CSCE 670 - Information Storage and Retrieval

Lecture 13: Recommender Systems (Matrix Factorization) and Quiz 2

Yu Zhang

yuzhang@tamu.edu

October 7, 2025

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html

Adapted from the slides by Prof. Jure Leskovec (Stanford)

Recap: Latent Factor Model (without Bias)

users (columns) × items (rows)

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

$\boldsymbol{U}$

$\approx$

factors (columns) × items (rows)

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

$\boldsymbol{Q}$

$\times$

users (columns) × factors (rows)

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

$\boldsymbol{P}^T$

$$U_{xi} = \sum_{\phi\text{: all factors}} Q_{i\phi} \cdot P_{x\phi}$$

Recap: Latent Factor Model (with Bias)

$$U_{xi} = \mu + b_x + b_i + \boldsymbol{q}_i \boldsymbol{p}_x^T$$

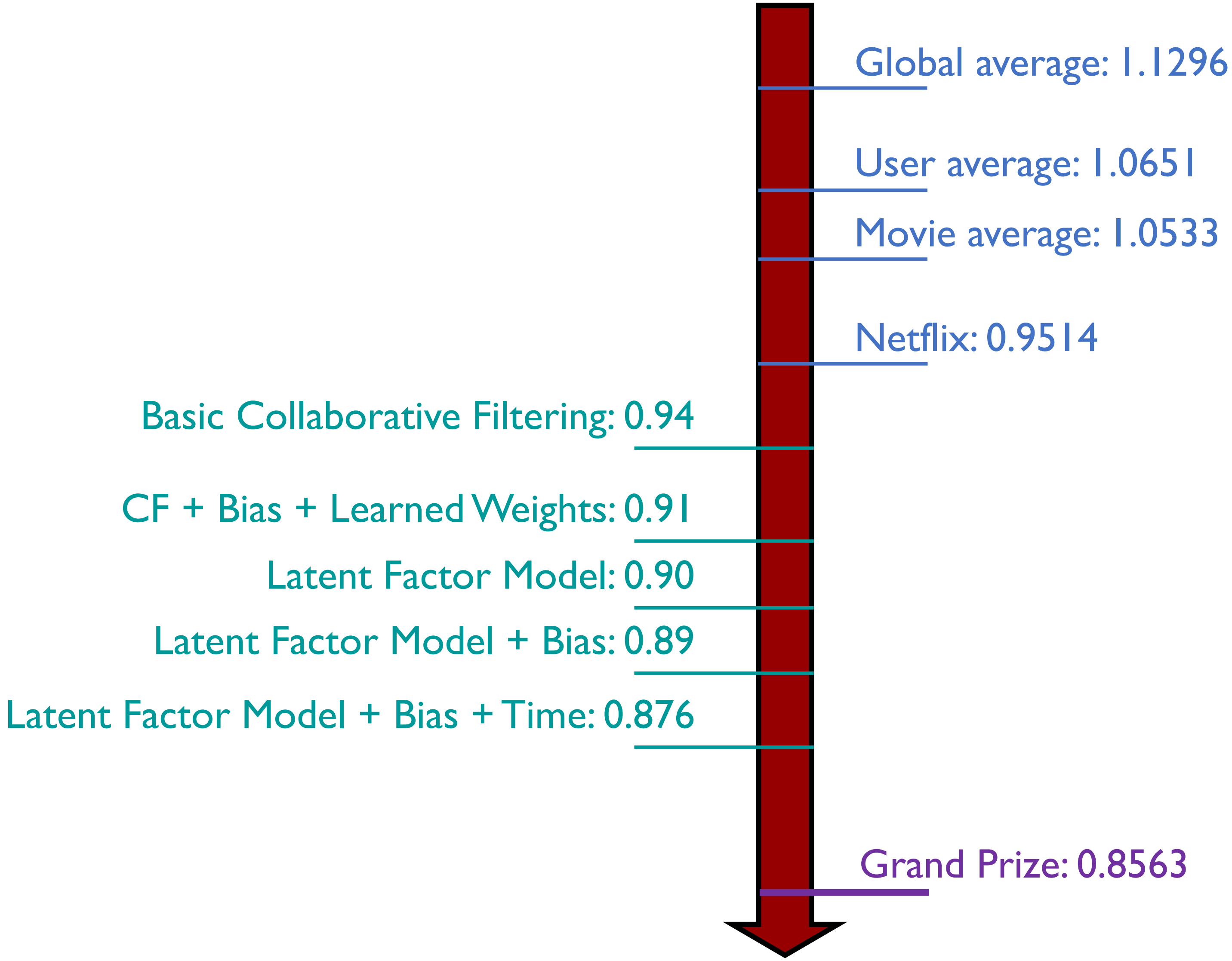
- μ: overall mean movie rating
  - E.g., $\mu = 2.7$
- b_x: rating deviation of user x (*to be learned*)
  - E.g., Bob is a critical reviewer. Based on the training data, his rating will be 0.7 star lower than the mean $\Rightarrow b_x = -0.7$.
- b_i: rating deviation of item i (*to be learned*)
  - E.g., Star Wars will get a mean rating of 0.5 higher than the average $\Rightarrow b_i = 0.5$
- $\boldsymbol{q}_i$ and $\boldsymbol{p}_x$: vector of user x and item i in the latent factor space (*to be learned*)
  - E.g., based on the genre, Bob likes Star Wars $\Rightarrow \boldsymbol{q}_i \boldsymbol{p}_x^T = 0.3$
- $U_{xi} = 2.7 - 0.7 + 0.5 + 0.3 = 2.8$

Recap: Latent Factor Model (with Bias)

$$\min_{\boldsymbol{Q},\boldsymbol{P},b_x,b_i} J = \sum_{(x,i)\text{ known}} (U_{xi} - (\mu + b_x + b_i + \boldsymbol{q}_i \boldsymbol{p}_x^T))^2$$
$$+ \left[c_1 \sum_x \|\boldsymbol{p}_x\|^2 + c_2 \sum_i \|\boldsymbol{q}_i\|^2 + c_3 \sum_x \|b_x\|^2 + c_4 \sum_i \|b_i\|^2 \right]$$

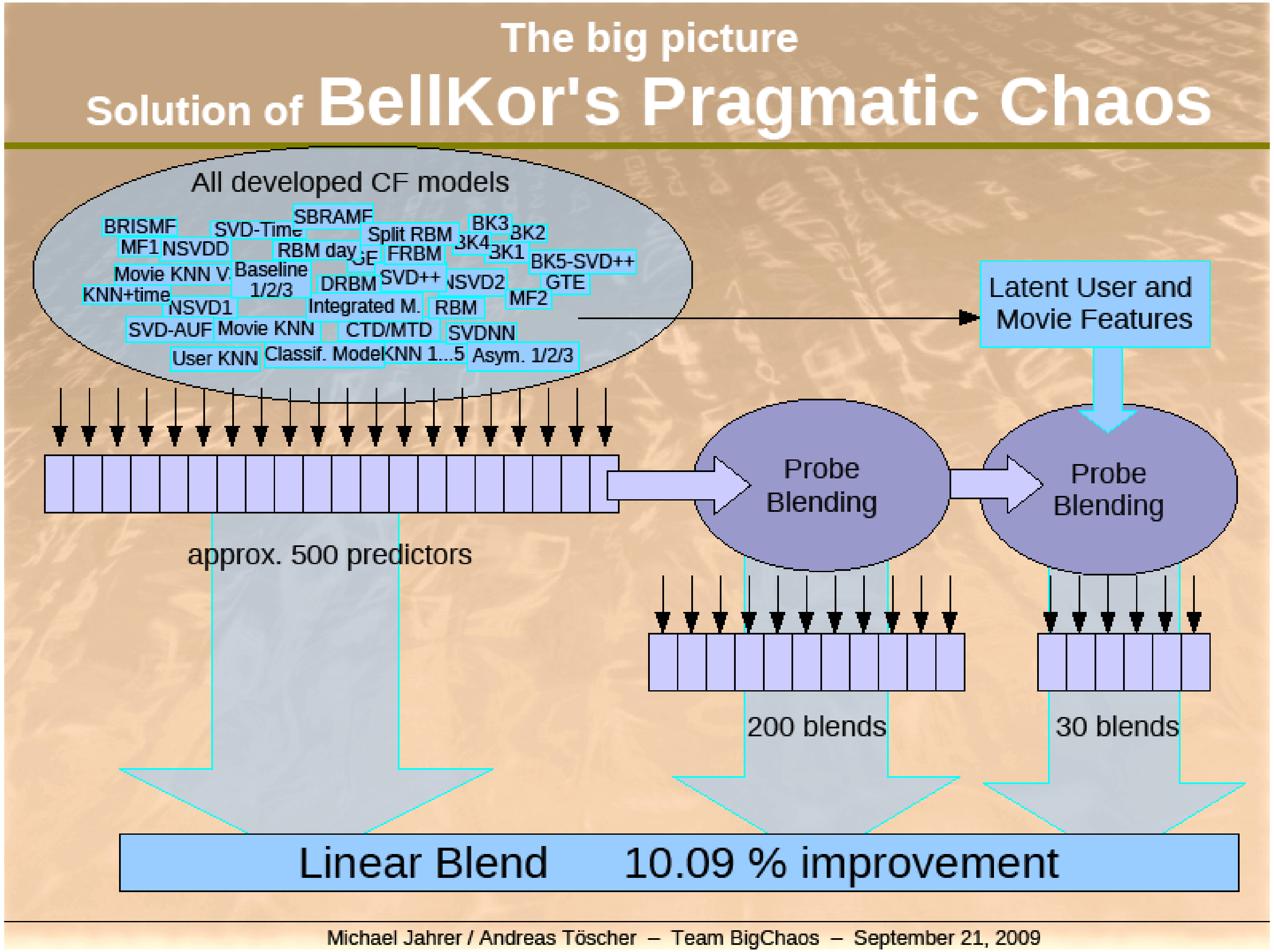
- Both biases b_x, b_i as well as interactions $\boldsymbol{q}_i, \boldsymbol{p}_x$ are treated as parameters to be learned via gradient descent
  - $P_{x\phi} = P_{x\phi} - \eta \frac{\partial J}{\partial P_{x\phi}}$, $\quad Q_{i\phi} = Q_{i\phi} - \eta \frac{\partial J}{\partial Q_{i\phi}}$
  - $b_x = b_x - \eta \frac{\partial J}{\partial b_x}$, $\quad b_i = b_i - \eta \frac{\partial J}{\partial b_i}$

Performance of Various Models



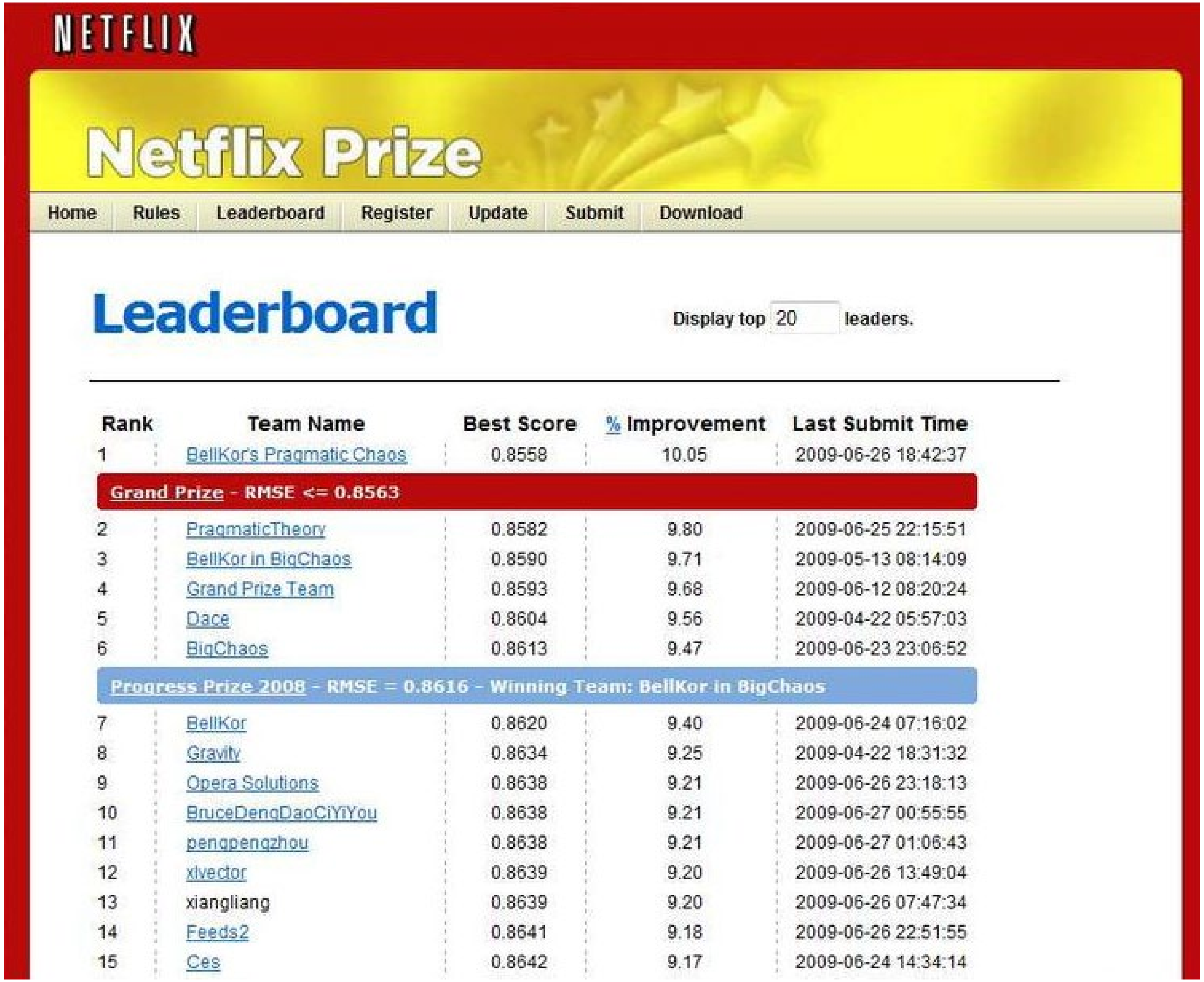
$ 1 Million Awarded on September 21st, 2009

BellKor: A “Kitchen Sink” Approach



- Linear combination of many models:
  - Different numbers of factors
  - Different ways to model time
  - Handling implicit feedback
  - Nearest-neighbor models
  - Restricted Boltzmann Machine
  - …

Leaderboard on June 26th, 2009



NETFLIX

Netflix Prize

Home | Rules | Leaderboard | Register | Update | Submit | Download

Leaderboard

Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	PragmaticTheory	0.8582	9.80	2009-06-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-05-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-06-12 08:20:24
5	Dace	0.8604	9.56	2009-04-22 05:57:03
6	BigChaos	0.8613	9.47	2009-06-23 23:06:52
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
7	BellKor	0.8620	9.40	2009-06-24 07:16:02
8	Gravity	0.8634	9.25	2009-04-22 18:31:32
9	Opera Solutions	0.8638	9.21	2009-06-26 23:18:13
10	BruceDengDaoCiYiYou	0.8638	9.21	2009-06-27 00:55:55
11	pengpengzhou	0.8638	9.21	2009-06-27 01:06:43
12	xlvector	0.8639	9.20	2009-06-26 13:49:04
13	xiangliang	0.8639	9.20	2009-06-26 07:47:34
14	Feeds2	0.8641	9.18	2009-06-26 22:51:55
15	Ces	0.8642	9.17	2009-06-24 14:34:14

The June 26th submission triggered a 30-day "last call".

The Last 30 Days

- An "*Ensemble*" team formed
  - Group of other teams on the leaderboard forms a new team
  - Relies on combining their models
  - Quickly also get a qualifying score over 10%

- BellKor
  - Continue to get small improvements in their scores
  - Realize that they are in direct competition with Ensemble

- Both teams carefully monitoring the leaderboard
- Only sure way to check for improvement is to submit a set of predictions
  - This alerts the other team of your latest score

24 Hours from the Deadline

- Submissions limited to 1 per day
    - Only 1 final submission could be made in the last 24h
- 24 hours before deadline…
    - BellKor team member in Austria noticed (by chance) that Ensemble posted a score that was slightly better than BellKor's

Leaderboard

Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	The Ensemble	0.8554	10.09	2009-07-25 18:32:29
2	BellKor's Pragmatic Chaos	0.8555	10.08	2009-07-25 15:53:34
Grand Prize - RMSE <= 0.8563				
3	Grand Prize Team	0.8571	9.91	2009-07-24 13:07:49
4	Opera Solutions and Vandelay United	0.8573	9.89	2009-07-25 20:05:52

24 Hours from the Deadline

- Frantic last 24 hours for both teams
  - Much computer time on final optimization
  - Carefully calibrated to end about an hour before deadline
- Final submissions
  - BellKor submitted a little early (on purpose), 40 mins before deadline
  - Ensemble submitted their final entry 20 mins later
  - … and everyone waited …

NETFLIX

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update | Download

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
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
16	Invisible Ideas	0.8653	9.15	2009-07-15 15:53:04
17	Just a guy in a garage	0.8662	9.06	2009-05-24 10:02:54
18	J Dennis Su	0.8666	9.02	2009-03-07 17:16:17
19	Craig Carmichael	0.8666	9.02	2009-07-25 16:00:54
20	acmehill	0.8668	9.00	2009-03-21 16:20:50
Progress Prize 2007 - RMSE = 0.8723 - Winning Team: KorBell				

Tie Breaker:
Time of submission!

Next Lecture: Implicit Feedback

- So far, we have focused mostly on estimating explicit ratings (e.g., 1 star – 5 stars)
- What if we only have implicit feedback?
  - E.g., clicks, likes, views, …
  - Only a (small) fraction of customers who purchase a product actually leave a rating
  - Not to mention that the number of users who merely view or click on it is much larger
  - Challenge 1: No negative feedback!
    - If I have not viewed a YouTube video, does that mean I hate it? Or I just have not been exposed to it yet?
  - Challenge 2: Evaluation is tricky — no RMSE to measure!

Bayesian Personalized Ranking [Rendle et al., UAI 2009]

BPR: Bayesian Personalized Ranking from Implicit Feedback

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme
{srendle, freudenthaler, gantner, schmidt-thieme}@ismll.de
Machine Learning Lab, University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany

Abstract

Item recommendation is the task of predicting a personalized ranking on a set of items (e.g. websites, movies, products). In this paper, we investigate the most common scenario with implicit feedback (e.g. clicks, purchases). There are many methods for item recommendation from implicit feedback like matrix factorization (MF) or adaptive k-nearest-neighbor (kNN). Even though these methods are designed for the item prediction task of personalized ranking, none of

sonalization is attractive both for content providers, who can increase sales or views, and for customers, who can find interesting content more easily. In this paper, we focus on item recommendation. The task of item recommendation is to create a user-specific ranking for a set of items. Preferences of users about items are learned from the user's past interaction with the system – e.g. his buying history, viewing history, etc.

Recommender systems are an active topic of research. Most recent work is on scenarios where users provide explicit feedback, e.g. in terms of ratings. Nevertheless, in real-world scenarios most feedback is not explicit but implicit. Implicit feedback is tracked au-

Quiz 2

Thank You!

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html