prakash Dasari,

(more)

Ever done a Python Bee before? Now imagine every day of your life is like that. One of my best friends in high school was diagnosed with Leber's hereditary optic neuropathy his senior year. LHON i... (more)

Upvote | 100

Downvote Comment Share | 2

...

Football (Soccer) Answers wanted • 4m

**Why was Mourinho so eager to replace Cech?**

Want Answers | 1

Write Answer

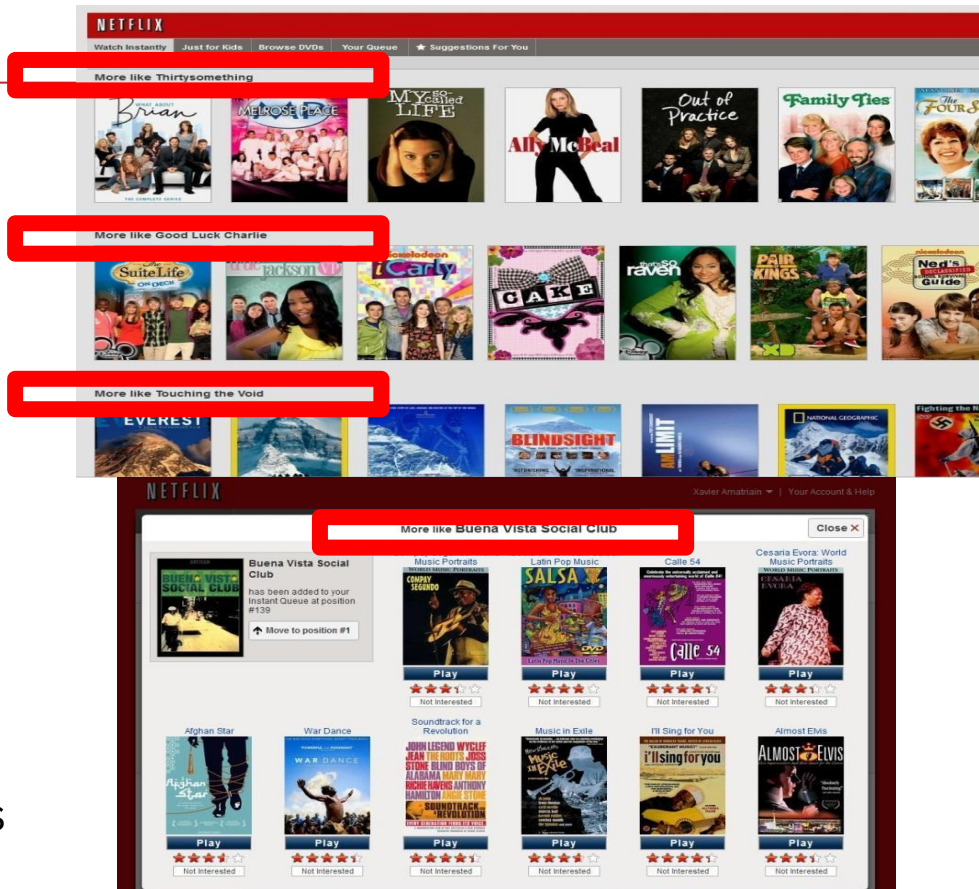
Share Downvote

...

## 3.2 Similarity

# Similars

- Displayed in many different contexts
  - In response to user actions/context (search, queue add...)
  - More like... rows



# Similar: Related Questions

- Given interest in question A (source) what other questions will be interesting?
- Not only about similarity, but also “interestingness”
- Features such as:
  - Textual
  - Co-visit
  - Topics
  - ...
- Important for logged-out use case

## RELATED QUESTIONS

How do you decide to regularize between L1/L2 or best/greedy subset selection?

What's a good way to provide intuition as to why the lasso (L1 regularization) results in sparse weight vectors?

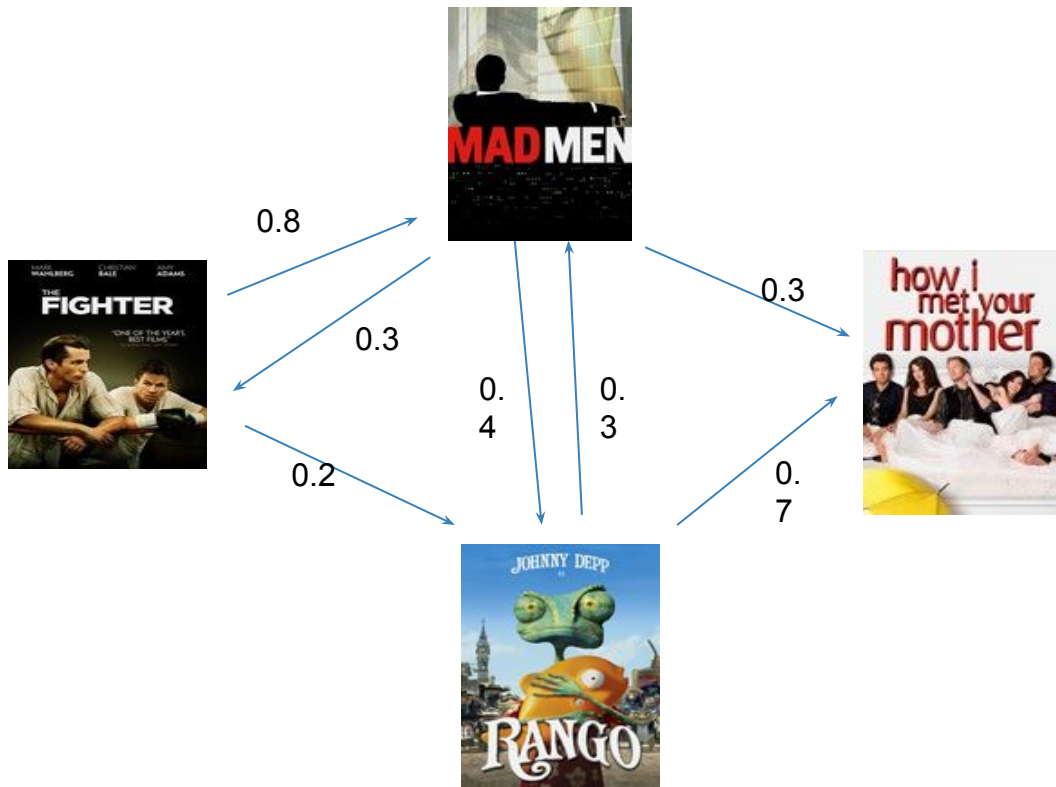
What is the difference between normalization, standardization, and regularization for data?

Why is L1 regularization supposed to lead to sparsity than L2?

What are the conditions of using L1 and L2 regularization respectively?

What are some papers/talks/lectures/notes that give high-level overviews of regularization, especially L1 and L2 regulariz... (continue)

# Graph-based similarities



# Example of graph-based similarity: SimRank

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

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

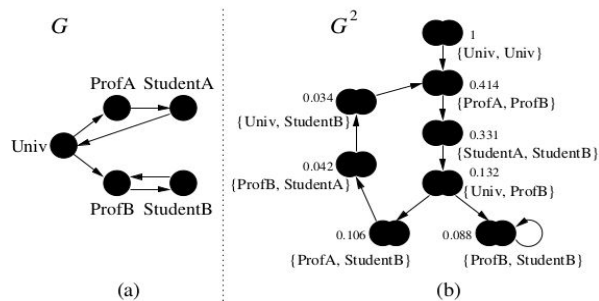


Figure 1: A small Web graph  $G$  and simplified node-pairs graph  $G^2$ . SimRank scores using parameter  $C = 0.8$  are shown for nodes in  $G^2$ .

# Similarity ensembles

- Similarity can refer to different dimensions
  - Similar in metadata/tags
  - Similar in user play behavior
  - Similar in user rating behavior
  - ...
- Combine them using an ensemble
  - Weights are learned using regression over existing response
  - Or... some MAB explore/exploit approach
- The final concept of “similarity” responds to what users vote as similar



## **3.3 Social Recommendations**

# Recommendations - Users

- **Goal: Recommend new users to follow**
- Based on:
  - Other users followed
  - Topics followed
  - User interactions
  - User-related features
  - ...

 Discover new people



**James Altucher**  
Blogger, author, soc...

Followed by Alaka Halder  
and 16 more

[Follow](#) | 49.5k



**Feifei Wang**  
用舍由时，行藏在我

Followed by Emily Nakano  
Co and 7 more

[Follow](#) | 24.6k



**Ellen Vrana**  
Writer

Followed by Katie Hoban  
and 15 more

[Follow](#) | 25.1k

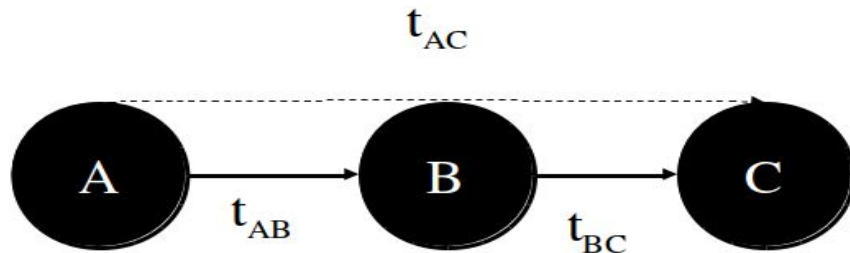
# User Trust/Expertise Inference

- **Goal: Infer user's trustworthiness in relation to a given topic**
- We take into account:
  - Answers written on topic
  - Upvotes/downvotes received
  - Endorsements
  - ...
- Trust/expertise propagates through the network
- Must be taken into account by other algorithms



# Social and Trust-based recommenders

- A social recommender system recommends items that are “popular” in the social proximity of the user.
- Social proximity = trust (can also be topic-specific)
- Given two individuals - the *source* (node A) and *sink* (node C) - derive how much the source should trust the sink.
- Algorithms
  - Advogato (Levien)
  - Appleseed (Ziegler and Lausen)
  - MoleTrust (Massa and Avesani)
  - TidalTrust (Golbeck)



## Other ways to use Social

- Social connections can be used in combination with other approaches
- In particular, “friendships” can be fed into collaborative filtering methods in different ways
  - replace or modify user-user “similarity” by using social network information
  - use social connection as a part of the ML objective function as regularizer
  - ...

## 3.4 Explore/Exploit

# Explore/Exploit

- One of the key issues when building any kind of personalization algorithm is how to trade off:
  - **Exploitation**: Cashing in on what we know about the user right now
  - **Exploration**: Using the interaction as an opportunity to learn more about the user
- We need to have informed and optimal strategies to drive that tradeoff
  - **Solution**: pick a reasonable set of candidates and show users only “enough” to gather information on them

# Multi-armed Bandits

- Given possible strategies/candidates (slot machines) pick the arm that has the maximum potential of *being good* (minimize **regret**)
- Naive strategy:  $\epsilon$ -greedy
  - Explore with a small probability  $\epsilon$  (e.g. 5%) -> choose an arm at random
  - Exploit with a high probability  $(1 - \epsilon)$  (e.g. 95%) -> choose the best-known arm so far
- Translation to recommender systems
  - Choose an arm = choose an item/choose an algorithm (MAB testing)
- Thompson Sampling

Given a posterior distribution, sample on each iteration and choose the action that maximizes the expected reward



# Multi-armed Bandits

## Explore-Exploit in Top-N Recommender Systems via Gaussian Processes

Hastagiri P Vanchinathan  
ETH Zürich  
hastagiri@inf.ethz.ch

Isidor Nikolic \*  
Microsoft, Zürich  
inikolic@microsoft.com

Fabio De Bona  
Google, Zürich  
fdb@google.com

Andreas Krause  
ETH Zürich  
krausea@ethz.ch

## A Contextual-Bandit Approach to Personalized News Article Recommendation

Lihong Li<sup>†</sup>, Wei Chu<sup>†</sup>,  
<sup>†</sup>Yahoo! Labs  
lihong.chuwei@yahoo-inc.com

John Langford<sup>†</sup>  
<sup>†</sup>Yahoo! Labs  
jl@yahoo-inc.com

Robert E. Schapire<sup>+</sup>  
<sup>+</sup>Dept of Computer Science  
Princeton University  
schapire@cs.princeton.edu

## Context Adaptation in Interactive Recommender Systems

Negar Hariri  
DePaul University  
Chicago, IL 60604, USA  
nhariri@cs.depaul.edu

Bamshad Mobasher  
DePaul University  
Chicago, IL 60604, USA  
mobasher@cs.depaul.edu

Robin Burke  
DePaul University  
Chicago, IL 60604, USA  
rburke@cs.depaul.edu



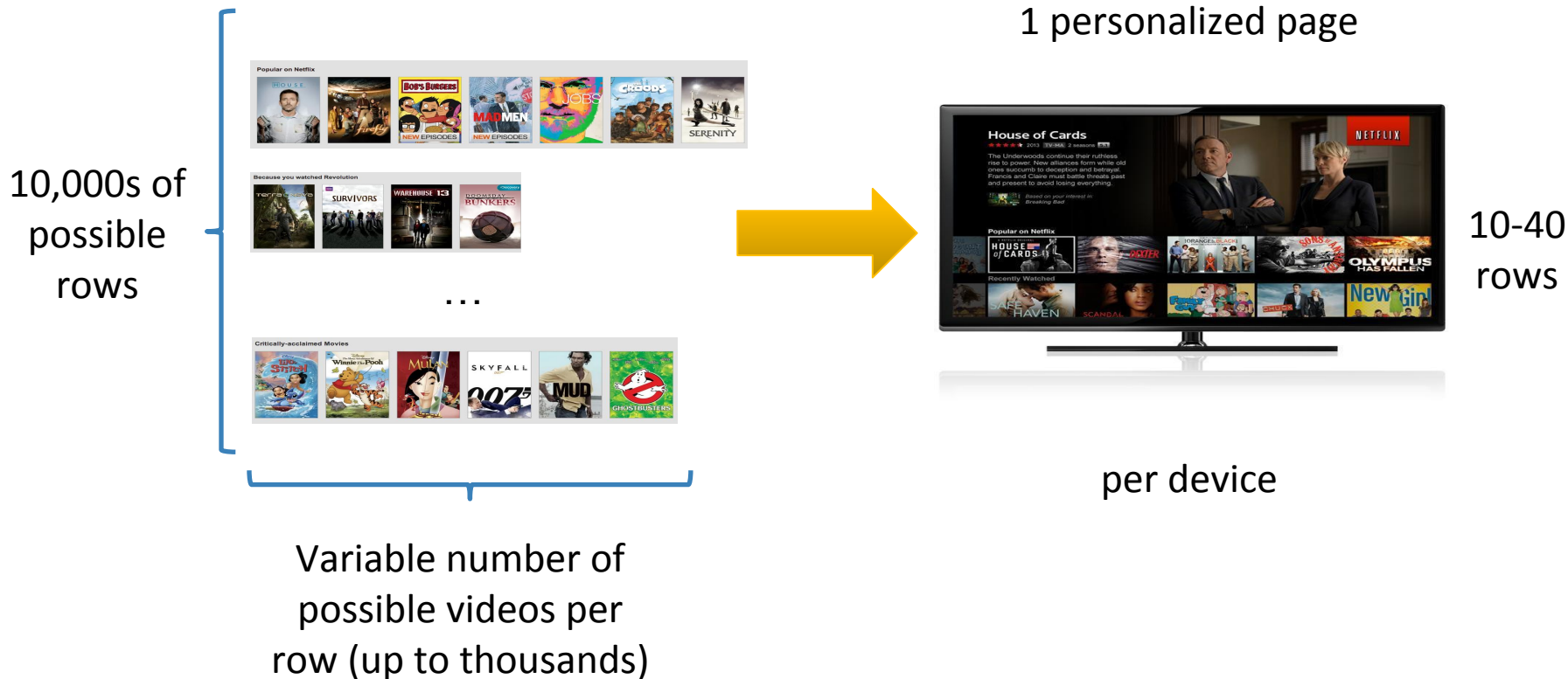
## Recommending Items to Users: An Explore Exploit Perspective

Deepak Agarwal, Director Machine Learning and  
Relevance Science, LinkedIn, USA

CIKM, 2013

# 3.5 Page Optimization

# Page Composition



# Page Composition

The image shows a web browser window with two main sections. The left section displays Google search results for the query "louvre 2006 donation". The right section displays the Wikipedia article for "Nexus 5".

**Search Results (Left):**

- Donating online | Louvre Museum | Paris - Musée du Louvre
- Support the Louvre | Louvre Museum | Paris - Musée du Louvre
- Louvre asks for donations to restore the Nike of Samothrace - Klam
- Louvre Museum
- Louvre Gets \$20 Million for New Islamic Wing - New York Times
- On a Mission to Louvre Up the Louvre - The New York Times
- A Frenchman's life painting donated to the Louvre - The Art Tribune
- Several French drawings donated to the Louvre on condition - The New York Times
- The Louvre - Wikipedia, the free encyclopedia
- Cid Michenero's Make Crucifix Donated to Louvre? - YouTube

**Wikipedia Article (Right):**

**Nexus 5**  
From Wikipedia, the free encyclopedia

The Nexus 5 is a smartphone co-developed by Google and LG Electronics that runs the Android operating system. The successor to the Nexus 4, the device is the fifth smartphone in the Nexus series, a family of Android consumer devices marketed by Google and built by an original equipment manufacturer partner. The Nexus 5 was unveiled on 31 October 2013, and released the same day for online purchase on Google Play, in selected countries.

The Nexus 5's hardware is similar to that of the LG G2, with a Snapdragon 800 system-on-chip (SoC), and a 4.96-inch 1080p display. The Nexus 5 is also the first device to feature version 4.4 of Android.

**Contents** [view]

- 1 Overview
- 2 Specifications
- 3 Accessories
- 4 See also
- 5 References
- 6 External links

**Release** [view]

The Nexus 5 was initially released for ordering at Google Play Store on October 31, 2013, in 16 GB and 32 GB versions.<sup>[a]</sup>

**Specifications** [view]

**Hardware** [view]

The exterior of the Nexus 5 is made from a polycarbonate shell with similarities to the new Nexus 7, unlike its predecessor, which used a glass-based construction.

Its hardware contains similarities to the LG G2; it is powered by a 2.26 GHz quad-core Snapdragon 800 processor with 2 GB of RAM, either 16 or 32 GB of internal storage, and a 2300 mAh battery. The Nexus 5 uses a 4.96-inch (126 mm) 1080p (FHD) display, and includes an 8-megapixel rear-facing camera with optical image stabilization (OIS). The Nexus 5 supports LTE networks where available, unlike the Nexus 4 which unofficially supported LTE on AT&T Band 4 only with a hidden software option, but was not formally approved or marketed for any LTE use.<sup>[b][c]</sup>

There are two variants of the Nexus 5; one is specific to North America (LG-D820), and the other is designed for the rest of the world (LG-D821). The differences between these two variants are in supported cellular frequency bands, see the table on the right for more details.<sup>[1]</sup>

Like its predecessor, the Nexus 5 does not have a microLED card slot.<sup>[1]</sup> While it features a multi-color LED notification light.<sup>[1]</sup> Despite the fact there is a pair of speaker grilles present on the lower edge of the Nexus 5, there is only one speaker; one grille is for a speaker, and another is for a microphone.<sup>[1][d]</sup>

Notable new hardware features also include two new composite sensors: a step detector and a step counter. These new sensors allow applications to easily track steps when the user is walking, running, or climbing stairs. Both sensors are implemented in hardware for low power consumption.<sup>[1]</sup>

**Software** [view]

Nexus 5 is the first Android device to ship with Android 4.4 "KitKat", which has a refreshed interface, improved performance, improved battery life (such as the ability to simulate a sleep state), a new "HDR+" camera shooting mode, native printing functionality, a screen lock utility, and other new and improved functionality.

Nexus 5 ships with Google Experience Launcher (GEL), a redesigned home screen which allows users to access Google Now on a dedicated page, and allows voice search to be activated on the home screen with a voice command. Unlike other features of Android 4.4, GEL is not feature-also part of Android, it is feature-also a component of the Google Search application. As of November 2013, GEL

**Developer:** Google, LG Electronics  
**Manufacturer:** LG Electronics  
**Series:** Google Nexus  
**Compatible networks:** 2G/3G/4G LTE GSM: 850/900/1800/1900 MHz Model LG-D820 (North America) CDMA: band class 1/10/15 WCDMA bands: 10/14/5/6/19 LTE bands: 10/14/15/19/20/24/41 Model LG-D821 (rest of World) WCDMA bands: 10/14/5/6/19 LTE bands: 10/14/15/19

**First released:** October 31, 2013, 35 days ago  
**Availability by country:** 31 October 2013 (France) 30 November 2013 (Germany) 24 November 2013 (Spain)

**Processor:** Nexus 5  
**Rated:** 1.0 (2)

**Form factor:** Slab  
**Dimensions:** 137.36 mm (5.407 in) x 69.17 mm (2.723 in) x 9.16 mm (0.359 in)  
**Weight:** 138 g (4.9 oz) (138 g)  
**Operating system:** Android 4.4  
**System on:** Qualcomm Snapdragon 800  
**Chip:** 2.26 GHz  
**OS:** Android 4.4  
**Memory:** 2 GB RAM

From "Modeling User Attention and Interaction on the Web" 2014 - PhD Thesis by Dmitry Lagun (Emory U.)

# User Attention Modeling

## Web Search (Google)    Social Network (Twitter)



## News (CNN)



## Shopping (Amazon)



From “Modeling User Attention and Interaction on the Web” 2014 - PhD Thesis by Dmitry Lagun (Emory U.)

# Page Composition

Accurate vs. Diverse  
Discovery vs. Continuation  
Depth vs. Coverage  
Freshness vs. Stability  
Recommendations vs. Tasks

- To put things together we need to combine different elements
  - Navigational/Attention Model
  - Personalized Relevance Model
  - Diversity Model

## Beyond Ranking: Optimizing Whole-Page Presentation

Yue Wang<sup>\*</sup>, Dawei Yin<sup>2</sup>, Luo Jie<sup>3</sup>, Pengyuan Wang<sup>2</sup>, Makoto Yamada<sup>2,4</sup>,  
Yi Chang<sup>2</sup>, Qiaozhu Mei<sup>1,2</sup>

<sup>1</sup>Department of EECS, University of Michigan, Ann Arbor, MI, USA

<sup>2</sup>Yahoo Labs, 701 First Avenue, Sunnyvale, CA, USA

<sup>3</sup>Snapchat, Inc., 64 Market St, Venice, CA, USA

<sup>4</sup>Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto, Japan

## Fair and Balanced: Learning to Present News Stories

Amr Ahmed<sup>\*1</sup>, Choon Hui Teo<sup>\*1</sup>, S.V.N. Vishwanathan<sup>2</sup>, Alex Smola<sup>1</sup>

<sup>\*</sup> Co-first authors.

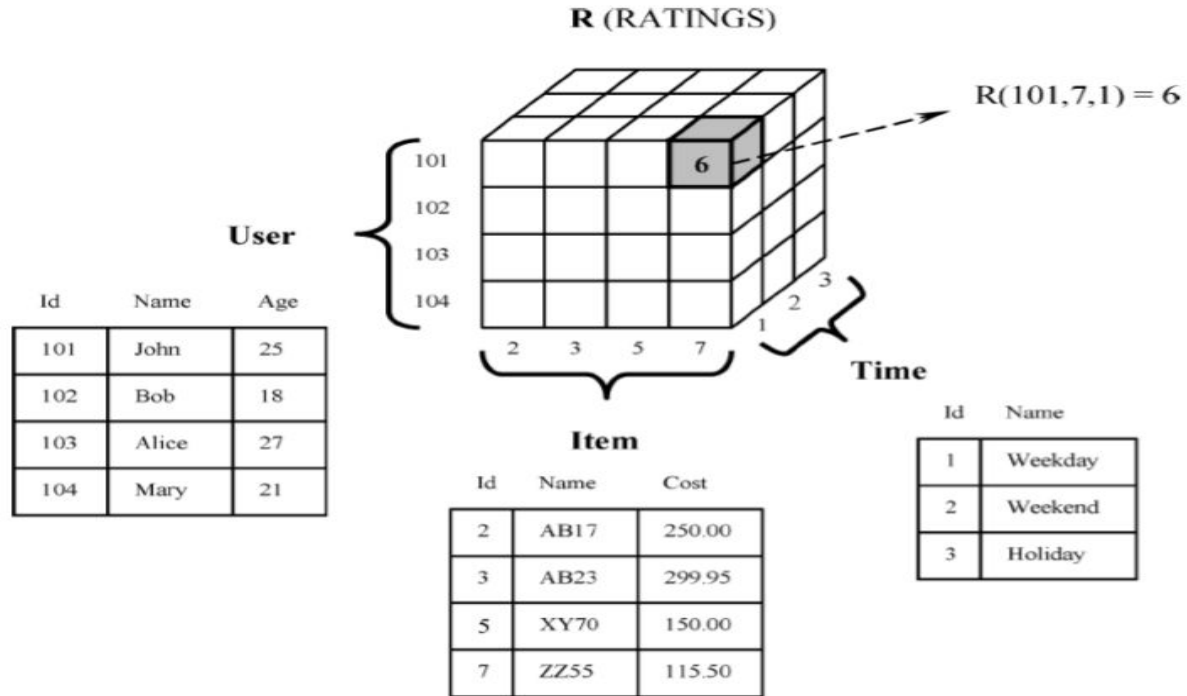
<sup>1</sup>Yahoo! Research, Santa Clara, CA 95053, USA

<sup>2</sup>Purdue University, West Lafayette, IN 47907, USA

{amahmed,choonhui,smola}@yahoo-inc.com, vishy@stat.purdue.edu

## 3.6 Beyond user/rating

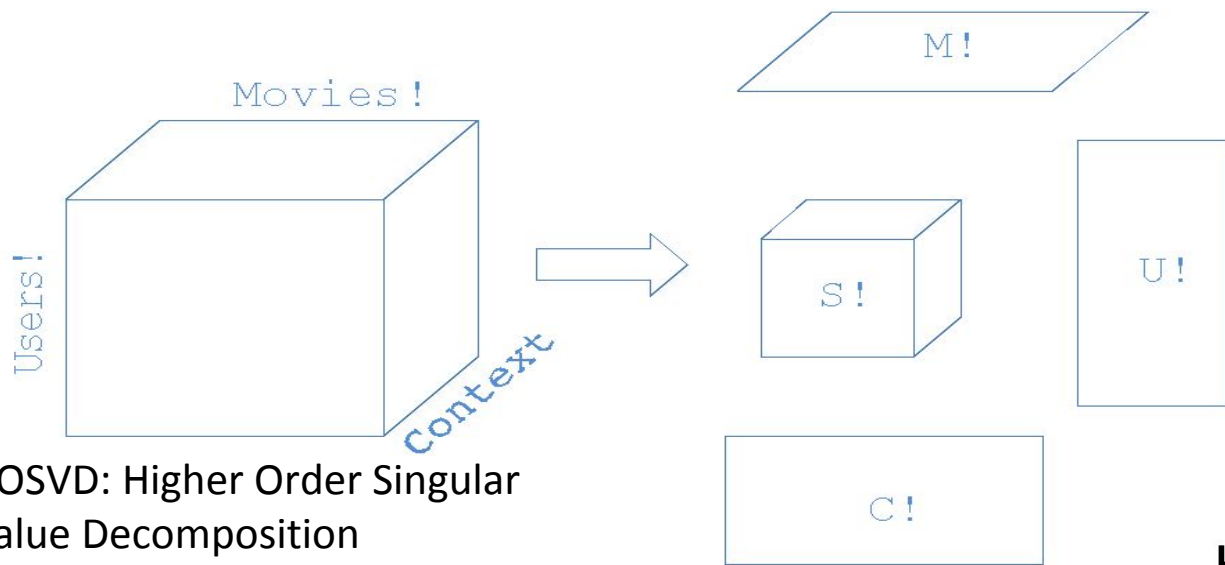
# N-dimensional model



[Adomavicius et al., 2005]



# Tensor Factorization



HOSVD: Higher Order Singular  
Value Decomposition

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

HOSVD  
Model

$$R[U, M, C, S] := L(F, Y) + \Omega[U, M, C] + \Omega[S]$$

# Tensor Factorization

$$R[U, M, C, S] := L(F, Y) + \Omega[U, M, C] + \Omega[S]$$

Where:

$$\Omega[F] = \lambda_M \|M\|_F^2 + \lambda_U \|U\|_F^2 + \lambda_C \|C\|_F^2 \quad \Omega[S] := \lambda_S \|S\|_F^2$$

- We can use a simple squared error loss function:

$$l(f, y) = \frac{1}{2}(f - y)^2$$

- Or the absolute error loss

$$l(f, y) = |f - y|$$

- The loss function over all users becomes

$$L(F, Y) = \sum_i^n \sum_j^m l(f_{ij}, y_{ij})$$

# Factorization Machines

- Generalization of regularized matrix (and tensor) factorization approaches combined with linear (or logistic) regression
- Problem: Each new adaptation of matrix or tensor factorization requires deriving new learning algorithms
  - Hard to adapt to new domains and add data sources
  - Hard to advance the learning algorithms across approaches
  - Hard to incorporate non-categorical variables

# Factorization Machines

- Approach: Treat input as a real-valued feature vector
  - Model both linear and pair-wise interaction of  $k$  features (i.e. polynomial regression)
  - Traditional machine learning will overfit
  - Factor pairwise interactions between features
  - Reduced dimensionality of interactions promote generalization
  - Different matrix factorizations become different feature representations
  - Tensors: Additional higher-order interactions
- Combines “generality of machine learning/regression with quality of factorization models”

# Factorization Machines

- Each feature gets a weight value and a factor vector
  - $O(dk)$  parameters

$$b \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^d, \mathbf{V} \in \mathbb{R}^{d \times k}$$

- Model equation:

$$\begin{aligned} f(\mathbf{x}) &= b + \sum_{i=1}^d w_i x_i + \sum_{i=1}^d \sum_{j=i+1}^d x_i x_j \mathbf{v}_i^T \mathbf{v}_j && O(d^2) \\ &= b + \sum_{i=1}^d w_i x_i + \frac{1}{2} \sum_{f=1}^k \left( \left( \sum_{i=1}^d x_i v_{i,f} \right)^2 - \sum_{i=1}^d x_i^2 v_{i,f}^2 \right) && O(kd) \end{aligned}$$

# Factorization Machines

- Two categorical variables ( $u, i$ ) encoded as real values:

Feature vector  $\mathbf{x}$

$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...
	A	B	C	...	TI	NH	SW	ST	...
	User				Movie				

- FM becomes identical to MF with biases:

$$f(\mathbf{x}) = b + w_u + w_i + \mathbf{v}_u^T \mathbf{v}_i$$

*From Rendle (2012) KDD Tutorial*

# Factorization Machines

- Makes it easy to add a time signal

Feature vector $\mathbf{x}$											
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.2	
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.6	
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.61	
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0.3	
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0.5	
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.1	
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.8	
	A	B	C	...	TI	NH	SW	ST	...		
	User				Movie						
											Time

- Equivalent equation:

$$f(\mathbf{x}) = b + w_u + w_i + x_t w_t + \mathbf{v}_u^T \mathbf{v}_i + x_t \mathbf{v}_u^T \mathbf{v}_t + x_t \mathbf{v}_i^T \mathbf{v}_t$$

From Rendle (2012) KDD Tutorial

# Factorization Machines (Rendle, 2010)

- L2 regularized
  - Regression: Optimize RMSE
  - Classification: Optimize logistic log-likelihood
  - Ranking: Optimize scores

Gradient:

$$\frac{\partial}{\partial \theta} f(\mathbf{x}) = \begin{cases} 1 & \text{if } \theta \text{ is } b \\ x_i & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^d v_{j,f} x_j - v_{i,f} x_i^2 & \text{if } \theta \text{ is } v_{i,f} \end{cases}$$

- Can be trained using:

- SGD
- Adaptive SGD
- ALS
- MCMC

Least squares SGD:

$$\theta' = \theta - \eta \left( (f(\mathbf{x}) - y) \frac{\partial}{\partial \theta} f(\mathbf{x}) + \lambda_{\theta} \theta \right)$$



# Factorization Machines (Rendle, 2010)

- Learning parameters:
  - Number of factors
  - Iterations
  - Initialization scale
  - Regularization (SGD, ALS) – Multiple
  - Step size (SGD, A-SGD)
  - MCMC removes the need to set those hyperparameters

# 3.7 Deep Learning

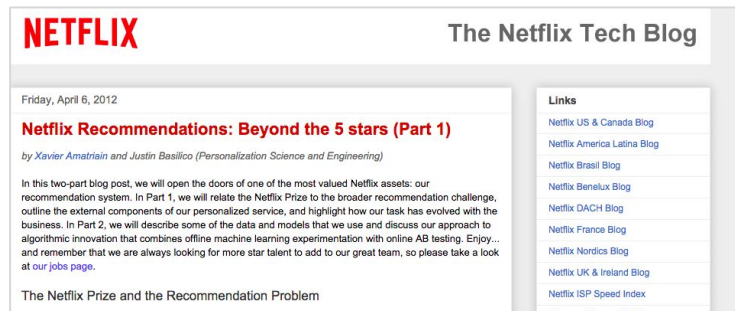
(See Balázs Hidasi's slides)

# 4. Lessons Learned

1. IMPLICIT SIGNALS BEAT  
EXPLICIT ONES  
(ALMOST ALWAYS)

# Implicit vs. Explicit

- Many have acknowledged that implicit feedback is more useful
- Is implicit feedback really always more useful?
- If so, why?



Seems like when it comes to ratings it's pretty much all or nothing. Great videos prompt action; anything less prompts indifference. Thus, the ratings system is primarily being used as a seal of approval, not as an editorial indicator of what the community thinks about a video. Rating a video joins favoriting and sharing as a way to tell the world that this is something you love.

# Implicit vs. Explicit

- Implicit data is (usually):
  - More dense, and available for all users
  - Better representative of user behavior vs. user reflection
  - More related to final objective function
  - Better correlated with AB test results
- E.g. Rating vs watching

The image displays two screenshots from the IMDb website, illustrating movie data categorized by grossing and rating.

**Top-US-Grossing Feature Films Released In 2014**

1-50 of 9,031 titles.

Sort by: Popularity | A-Z | User Rating | Num Votes | US Box Office | Runtime | Year | US Release Date

Rank	Movie Title	Year	Rating	Box Office
1	American Sniper	2014	7.3	\$350M
2	The Hunger Games: Mockingjay - Part 1	2014	6.8	\$337M
3	Guardians of the Galaxy	2014	8.1	\$333M
4	Captain America: The Winter Soldier	2014	7.2	\$260M
5	The Lego Movie	2014	7.8	\$258M

**Highest Rated Feature Films Released In 2014**

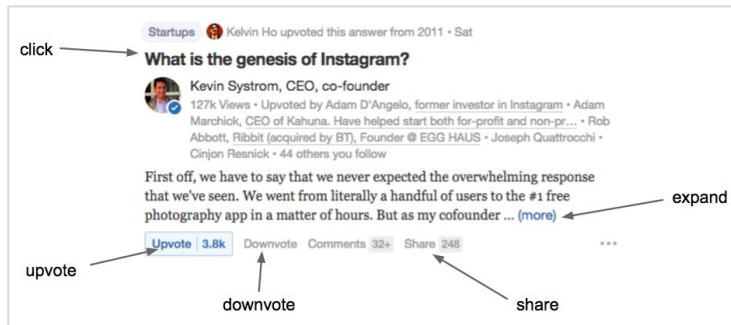
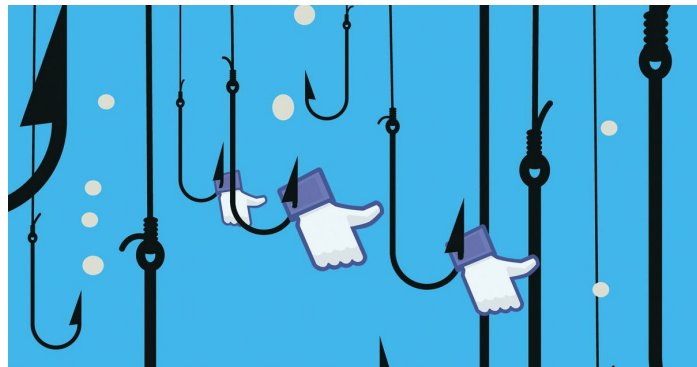
1-50 of 9,031 titles.

Sort by: Popularity | A-Z | User Rating | Num Votes | US Box Office | Runtime | Year | US Release Date

Rank	Movie Title	Year	Rating
1	Forgive and Forget	2014	9.6
2	Mahjong and the West	2014	8.4
3	National Theatre Live: Coriolanus	2014	9.1
4	Burning Dog	2014	8.9
5	The Rule of Law	2014	9.0

# Implicit vs. Explicit

- However
  - It is not always the case that direct implicit feedback correlates well with long-term retention
  - E.g. clickbait
- Solution:
  - Combine different forms of implicit + explicit to better represent long-term goal



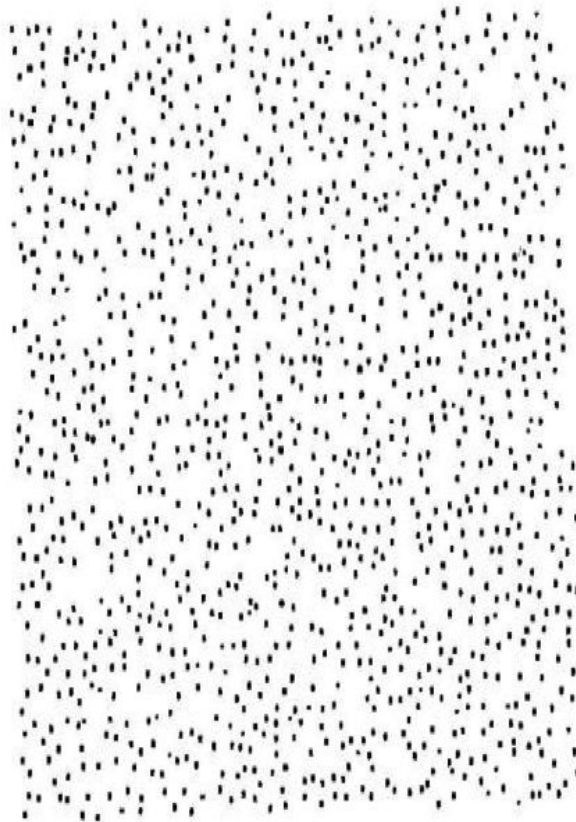


## 2. BE THOUGHTFUL ABOUT YOUR TRAINING DATA



# Defining training/testing data

- Training a simple binary classifier for good/bad answer
  - Defining positive and negative labels -> Non-trivial task
    - *Is this a positive or a negative?*
      - funny uninformative answer with many upvotes
      - short uninformative answer by a well-known expert in the field
      - very long informative answer that nobody reads/upvotes
      - informative answer with grammar/spelling mistakes
      - ...



3. YOUR MODEL WILL LEARN  
WHAT YOU TEACH IT TO LEARN

# Training a model

- Model will learn according to:
  - Training data (e.g. implicit and explicit)
  - Target function (e.g. probability of user reading an answer)
  - Metric (e.g. precision vs. recall)
- Example 1 (made up):
  - *Optimize probability of a user going to the cinema to watch a movie and rate it “highly” by using purchase history and previous ratings. Use NDCG of the ranking as final metric using only movies rated 4 or higher as positives.*

## Example 2 - Quora's feed

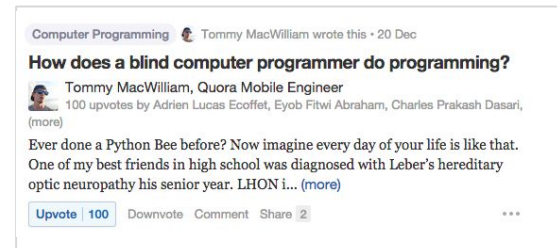
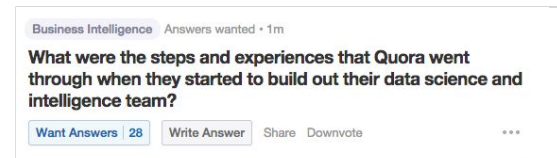
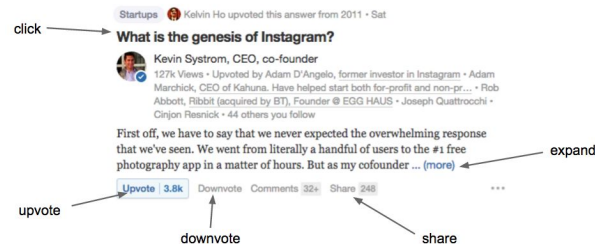
- Training data = implicit + explicit
- Target function: Value of showing a story to a

user  $\sim$  weighted sum of actions:

$$v = \sum_a v_a 1\{y_a = 1\}$$

- predict probabilities for each action, then compute expected value:  $v_{\text{pred}} = E[V | x] = \sum_a v_a p(a | x)$

- Metric: any ranking metric



4. EXPLANATIONS MIGHT MATTER  
MORE THAN THE PREDICTION



# Explanation/Support for Recommendations

 Sarah Smith  Richard Henry and 3 more upvoted this • 7h

**How can I complain about my roommate who is cheating on his Google phone interviews?**



Ben Garrison, Software Engineer at Google

304.3k Views • Upvoted by Jeremy Miles, Quantitative analyst at Google, Mayeesha Tahsin, Sarah Smith, and 3 others you follow

First off, I really appreciate your trying to make sure the right thing happens. I think that's great. Cheating sucks. However, the answer is "don't worry about it". Phone screens here at Google ar... [\(more\)](#)

Upvote 968 Downvote Comments 23+ Share

 Discover new topics

 **Last.fm**  
Last.fm builds detail...  
Followed by Neal Lathia and 8 more

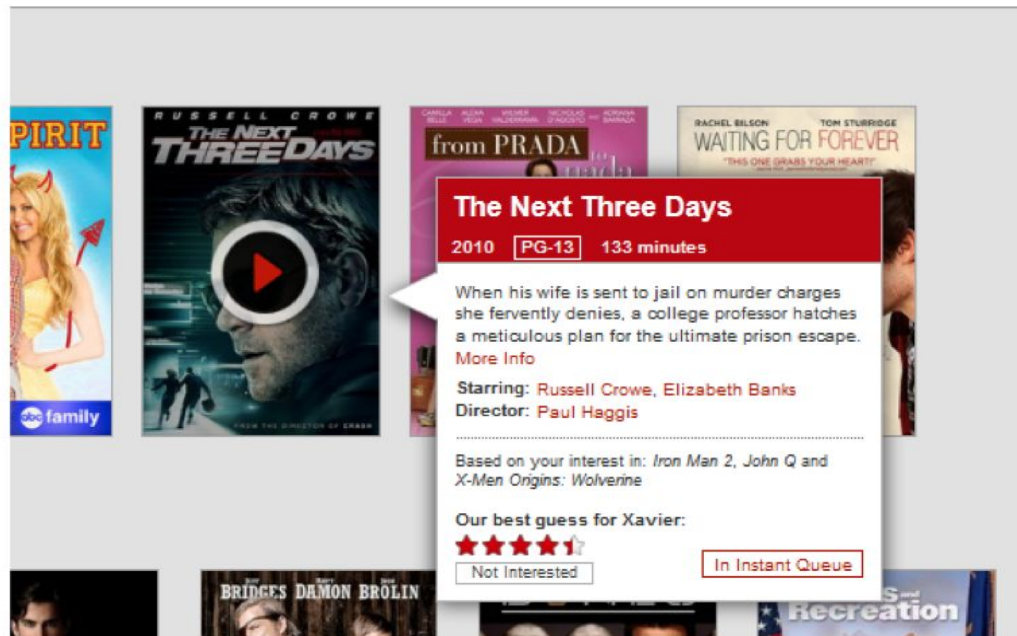
Follow 21.9k

 **Quantitative Finance**  
Quantitative finance ...  
Followed by Katie Hoban and 22 more

Follow 74.1k

 **California Stat**  
California State  
Followed by Rachelle Baratto

Follow 2.4k



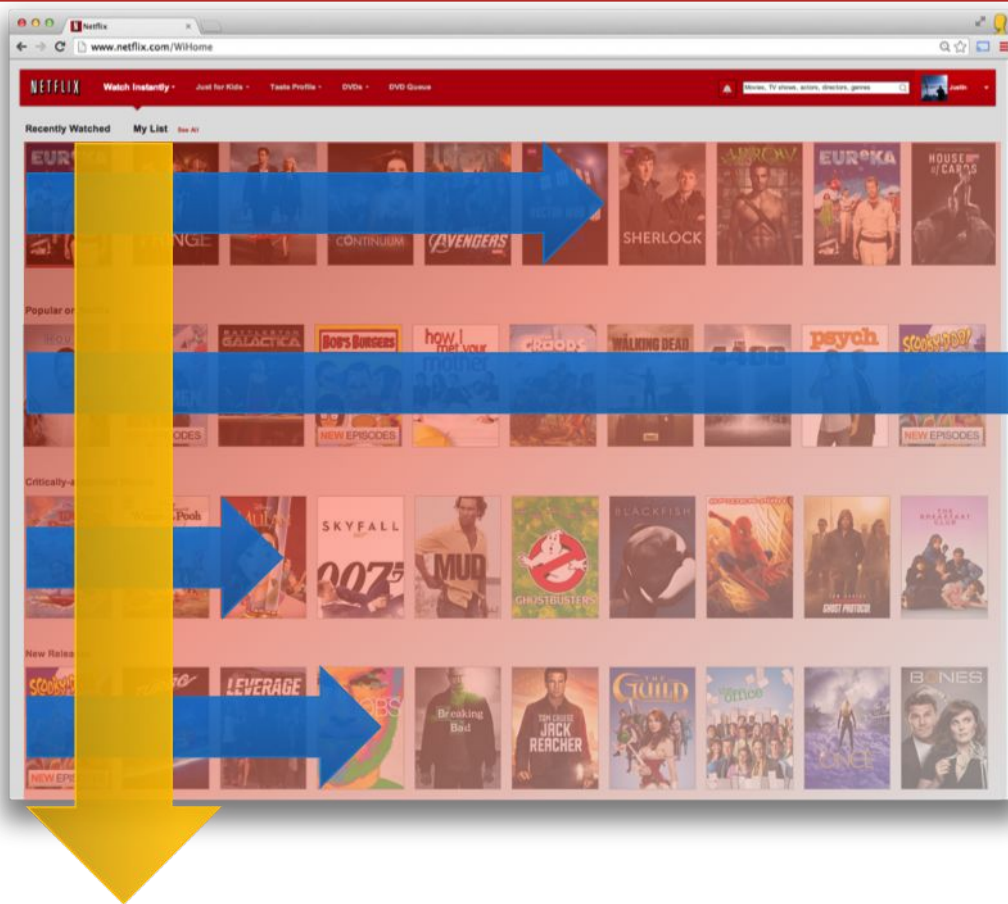
The interface displays a grid of movie posters. A large play button is overlaid on the poster for 'The Next Three Days'. A detailed overlay for this movie provides the following information:

- Title:** The Next Three Days
- Year:** 2010
- Rating:** PG-13
- Duration:** 133 minutes
- Description:** When his wife is sent to jail on murder charges she fervently denies, a college professor hatches a meticulous plan for the ultimate prison escape.
- More Info:** [More Info](#)
- Starring:** Russell Crowe, Elizabeth Banks
- Director:** Paul Haggis
- Recommendation Basis:** Based on your interest in: Iron Man 2, John Q and X-Men Origins: Wolverine
- Our best guess for Xavier:** ★★★★★
- Buttons:** Not Interested, In Instant Queue

# 5. LEARN TO DEAL WITH PRESENTATION BIAS

# 2D Navigational modeling

More likely  
to see



Less likely



# The curse of presentation bias

- User can only click on what you decide to show
  - But, what you decide to show is the result of what your model predicted is good
- Simply treating things you show as negatives is not likely to work
- Better options
  - Correcting for the probability a user will click on a position -> Attention models
  - Explore/exploit approaches such as MAB

**Collaborative Competitive Filtering: Learning Recommender Using Context of User Choice**

Shuang Hong Yang  
Georgia Tech  
shy@gatech.edu

Bo Long  
Yahoo! Labs  
bolong@yahoo-inc.com

Alex Smola  
Yahoo! Research  
smola@yahoo-inc.com

Hongyuan Zha  
Georgia Tech  
zha@cc.gatech.edu

Zhaohui Zheng  
Yahoo! Labs Beijing  
zhaohui@yahoo-inc.com

6. IF YOU HAVE TO PICK ONE SINGLE  
APPROACH, MATRIX FACTORIZATION IS YOUR  
BEST BET

- MF can be interpreted as

- Unsupervised:
  - Dimensionality Reduction a la PCA
  - Clustering (e.g. NMF)
- Supervised:
  - Labeled targets  $\sim$  regression

$$\begin{matrix} & d \\ n & \mathbf{X} \end{matrix} = \begin{matrix} & h \\ n & \mathbf{U} \end{matrix} \times \begin{matrix} & d \\ h & \mathbf{V}^T \end{matrix}$$

- Very useful variations of MF

- BPR, ALS, SVD++
- Tensor Factorization, Factorization Machines

- However...

7. EVERYTHING IS AN ENSEMBLE

- Netflix Prize was won by an ensemble
  - Initially Bellkor was using GDBTs
  - BigChaos introduced ANN-based ensemble
- Most practical applications of ML run an ensemble
  - Why wouldn't you?
  - At least as good as the best of your methods
  - Can combine different approaches (e.g. CF and content-based)
  - Can use different models at the ensemble layer: LR, GDBTs, RFs, ANNs...

## The BellKor Solution to the Netflix Grand Prize

Yehuda Koren  
August 2009

## The BigChaos Solution to the Netflix Grand Prize

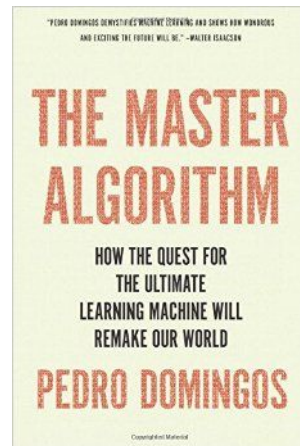
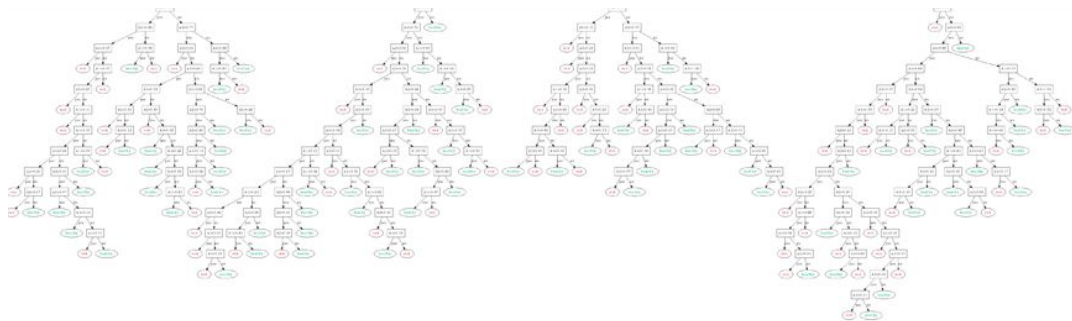
Andreas Töschel and Michael Jahrer  
*commendo research & consulting*  
Neuer Weg 23, A-8580 Köflach, Austria  
{andreas.toeschel,michael.jahrer}@commendo.at

Robert M. Bell\*  
*AT&T Labs - Research*  
Florham Park, NJ

September 5, 2009

# Ensembles & Feature Engineering

- Ensembles are the way to turn any model into a feature!
- E.g. Don't know if the way to go is to use Factorization Machines, Tensor Factorization, or RNNs?
  - Treat each model as a “feature”
  - Feed them into an ensemble





8. BUILDING RECOMMENDER SYSTEMS IS  
ALSO ABOUT FEATURE ENGINEERING

## Need for feature engineering

In many cases an understanding of the domain will lead to optimal results.



## What is a good Quora answer?

- truthful
- reusable
- provides explanation
- well formatted
- ...

### What music do data scientists usually listen to while working?



**Paula Griffin**, data scientist and biostatistics PhD ... (more)

13 upvotes by William Chen, Alexandr Wang (王誉舜), Sheila Christine Lee, (more)

I was figuring that this question was just fishing for someone to answer that Big Data is their favorite band. Unfortunately, the question log indicates this was asked about 6 months before their EP came out, so there goes that theory.

This is going to be a pretty odd list, but here's the list, in order of decreasing social acceptability:

- Electropop -- Banks and CHVRCHES are my favorites at the moment.
- Miscellaneous alt-rock -- this category basically includes anything I found out about from listening to Sirius XM in the car.
- Nerd rock -- What kind of geek would I be if Jonathan Coulton wasn't on this list?



**Shankar Iyer**, data scientist at Quora

10 upvotes by William Chen, Sheila Christine Lee, Don van der Drift, (more)

Based on the Pandora stations that I've been listening to, my recent work-time listening consists of:

1. **Acoustic folk music:** John Fahey, Leo Kottke, Six Organs of Admittance, etc.
2. **Post-Rock / Ambient Music:** Sigur Rós, Gregor Samsa, the Japanese Mono, Eluvium, El Ten Eleven, etc.
3. **Hindustani:** mostly Vishwa Mohan Bhatt
4. **Carnatic:** recently Rajeswari Pariti
5. **Classical Guitar:** recently Paul Galbraith, Konrad Ragossnig, etc.

## How are those dimensions translated into features?

- Features that relate to the answer quality itself
- Interaction features (upvotes/downvotes, clicks, comments...)
- User features (e.g. expertise in topic)



**Paula Griffin**, data scientist and biostatistics PhD ... (more)

13 upvotes by William Chen, Alexandr Wang (王誉舜), Sheila Christine Lee, (more)

I was figuring that this question was just fishing for someone to answer that Big Data is their favorite band. Unfortunately, the question log indicates this was asked about 6 months before their EP came out, so there goes that theory.

This is going to be a pretty odd list, but here's the list, in order of decreasing social acceptability:

- Electropop -- Banks and CHVRCHES are my favorites at the moment.
- Miscellaneous alt-rock -- this category basically includes anything I found out about from listening to Sirius XM in the car.
- Nerd rock -- What kind of geek would I be if Jonathan Coulton wasn't on this list?
- Straight-up nostalgia -- I have an admittedly weird habit of listening to the same album (sometimes just one song) over and over for hours on end which was formed during all-nighters in high school. Motion City Soundtrack, Jimmy Eat World, and Weezer are my go-to's in this category.
- Soundtracks of all sorts -- *Chicago*, *Jurassic Park*, *Bastion*, *The Book of Mormon*, the Disney version of *Hercules*... again, basically anything that works on a repeat loop for ~3 hours.
- Pop -- don't make me list the artists. I've already told you I listen to Disney soundtracks; you can't possibly need more dirt on me. The general principle is that if you can dance to it, you can code to it.

Now, if you don't mind, I'm just going to sit at my desk and be super-embarrassed that my coworkers know what's in my headphones.

Written 4 Dec. 353 views. Asked to answer by William Chen.

Upvote

13

Downvote

Comment

Share

...

- Properties of a well-behaved ML feature:
  - Reusable
  - Transformable
  - Interpretable
  - Reliable

## Deep Learning

NIPS'2015 Tutorial

Geoff Hinton, Yoshua Bengio & Yann LeCun



Deep Learning:  
Automating  
Feature Discovery

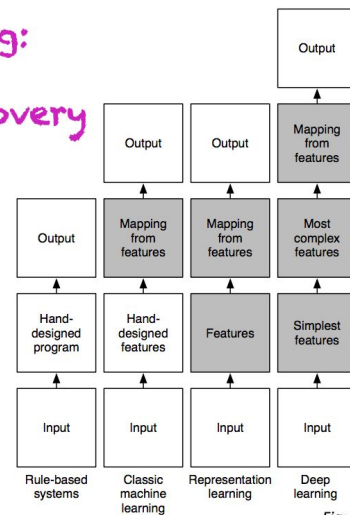
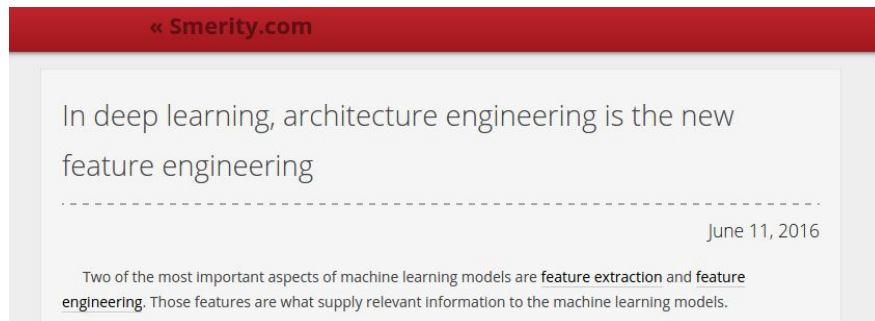
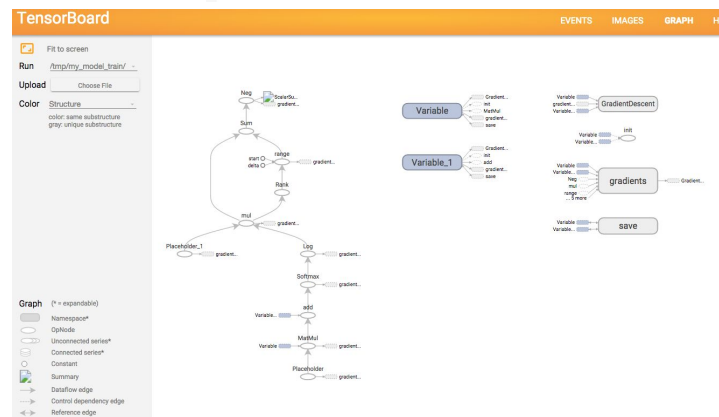
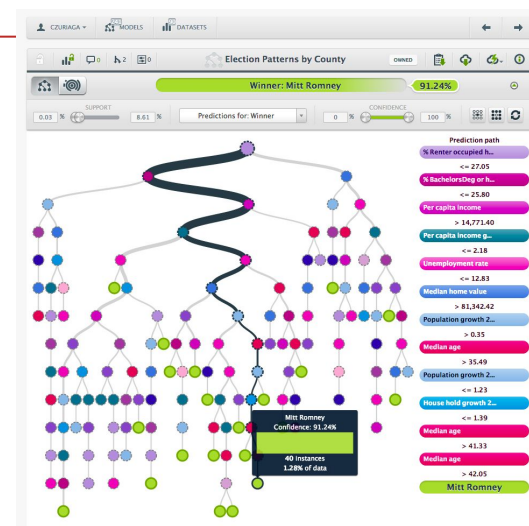


Fig. 1. Goodfellow

9. WHY YOU SHOULD CARE ABOUT  
ANSWERING QUESTIONS  
(ABOUT YOUR RECSYS)

# Model debuggability


- Value of a model = value it brings to the product
- Product owners/stakeholders have expectations on the product
- It is important to answer questions to why did something fail
- Model debuggability is so important it can determine:
  - Particular model to use
  - Features to rely on
  - Implementation of tools





# Model debuggability

- E.g. Why am I seeing or not seeing this on my homepage feed?

Feature Name		aid 14862324	aid 2546362
US	What is more dangerous, road or mountain biking?		
US			
OB	 Jack Rae, Gold medalist at British XC University Champs. President of UoBCC 2011-2012....		
US	Upvoted by Richard Henry · Vo Nghi Nguyen		
US	Encountering <b>minor injuries</b> : You'll get a lot more of that in <b>cross country mountain biking</b> . Brushing a tree, going over the bars and bruising your shoulder, scraping your legs... Maybe even fractu... (more)		
USER LONG HISTORY ACTION TYPE UPVOTE RATE BY STORY TYPE		0.0094589	0.0787334

feed / feature analysis using score / feature analysis using model score

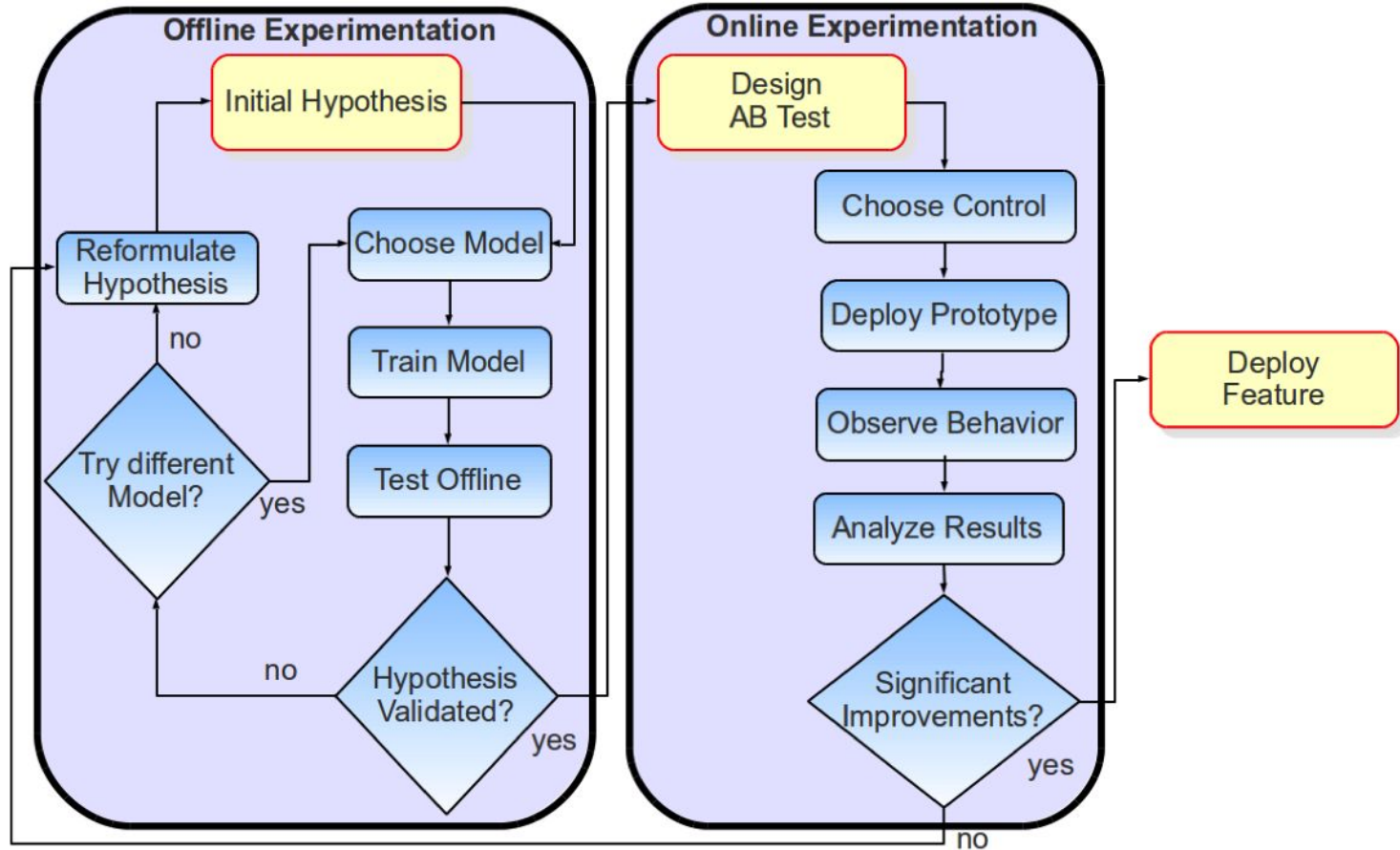
This table shows feature values for the debug story (using feedStory or debug\_aid/qid above) and for the top 10 comparison stories from the same leaf node. For each comparison story, the color (and hover text) of a feature cell shows how the score of the debug story would change if feature values were swapped between the debug story the comparison story. Feature rows are sorted by the maximum absolute score gain among the comparison stories.

Feature Name	aid 14862324	aid 2546362	aid 2296
USER_I	0.0094589	0.2130526	0.213052
USER_I	0.0514545	0.2039045	0.203904
OBJEC'	8	None	7
OBJEC'	128263005100	70919435147759	7538566
USER_I	0.0648323	0.2112874	0.211287
USER_S	0	None	1
USER_I	0.0094589	0.0787334	0.078733
OBJEC'	0	0.3824919	0.245169
OBJEC'	0.1047419	None	None
NUM_R	1	None	None
USER S	0	None	1

10. DATA AND MODELS ARE GREAT. YOU KNOW  
WHAT'S EVEN BETTER?

THE RIGHT EVALUATION APPROACH!

# Offline/Online testing process

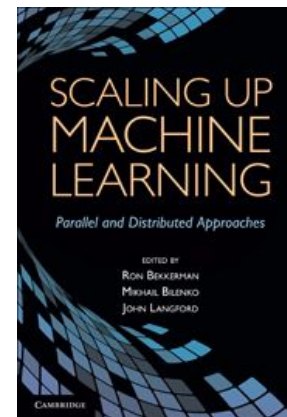
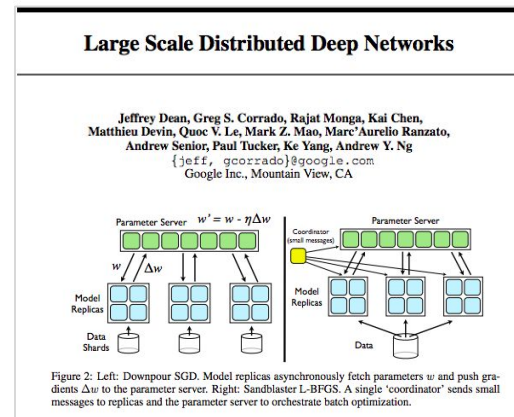




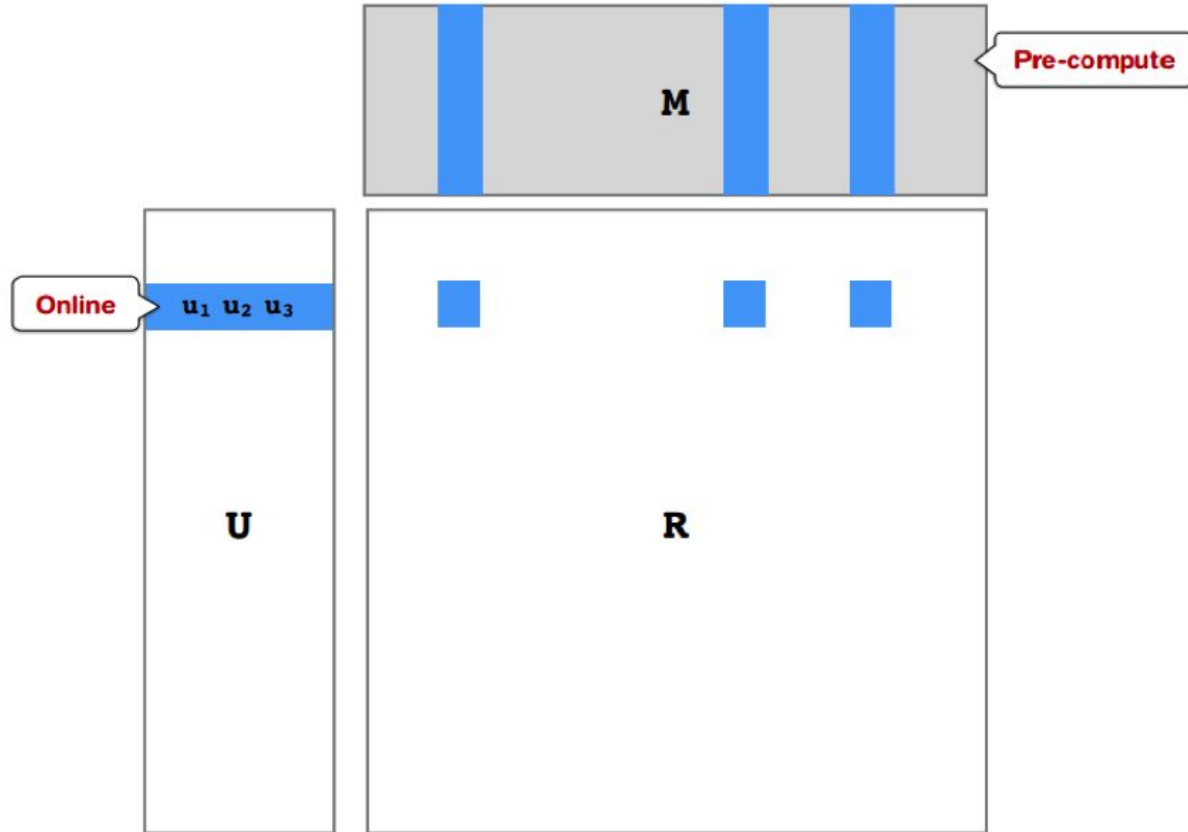
11. YOU DON'T NEED TO DISTRIBUTE YOUR  
RECSYS

# Distributing Recommender Systems

- Most of what people do in practice can fit into a multi-core machine
  - As long as you use:
    - Smart data sampling
    - Offline schemes
    - Efficient parallel code
- (... but not Deep ANNs)
- Do you care about costs? How about latencies or system complexity/debuggability?



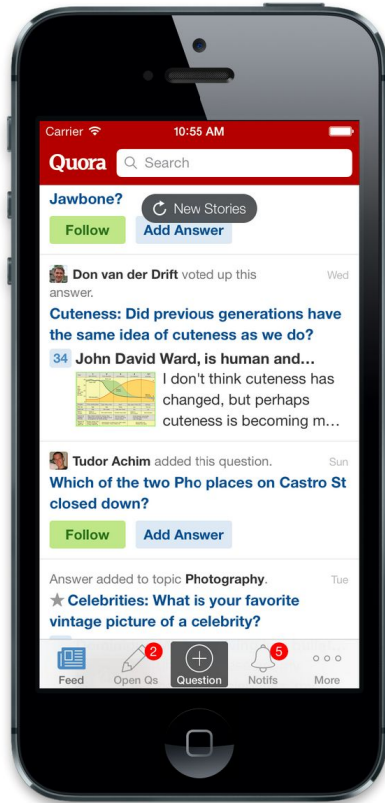
# Matrix Factorization Example



12. THE UI IS THE ONLY COMMUNICATION  
CHANNEL WITH WHAT MATTERS THE MOST:

USERS

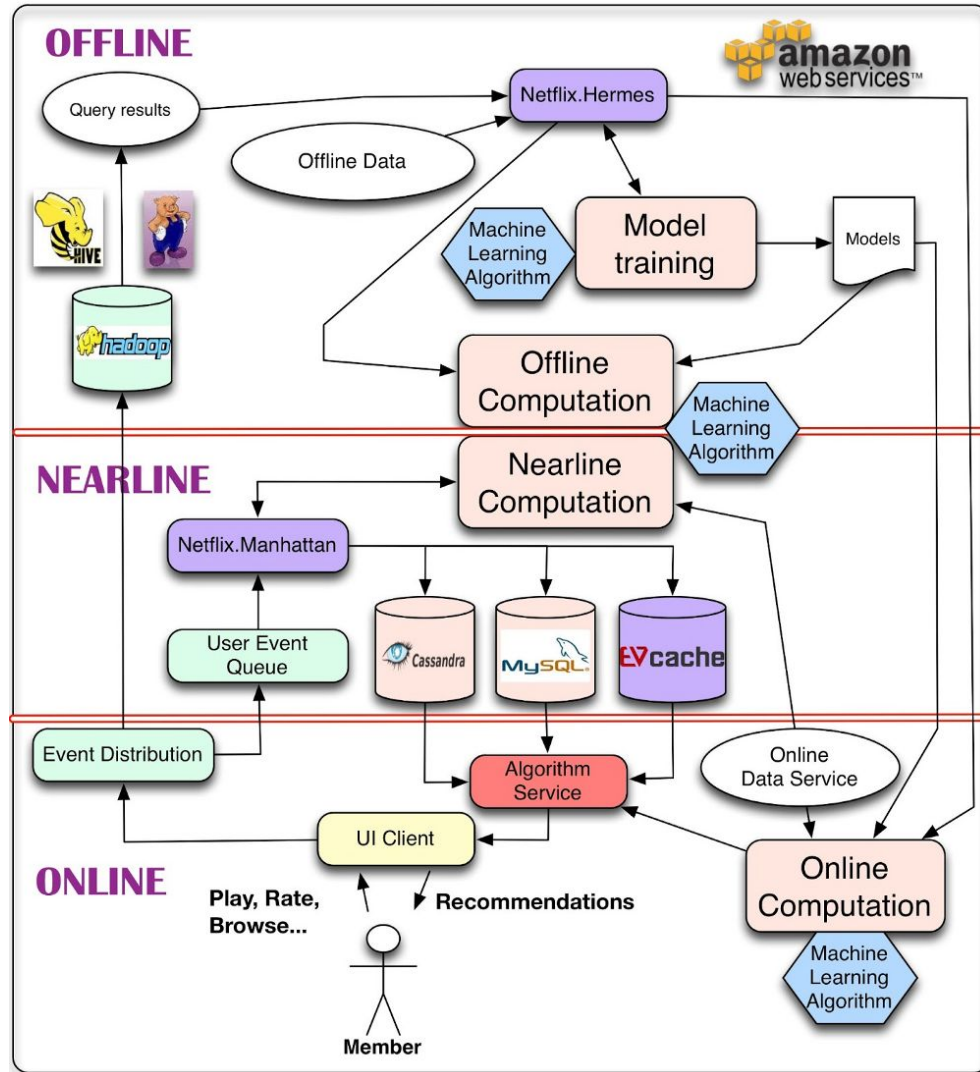
# UI->Algorithm->UI



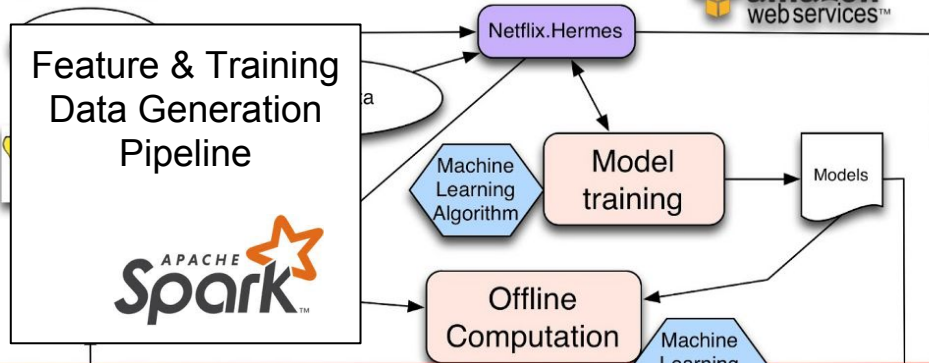
- The UI generates the user feedback that we will input into the algorithms
- The UI is also where the results of our algorithms will be shown
- A change in the UI might require a change in algorithms and viceversa

# **5. A Recsys Architectural Blueprint**

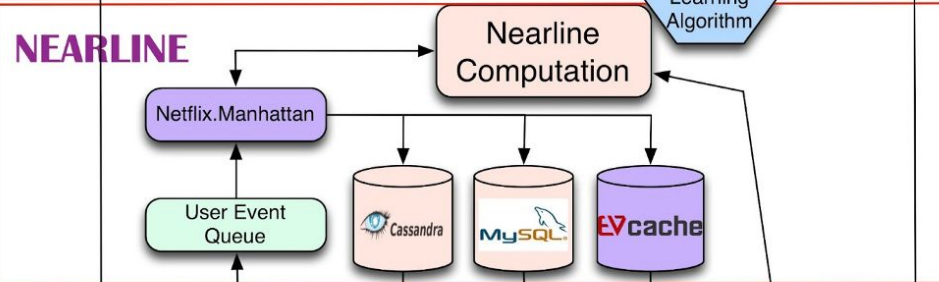




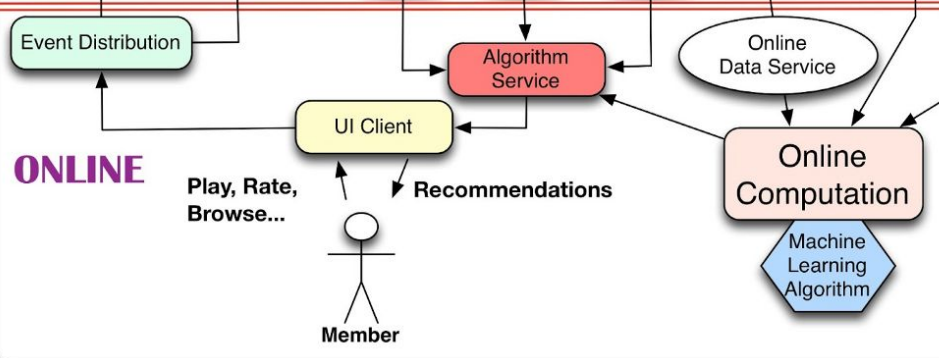
## OFFLINE



## NEARLINE

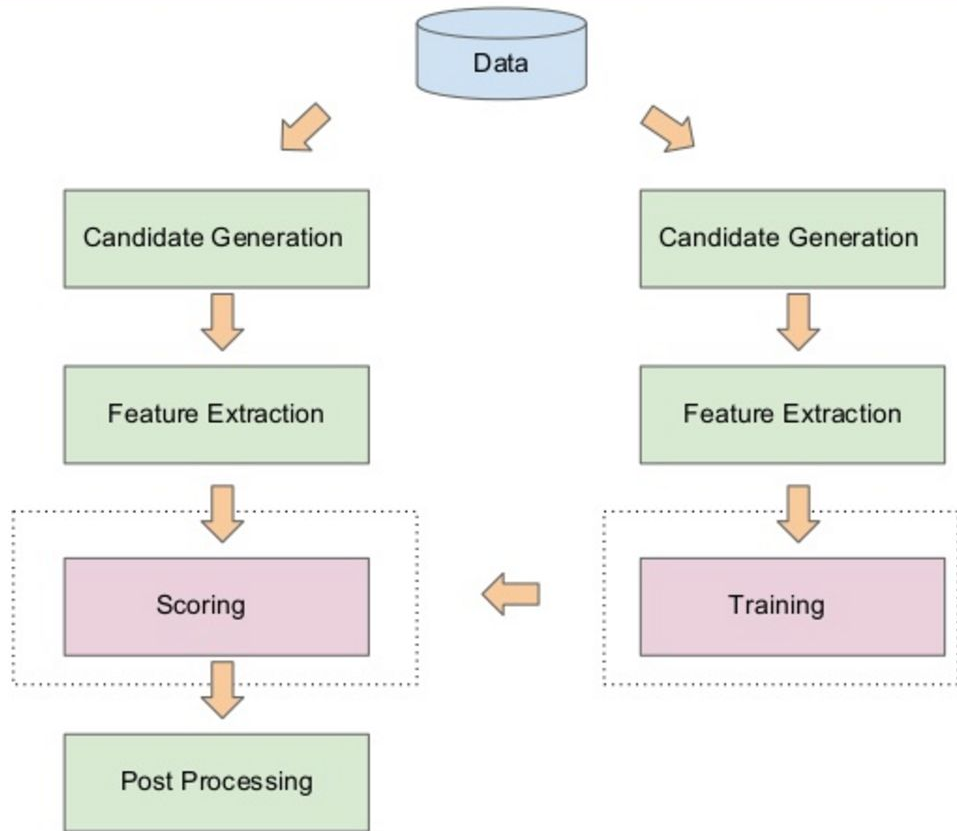


## ONLINE

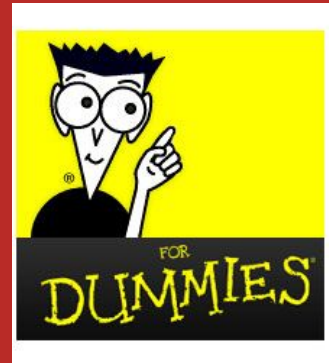




- We want the same code/systems/tools to work for both experimentation & production.
- But we need to carefully “control” the production code to keep it be fast.
- So need to “control” offline experimentation systems too.



# 6. Building a state-of-the-art Recsys



# 6.1 Training, testing, and metrics

## Training, testing, metrics

---

- As mentioned in the lessons, this is essential
- Choose implicit data and metrics that connect to your business goal
- Sample negatives smartly
- Select validation and test set carefully (e.g. avoid time traveling)

## Training, testing, metrics

---

- For metrics, prefer ranking or ranking-related metrics

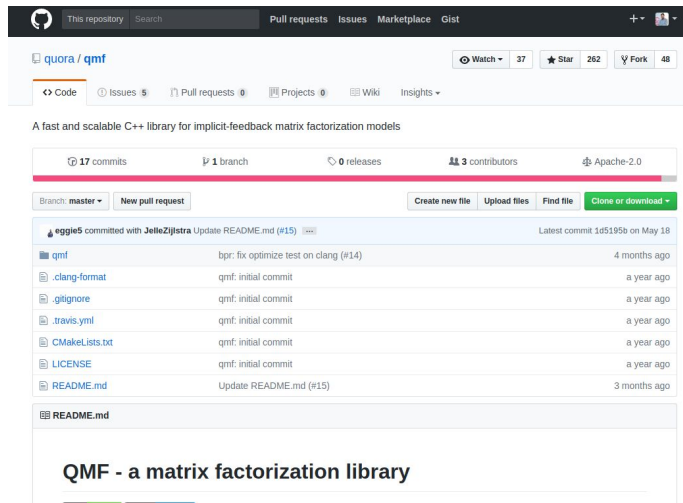
## 6.2 Implicit Matrix Factorization

# Implicit Matrix Factorization

- Experience says, best single (simple) approach:  
implicit matrix factorization:
  - ALS. Alternating Least Squares (Hu et al. 2008)
  - BPR. Bayesian Personalized Ranking (Rendle et al. 2009)

# Recommended Implementations

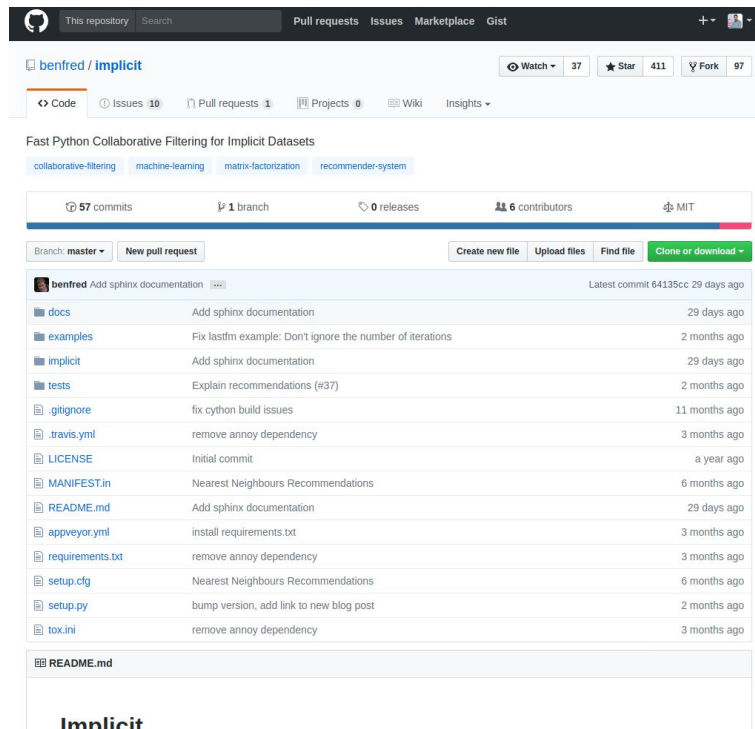
- Quora's QMF
  - Efficient compiled C++ code
  - Supports many evaluation metrics





# Recommended Implementations

- Implicit
  - Efficient
  - Python
  - Well-maintained



The screenshot shows the GitHub repository page for **benfred / Implicit**. The repository is described as "Fast Python Collaborative Filtering for Implicit Datasets" and includes tags for `collaborative-filtering`, `machine-learning`, `matrix-factorization`, and `recommender-system`. It has 57 commits, 1 branch, 0 releases, 6 contributors, and 411 stars. The file list shows various documentation and configuration files, with the latest commit being "Add sphinx documentation" 29 days ago. The README section is partially visible at the bottom.

File	Commit Message	Time Ago
docs	Add sphinx documentation	29 days ago
examples	Fix lastfm example: Don't ignore the number of iterations	2 months ago
implicit	Add sphinx documentation	29 days ago
tests	Explain recommendations (#37)	2 months ago
.gitignore	fix cython build issues	11 months ago
.travis.yml	remove annoy dependency	3 months ago
LICENSE	Initial commit	a year ago
MANIFEST.in	Nearest Neighbours Recommendations	6 months ago
README.md	Add sphinx documentation	29 days ago
appveyor.yml	install requirements.txt	3 months ago
requirements.txt	remove annoy dependency	3 months ago
setup.cfg	Nearest Neighbours Recommendations	6 months ago
setup.py	bump version, add link to new blog post	2 months ago
tox.ini	remove annoy dependency	3 months ago

**Implicit**

## Others?

---

- Sorry to say, but I cannot recommend any others (no, not Mahout)

## 6.3 A/B Test

# AB Test

---

- So, you have your first implementation
  - Have tuned hyperparameters to optimize offline metric
  - How do you know this is working?
- Run AB Test!
  - Make sure offline metric (somewhat) correlates to online effect

# AB Test

---

- Ideally, you would run several AB tests with different offline metrics and data sampling strategies

## 6.4 Ensemble

# Ensemble

---

- Now, it's time to turn the model into a signal
- Brainstorm about some simple potential features that you could combine with implicit MF
  - E.g. user tenure, average rating for the item, price of the item...
- Add to MF through an ensemble

# Ensemble

- What model to use at the ensemble layer?
  - Always favor most simple -> L2-regularized Logistic Regression
  - Eventually introduce models that can benefit from non-linear effects and many features -> Gradient Boosted Decision Trees
  - Explore Learning-to-rank models -> LambdaRank



# 6.5 Iterate, Feature Engineering

# Iterate

---

- Experiment/add more features
- Experiment with more complex models
- Do both things in parallel
- Continue AB testing

# 7. Practical exercise

## Exercise

---

- Train an ALS implicit matrix factorization recommender system
- Do basic feature engineering to add other features
- Add the mix to an XGBoost-based ensemble
- This is very close to what you could be using in real-life (minus scalability/performance issues)

# **8. Future Research Directions**

# Many interesting future directions

1. Indirect feedback
2. Value-awareness
3. Full-page optimization
4. Personalizing the how

## Others

- Intent/session awareness
- Interactive recommendations
- Context awareness
- Deep learning for recommendations
- Conversational interfaces/bots for recommendations
- ...

# Indirect Feedback

## Challenges

User can only click on what you show

But, what you show is the result of what your model predicted is good

No counterfactuals

Implicit data has no real “negatives”

## Potential solutions

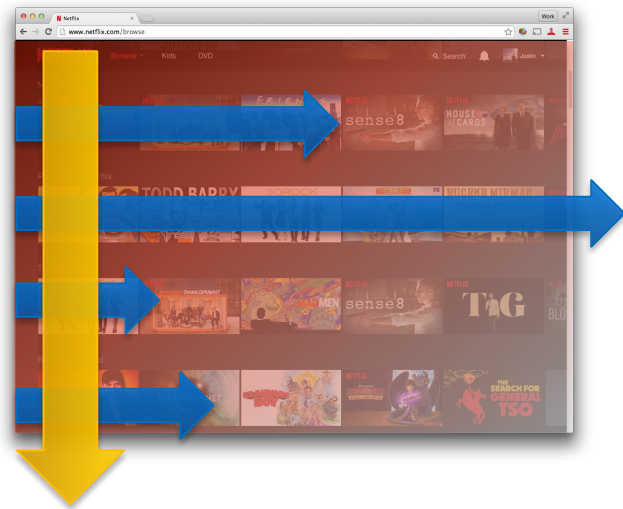
Attention models

Context is also indirect/implicit feedback

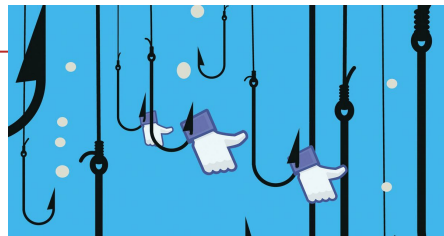
Explore/exploit approaches and learning across time

...

## Quora



# Value-aware recommendations

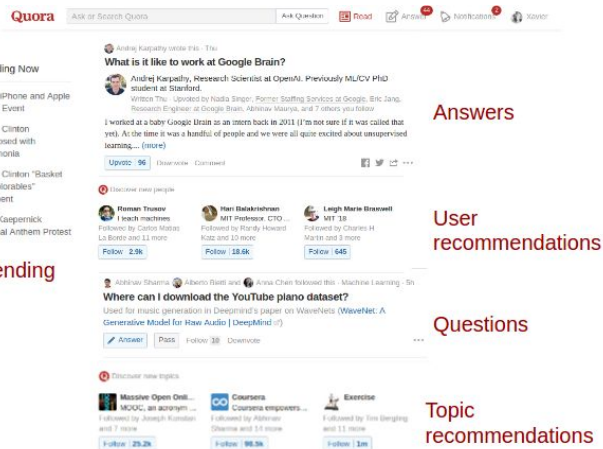
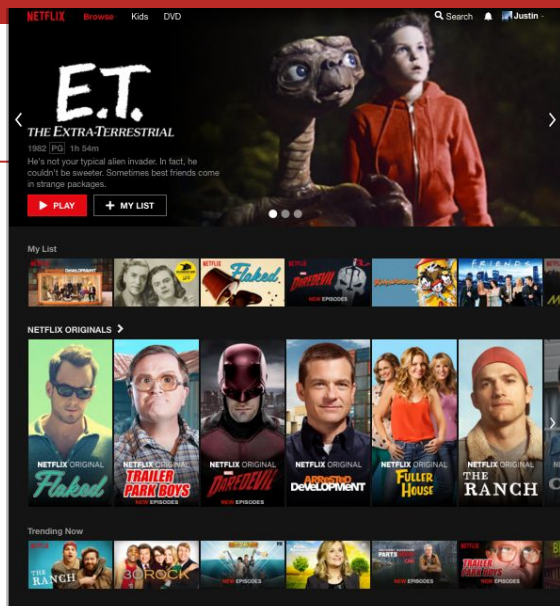


- Recsys optimize for probability of action
- Not all clicks/actions have the same “reward”
  - Different margin in ecommerce
  - Different “quality” of content
  - Long-term retention vs. short-term clicks (clickbait)
  - ...
- In Quora, the value of showing a story to a user is approximated by weighted sum of actions:
  - $v = \sum_a v_a 1\{y_a = 1\}$
- Extreme application of value-aware recommendations: suggest items to **create** that have the highest value
  - Netflix: Which shows to produce or license
  - Quora: Answers and questions that are not in the service



# Full-page optimization

- Recommendations are rarely displayed in isolation
  - Rankings are combined with many other elements to make a page
- Want to optimize the **whole page**
- **Jointly** solving for set of items and their placement
- While incorporating
  - Diversity, freshness, exploration
  - Depth and coverage of the item set
  - Non-recommendation elements (navigation, editorial, etc.)
- Needs work hand-in-hand with the UX



# Personalizing how we recommend (not just what)

- **Algorithm level:** Ideal balance of diversity, novelty, popularity, freshness, etc. may depend on the person
- **Display level:** How you present items or explain recommendations can also be personalized
  - Select the **best information** and presentation for a user **to quickly decide** whether or not they want an item
- **Interaction level:** Balancing the needs of lean-back users and power users

# Example rows and beyond

Predicted  
rating

Metadata

Hero  
Image

Synopsis

Evidence

Row Title

ROWS

## House of Cards

★★★★★ 2013-2015 TV-MA 3 Seasons 5.1

A ruthless politician will stop at nothing to conquer Washington, D.C., in this Emmy and Golden Globe-winning political drama.

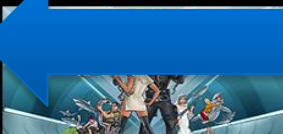


Based on your interest in:  
Marco Polo

### Popular on Netflix



### Recently Watched



Ranking

Horizontal  
Image

# 9. Conclusions

# Conclusions

- Recommendation is about much more than just predicting a rating
- All forms of recommendation require of a tight connection with the UI
  - Capture the right kind of feedback
    - Explicit/implicit feedback
    - Correct for presentation bias
    - ...
  - Present the recommendations correctly
    - Explanations
    - Diversity
    - Exploration/Exploitation
    - ....

# Conclusions

---

- For the algorithm:
  - Use implicit feedback if possible
  - Build a Matrix Factorization recommender system
  - Think of using ensembles and turning your problem into a feature engineering problem
  - Always think of the metric you are optimizing to and the data you are using
- Whatever you do in the lab, you should trust your AB tests

# 10. References

## Other resources

- 4 hour video of my lecture at MLSS at CMU (Youtube)
- *“Recommender systems in industry: A netflix case study”* (X. Amatriain, J. Basilico) in Recommender System Handbook
- *“Past, Present, and Future of Recommender Systems: An Industry Perspective”* (X. Amatriain, J. Basilico. Recsys 2016)
- *“Mining large streams of user data for personalized recommendations”* (X. Amatriain) - ACM SigKDD Explorations Newsletter
- *“Big & personal: data and models behind netflix recommendations”* (X. Amatriain) - ACM Workshop on Big Data
- Visit my slideshare page: <https://www.slideshare.net/xamat>