Improving Matrix Factorization 10 years on

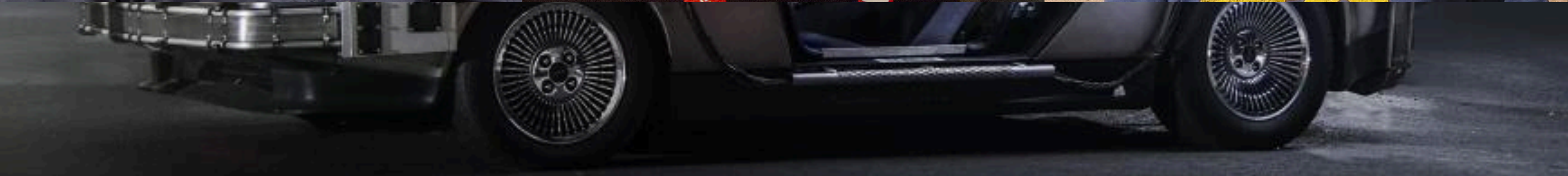
Markus Weimer, Alexandros Karatzoglu, Alex Smola

Microsoft, Telefonica, Amazon AWS

alex@smola.org, @smolix

Back to 2008 …



Back to 2008 …

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

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update

Leaderboard

These guys rock!

Showing Test Score. Click here to show qui

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

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update

Leaderboard

All of these guys rock!

Showing Test Score. Click here to show qui

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

The Netflix Prize

Scary

- **Lots** of smart teams working on this
- Lots of smart teams working on this **full time**
- Lots of compute power (or so we thought)
- Winning required **lots of engineering** besides science

Good

- Everyone was minimizing $\sum_{(u,m)} (f(u,m) - r_{um})^2$
- If you can't win, why not change the game?

Recommender Systems vs. RMSE

- User cares about the **recommended items**
  - **High scores matter** (we don’t care in detail about the bottom)
  - **Relative order matters**
- **This is a personalized ranking problem**
  just like web search, product recommendation, news, social media, advertising, friend recommendation, papers …

Recommender Systems vs. RMSE

- User cares about the **recommended items**
  - **High scores matter** (we don't care in detail about the bottom)
  - **Relative order matters**
- **This is a personalized ranking problem**
  just like web search, product recommendation, news, social media, advertising, friend recommendation, papers …

Recommender Systems vs. RMSE

- User cares about the **recommended items**
  - **High scores matter** (we don’t care in detail about the bottom)
  - **Relative order matters**
- **This is a personalized ranking problem**
  just like web search, product recommendation, news, social media, advertising, friend recommendation, papers …

Recommender Systems vs. RMSE

- **RMSE**

$$\sum_{(u,m)} \left(f(u,m) - r_{um}\right)^2$$

- **Ordinal Regression (Herbrich et al., 1999)**
  Require that large margin condition between ratings
  $$f(u,m) \geq f(u,m') + 1 \text{ if } r_{um} > r_{um'}$$
  Replace with soft-margin loss function
  $$l(u,m,m') = \max(0, 1 - f(u,m) + f(u,m'))$$
- **Optimize this for recommender systems**
- **Sort to reduce cost from $O(n^2)$ to $O(n \log n)$**

Recommender Systems vs. RMSE

- **Optimization**
  No longer separable in (users, movies)
- **2008**
  - Bundle method for each pass over movies
  - Add specific biases for users / movies
  - User-specific regularization and graph regularization
- **2018**
  - Decouple terms with constraint (= isotonic regression)
  - Also works well for Siamese Network
    (triplet to pairwise conversion - Wu et al., 2017)
  - Stochastic Gradient Descent
    (and stop worrying about convexity)

Back to 2018 …

What changed?

- **Function Classes**
  - Deep learning for more powerful function classes
  - Inference frameworks (TF, MxNet Gluon, PyTorch …)
- **Time**
  - For modeling
  - State updates
- **Loss functions**
  - Replace with user activity models (click, skip, diversity)
  - Dueling bandits
  - IR GAN loss
  - Bimodality
- **Cold Start**
  - Features, Networks
- **Reinforcement Learning**

recommendation =
search & ranking =
advertising =
social networks =
news

Function Classes

Inner products and beyond

Inner Products

$$f(u,m) = \langle v_u, v_m \rangle$$

Factorization Machines (Rendle et al., 2012 …)

$$f(u,m) = \langle w, [z_u, z_m] \rangle + z_u^\top U U^\top z_m$$

$$\text{where } z_u = [e_u, x_u] \text{ and } z_m = [e_m, x_m]$$

Sparse Vectors - ACCAMS (Beutel et al., 2016)

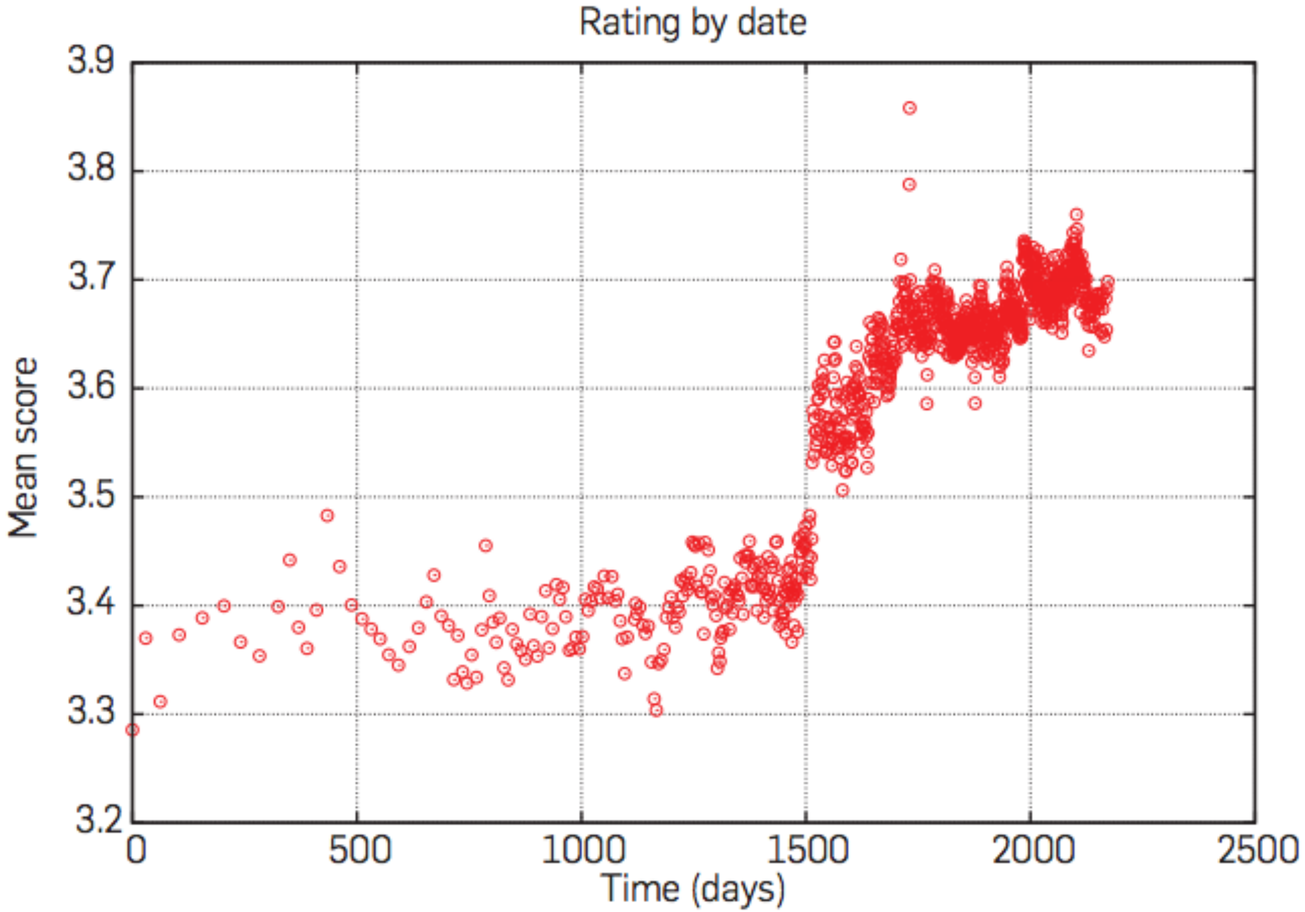
$$f(u,m) = \sum_i M[i, v_{ui}, v_{mi}]$$

Deep Networks

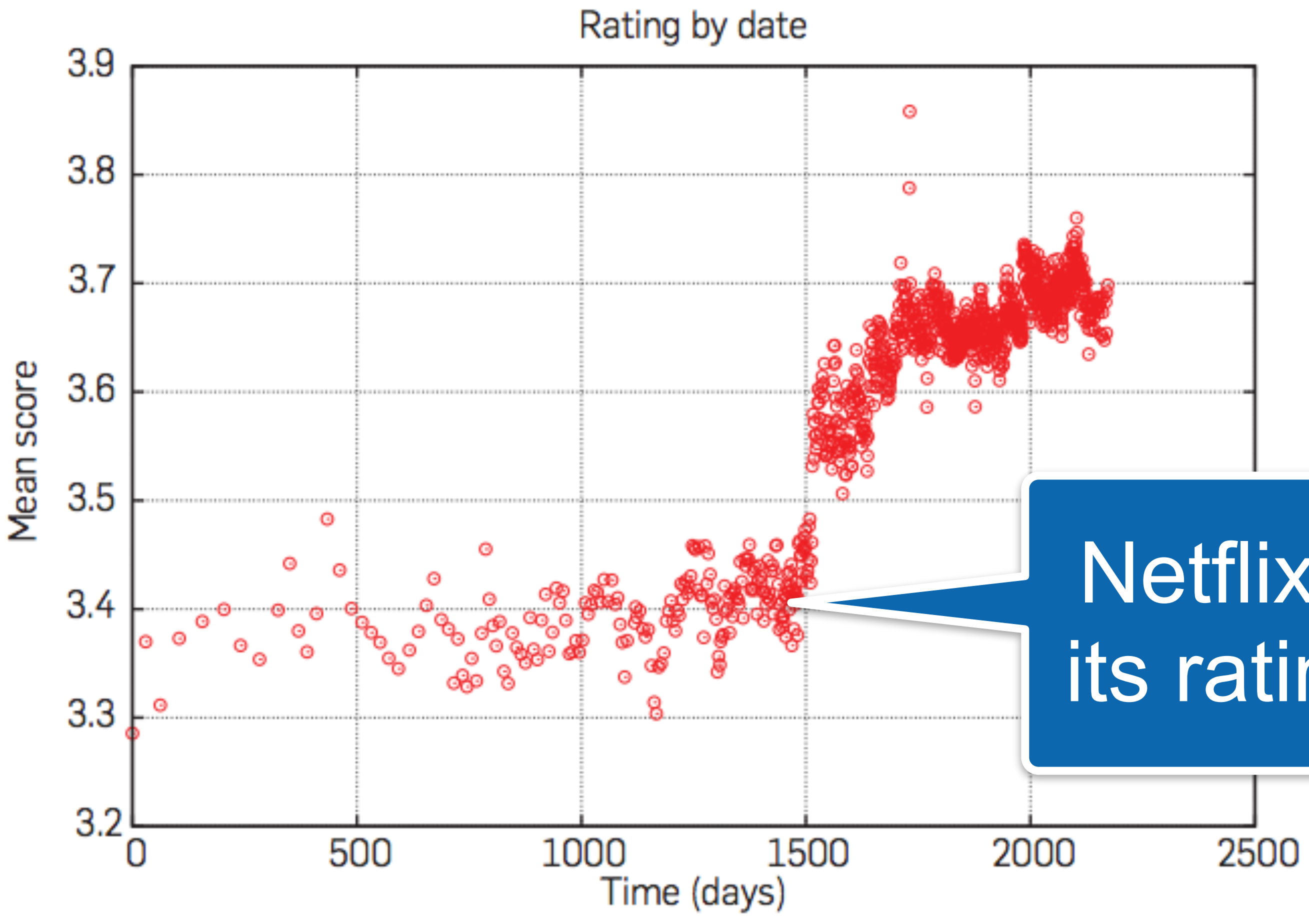
$$f(u,m) = \text{Network}(e_u, x_u, e_m, x_m)$$

Time

Time matters (Koren, 2009)

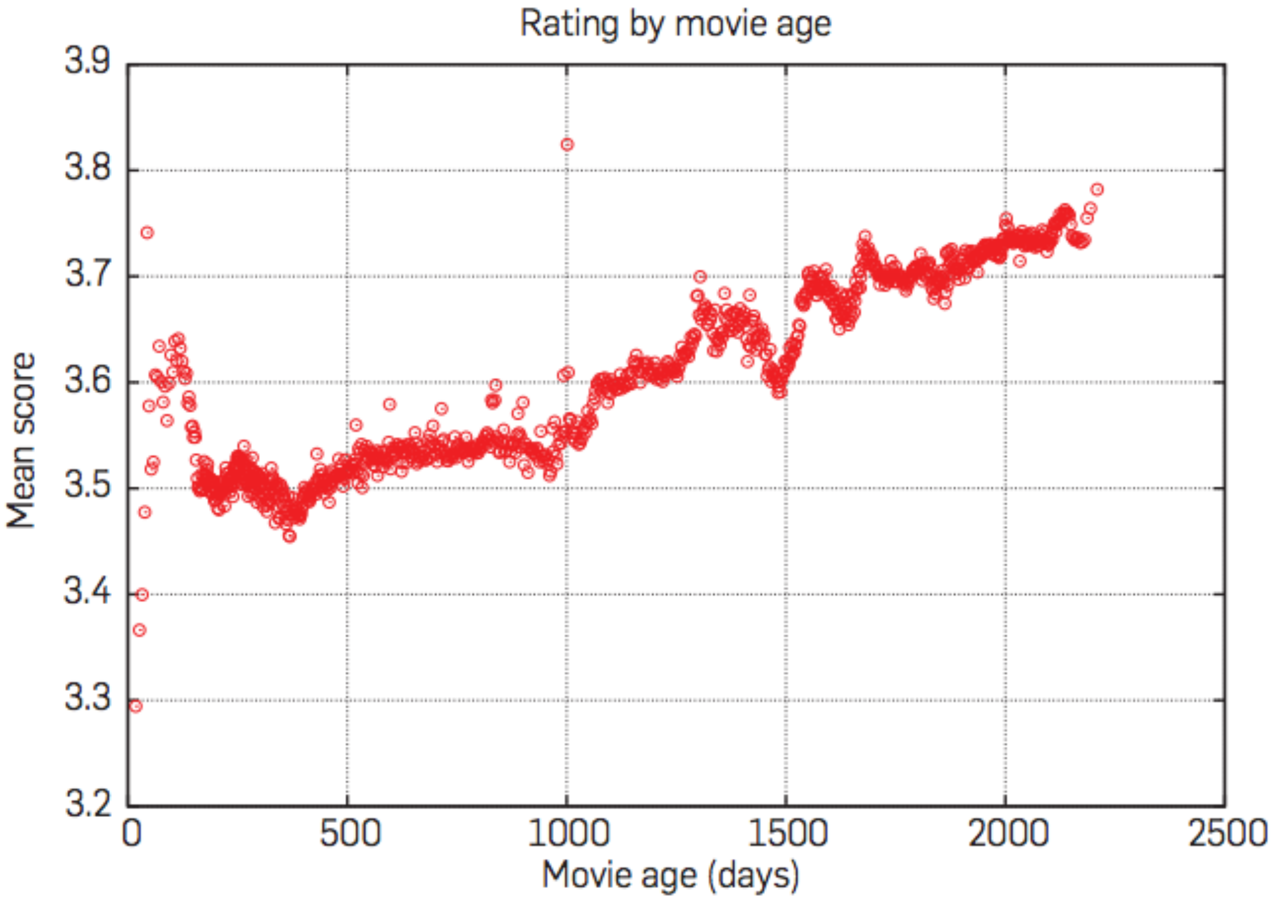


Time matters (Koren, 2009)

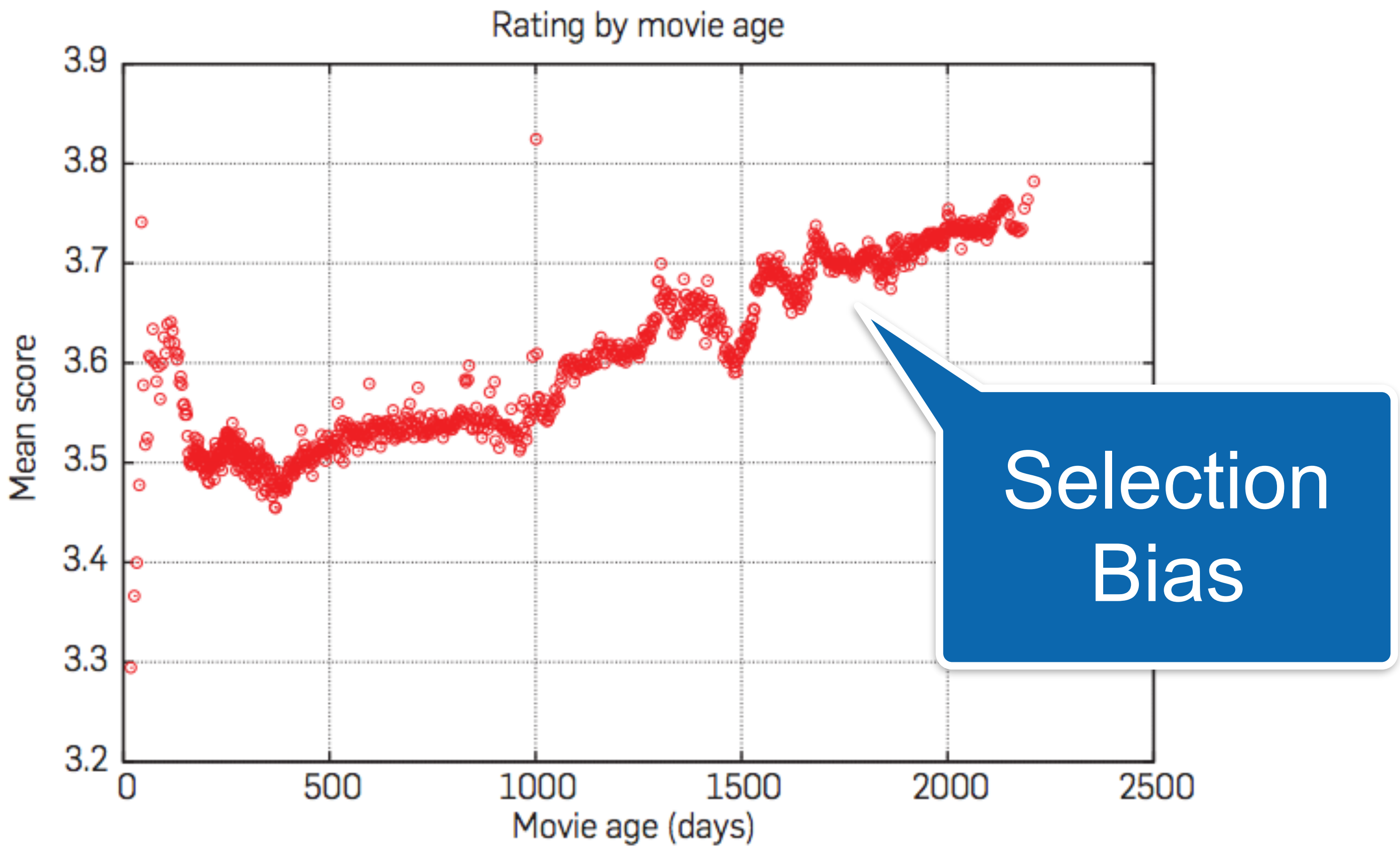


Netflix changed its rating system

Time matters (Koren, 2009)

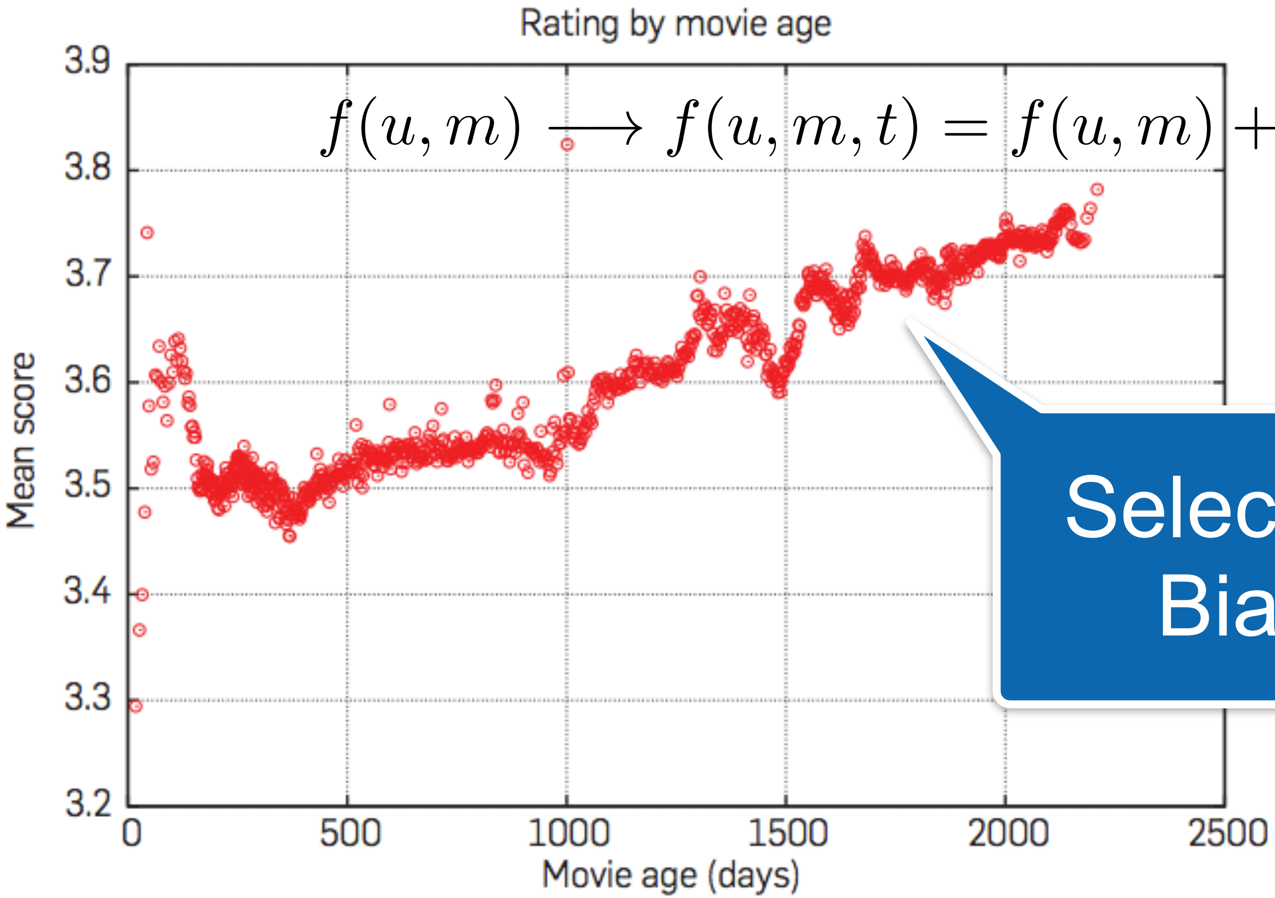


Time matters (Koren, 2009)



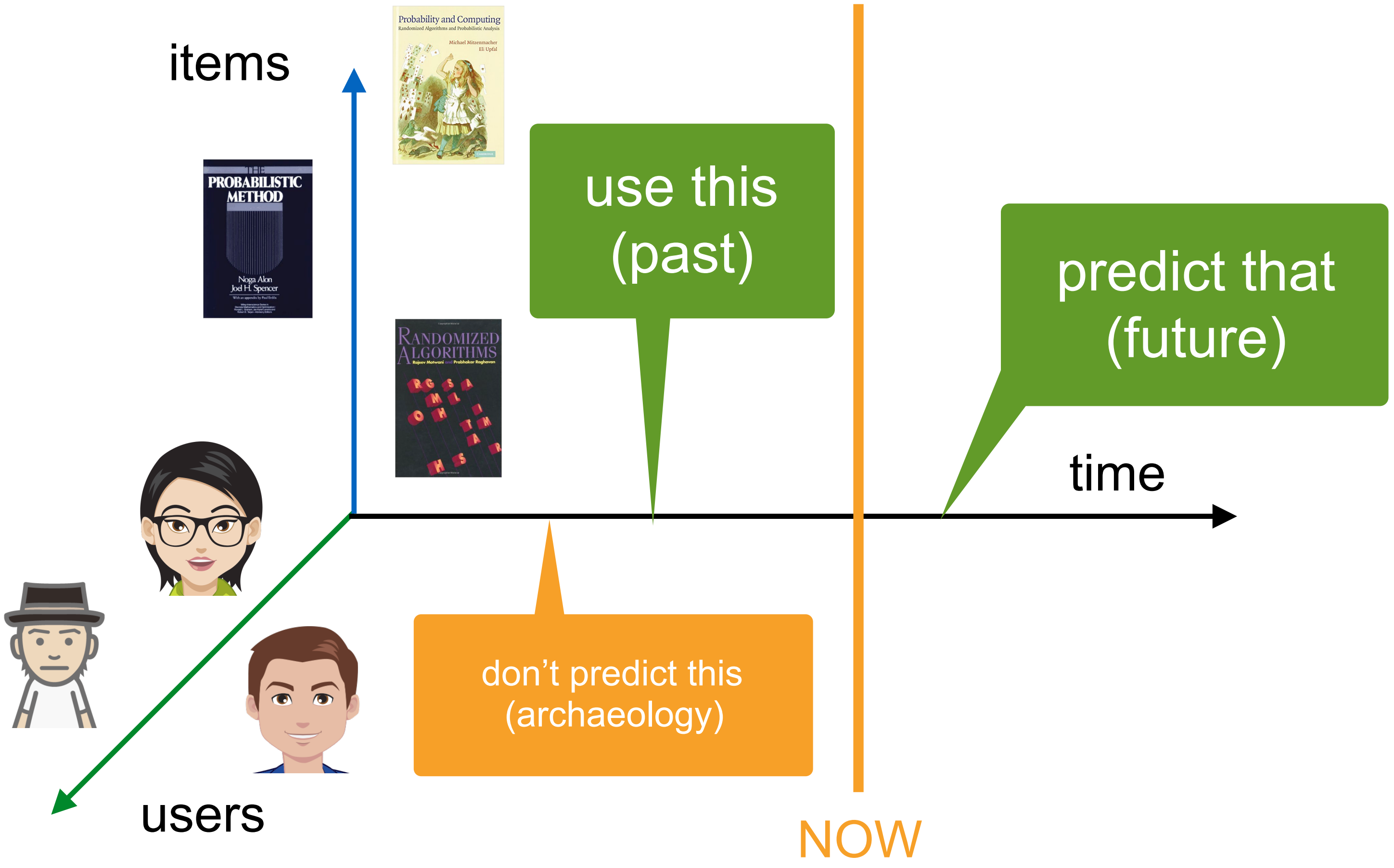
Time matters (Koren, 2009)

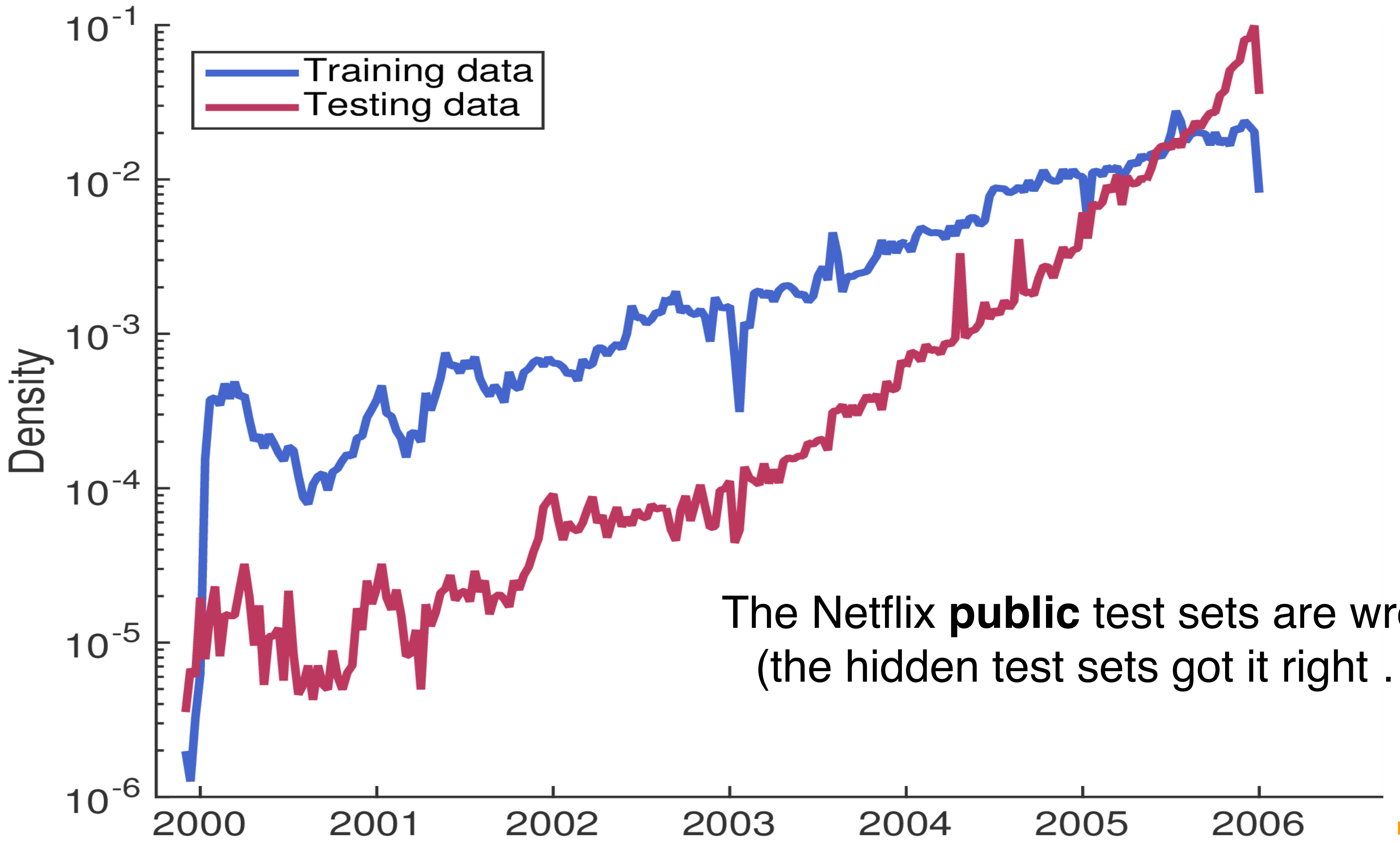
$$f(u,m) \longrightarrow f(u,m,t) = f(u,m) + (w_m + w_u)t$$



Selection Bias

Recommender systems, not recommender archaeology

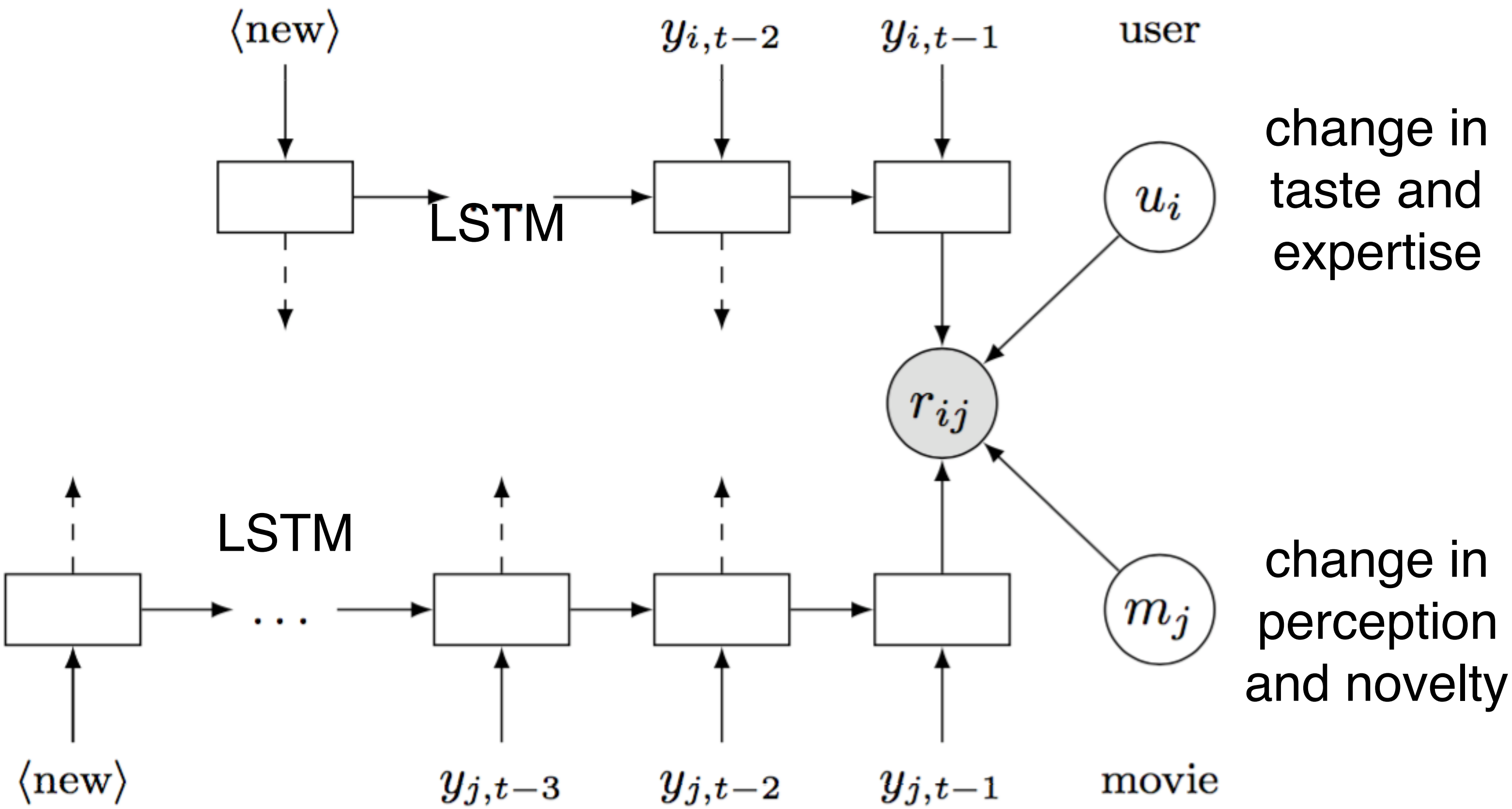




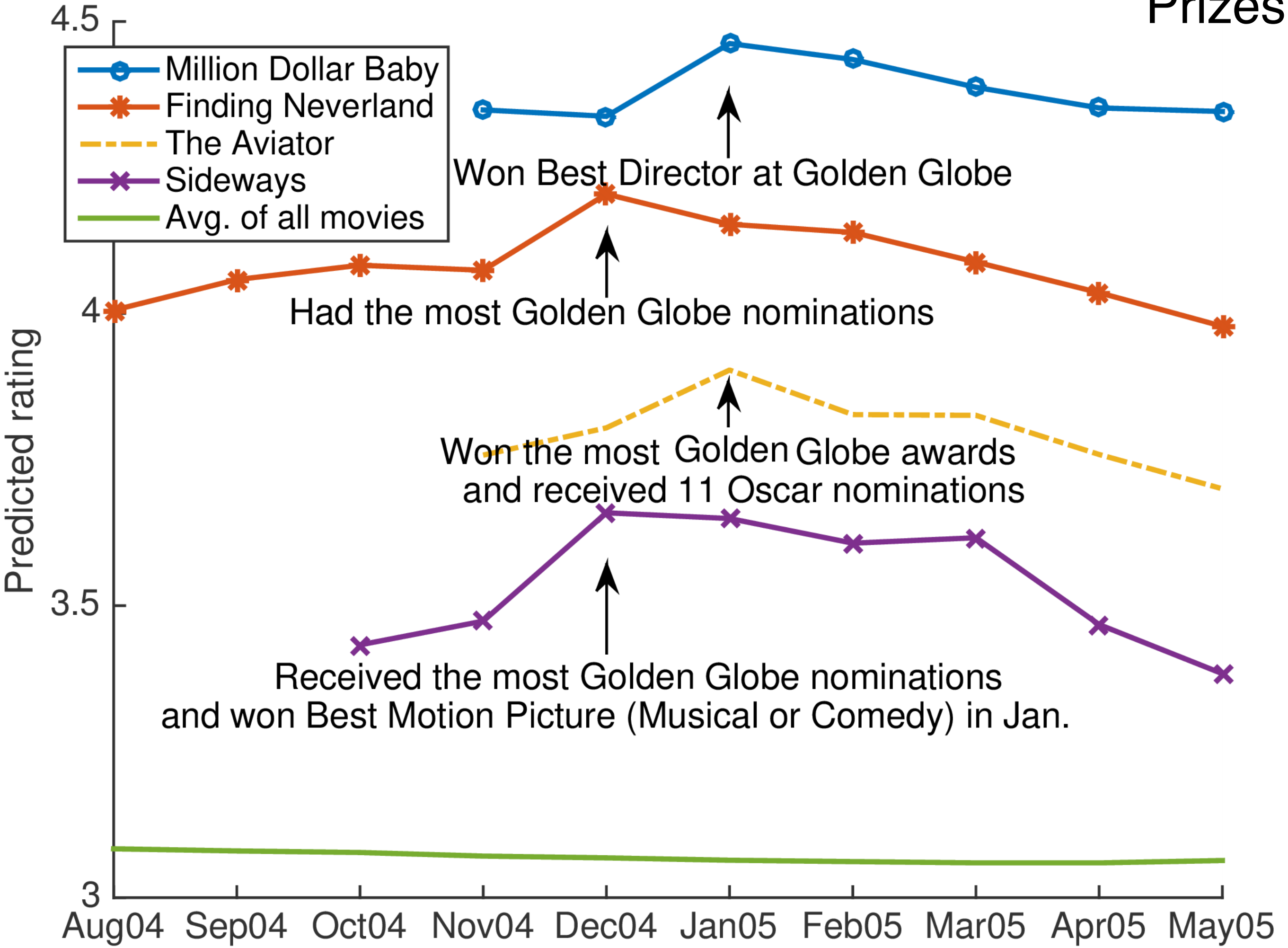
The Netflix **public** test sets are wrong
(the hidden test sets got it right …)

amazon web services

Recurrent Recommender Networks (Wu et al., 2017)



Prizes



amazon web services

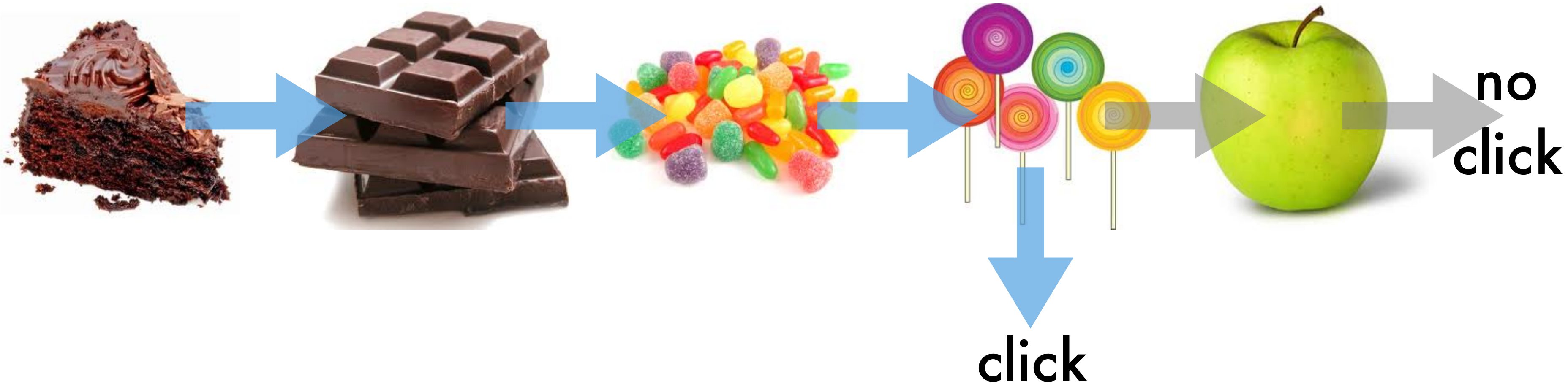
Objective Functions

Scores are not the answer

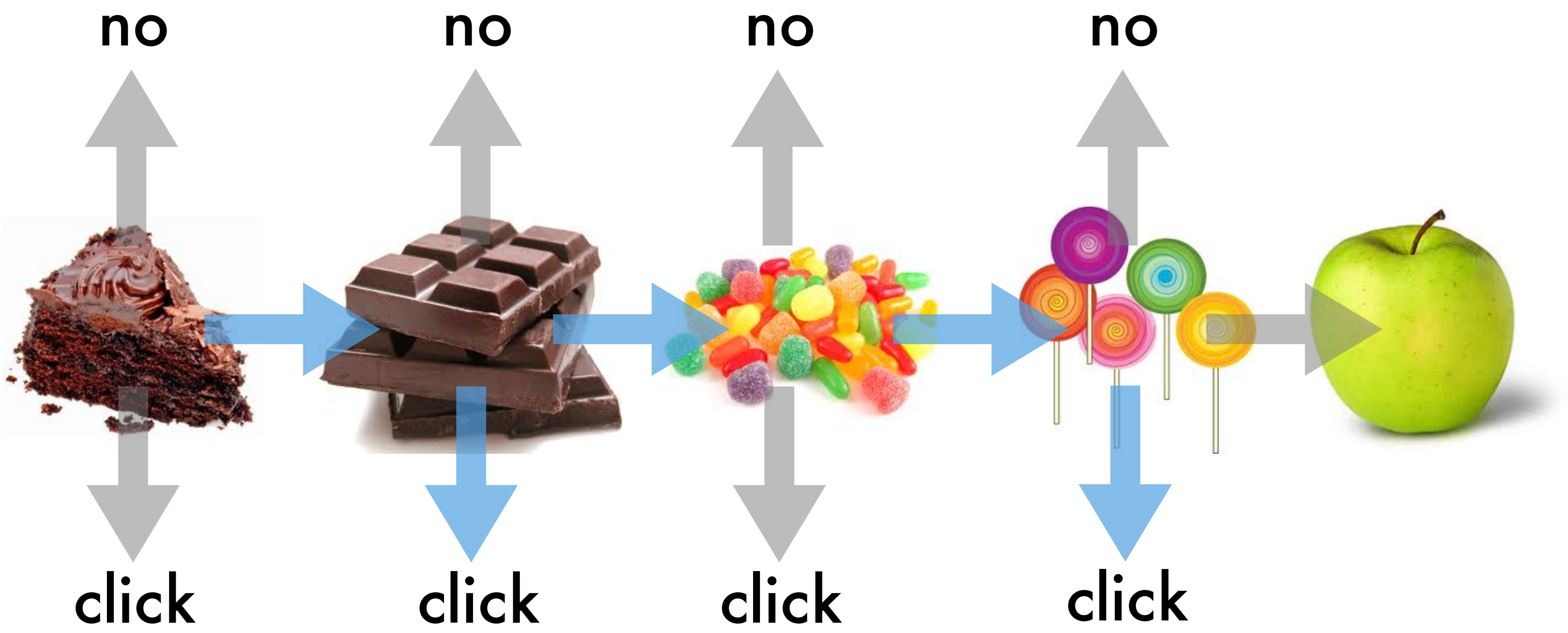
‘Die Hard’

‘Apple’ vs. ‘apple’

Solution - model viewing behavior

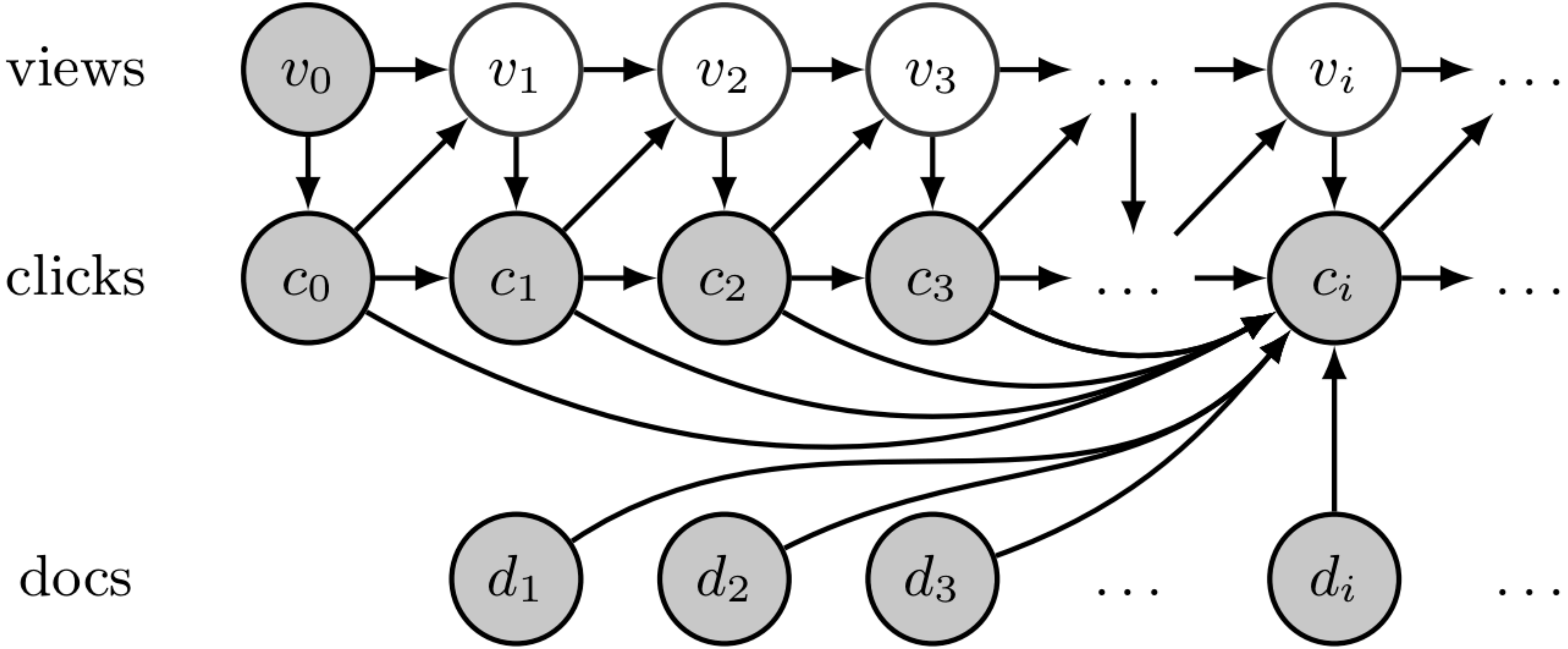


Solution - model viewing behavior



e.g. Chapelle et al, 2010

Solution - model viewing behavior



e.g. Ahmed et al., 2012

Likert Scale vs. Regression

Buy this Harddisk?

Likert Scale vs. Regression

Buy this Harddisk?

Really?

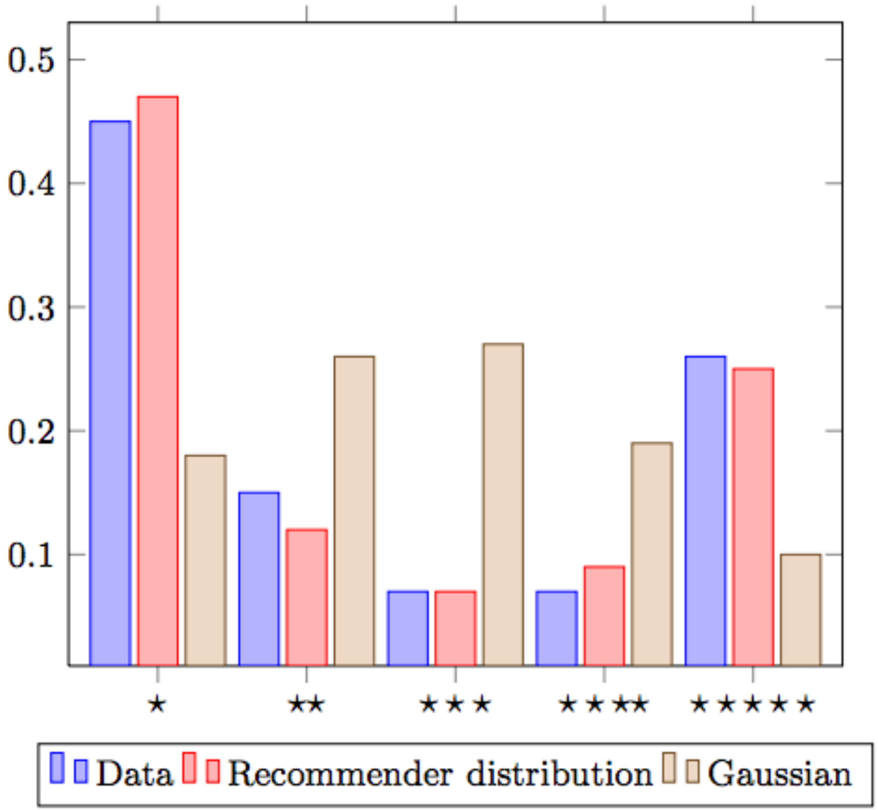
Likert Scale vs. Regression

Buy this Harddisk?

Really?

Restaurant for First Date?

Likert Scale vs. Regression (Beutel et al., 2014)

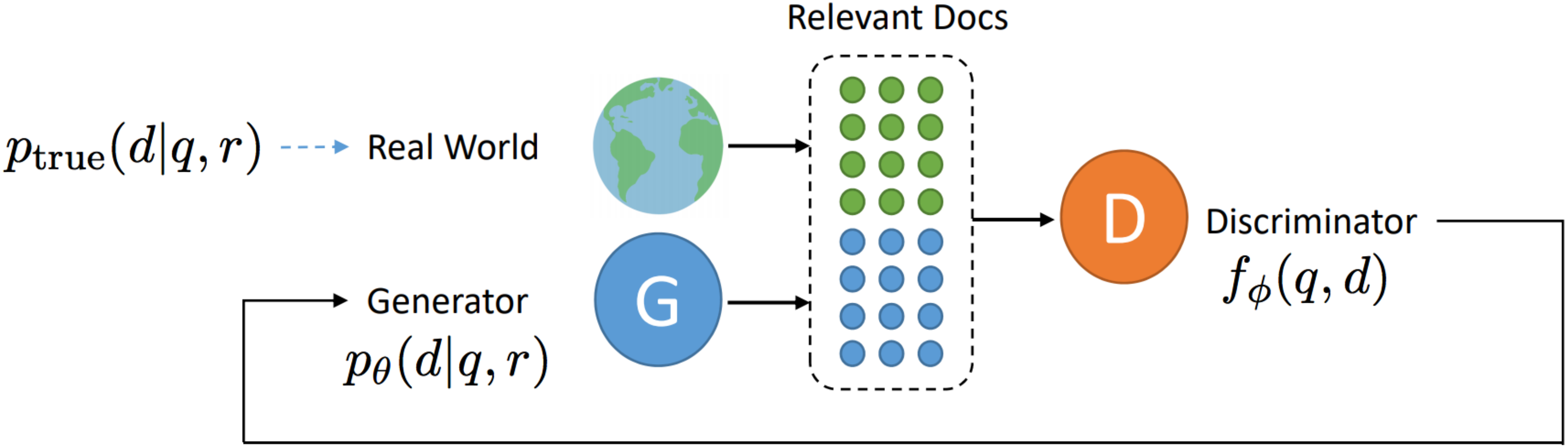


$$p(r) \propto \exp(\alpha r + \beta r^2)$$

- Quadratic exponential family model with discrete domain
- Convex optimization problem
- Fits data more accurately

IR-GAN: Throwing out the loss entirely (Zhang et al, 2017)

- Even for ranking loss function is a mess
- Use GAN to model scores directly
- Works **amazingly** well

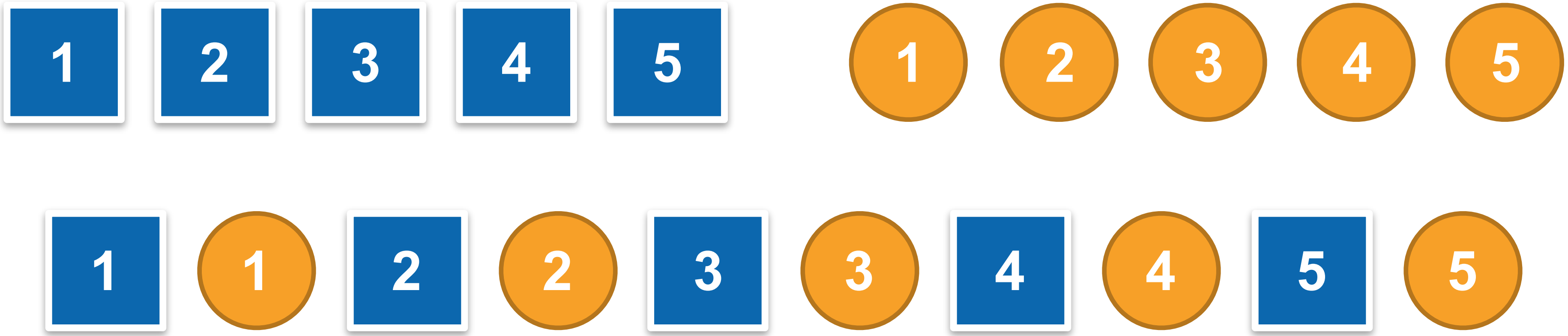


Dueling Bandits (Yue et al., 2009)

- Aim for weak gradient signal (is ranking A better than B?)
  - User models are a mess
  - Can we get this directly from A/B experiments

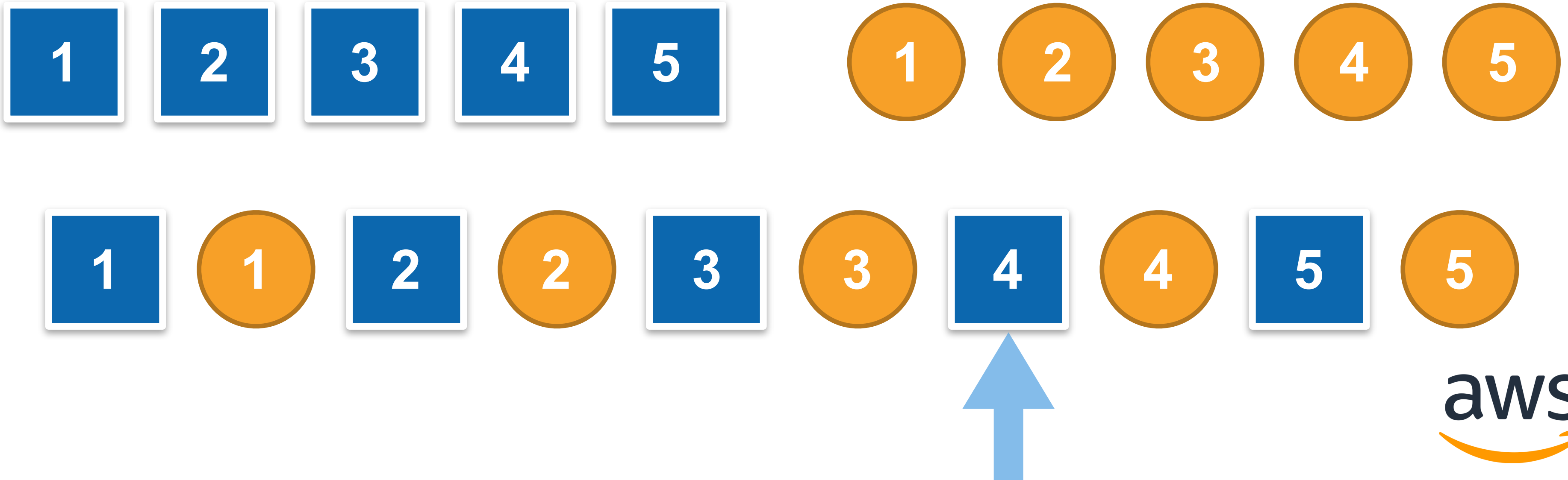
Dueling Bandits (Yue et al., 2009)

- Aim for weak gradient signal (is ranking A better than B?)
  - User models are a mess
  - Can we get this directly from A/B experiments



Dueling Bandits (Yue et al., 2009)

- Aim for weak gradient signal (is ranking A better than B?)
  - User models are a mess
  - Can we get this directly from A/B experiments



What changed?

- **Function Classes**
  - Deep learning for more powerful function classes
  - Inference frameworks (TF, MxNet Gluon, PyTorch …)
- **Time**
  - For modeling
  - State updates
- **Loss functions**
  - Replace with user activity models (click, skip, diversity)
  - Dueling bandits
  - IR GAN loss
  - Bimodality
- **Cold Start**
  - Features, Networks
- **Reinforcement Learning**

Crystal Ball (the next 10 years)

- **Toolkits**
  Deep Learning based & scalable & differentiable
- **Graphs**
- **Temporal Models**
- **Loss Functions**
  - Model Layout Directly
  - Model Interaction Directly
- **Reinforcement Learning**
  - Couple observations and model updates **live**
  - Generate recommended sequence

Improving Matrix Factorization 10 years on

Markus Weimer, Alexandros Karatzoglu, Alex Smola

Microsoft, Telefonica, Amazon AWS

And yes, we're all hiring …