Recommendation

Hsiang-Hsuan Hung (HHH)

Hsiang.Hung2015@gmail.com

Why Do We Need Recommendation?

I like Budweiser!!!

, , ….

> 3000 products

Enhance customers’ satisfaction, so more purchase

Why Do We Need Recommendation?

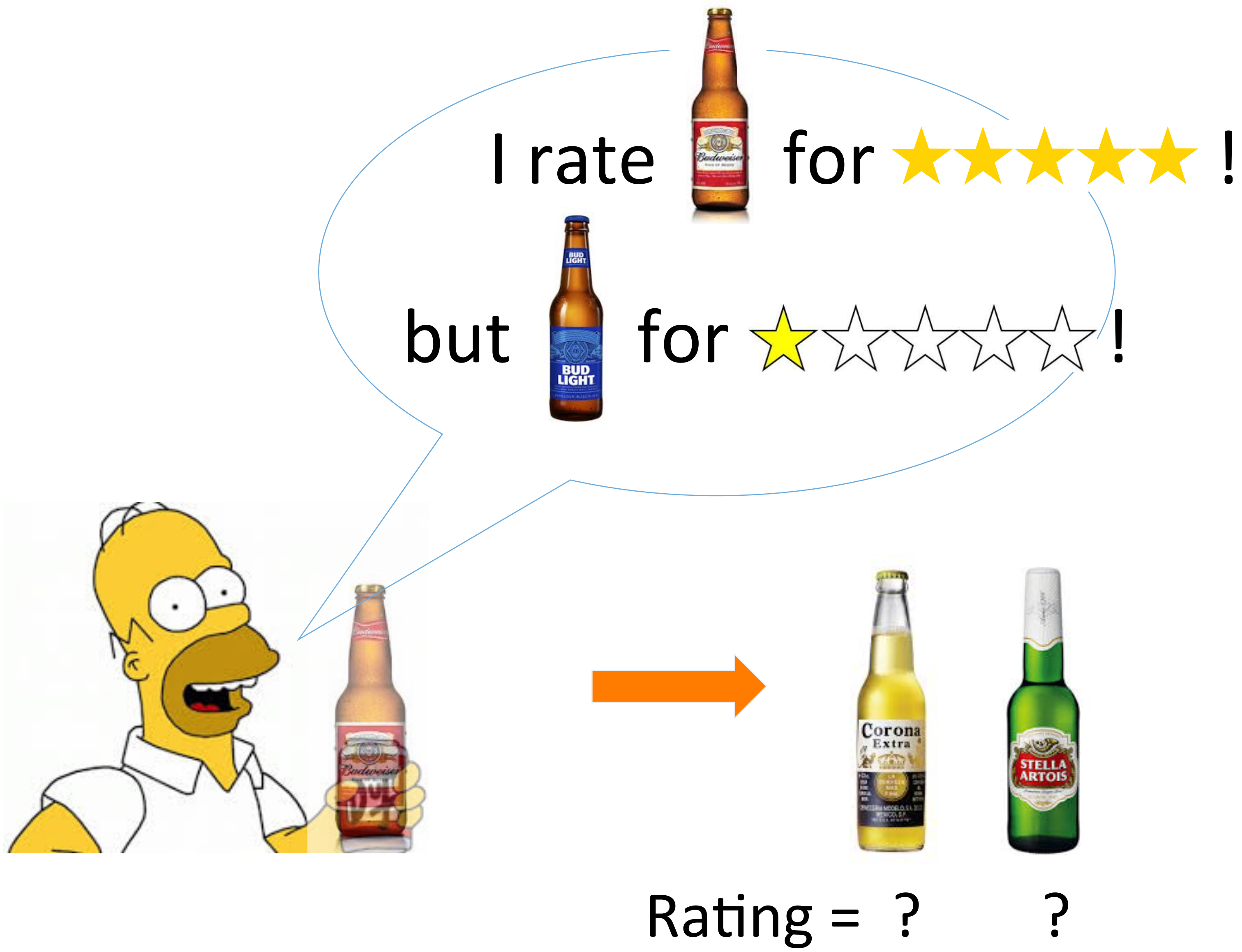
- Always give good recommendation

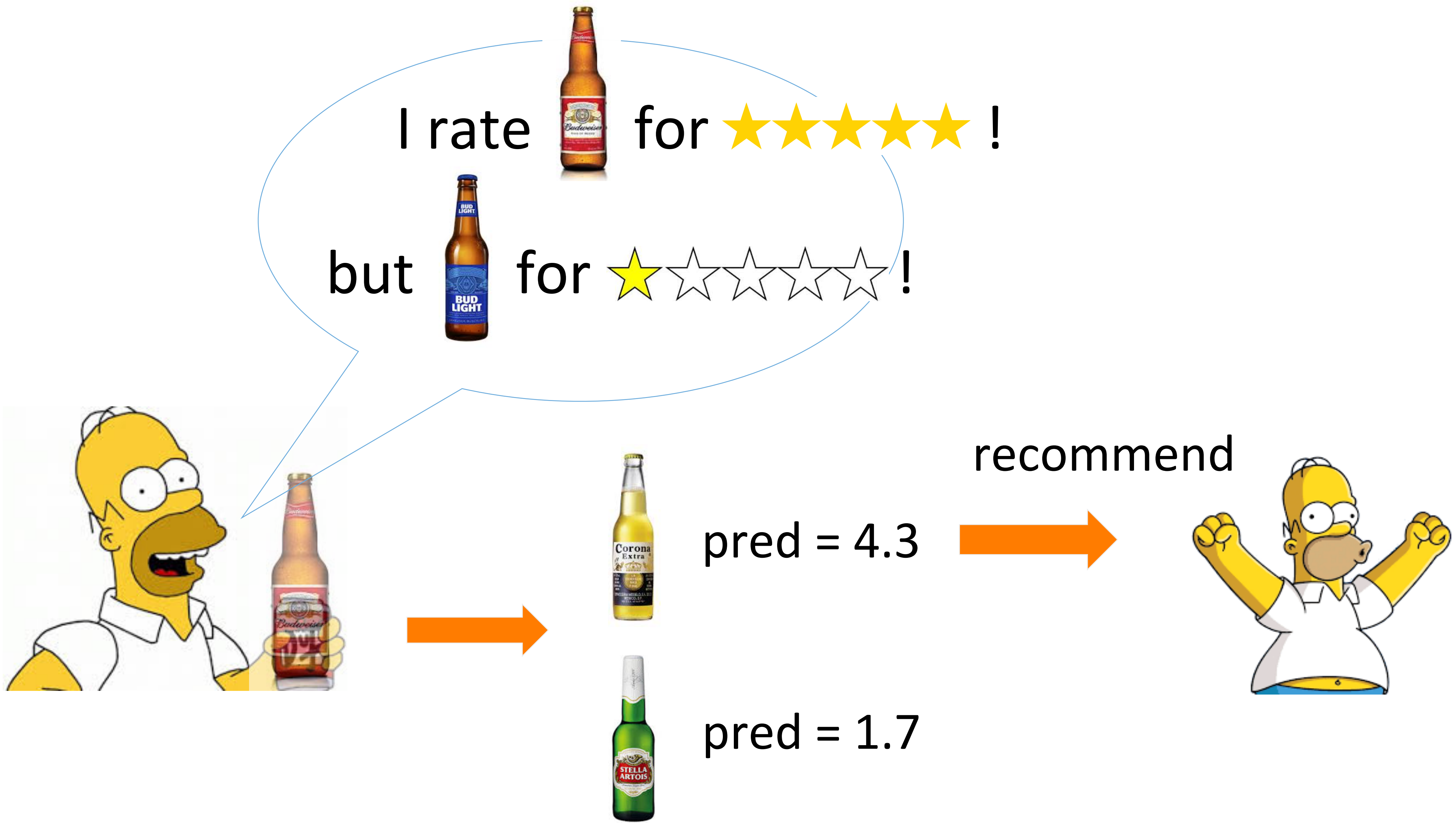

3 February, 2012

Recommendations help drive 27.9% holiday sales growth at John Lewis

By David Moth @ Econsultancy

- Understand customers' demographic characteristics


I rate for ★★★★★ !
Budweiser
but for ★☆☆☆☆ !
BUD LIGHT
Corona Extra
STELLA ARTOIS
Rating = ? ?


I rate for ★★★★★ !
but for ★☆☆☆☆!
BUD LIGHT
Corona Extra
pred = 4.3
STELLA ARTOIS
pred = 1.7
recommend

Personalized Recommendation

Different people have different tastes, so rating.

	5	4.5	3	1
	1.5	1	3.5	5

Personalized Recommendation

- Supervised learning, regression problem.
- Central concept: **Similarity**

    a.) item-item recommendation (Amazon)

    b.) user-item recommendation

        Content-based filtering (Pandora)

        Collaborative filtering (Netflix, Spotify)

Item-Item recommendation

Recommendation Engine UI

When Simpsons (id=7) is browsing #144 beer:

user_id= 7

	user_id	beer_id	pred_rating
0	7	146	5
1	7	20	5
2	7	27	5
3	7	31	5
4	7	45	5
5	7	54	5
6	7	57	5
7	7	79	5
8	7	81	5
9	7	106	5

beer: 144

	beer1	beer2	score
0	144	144	1.000000
1	144	2671	0.352322
2	144	3467	0.281549
3	144	120	0.251629
4	144	129	0.251033
5	144	123	0.249825
6	144	79	0.240634
7	144	32	0.225036
8	144	6959	0.217816
9	144	95	0.214369

Recommendation Engine UI

When Simpsons (id=7) is browsing #144 beer:

user_id= 7

	user_id	beer_id	pred_rating
0	7	146	5
1	7	20	5
2	7	27	5
3	7	31	5
4	7	45	5
5	7	54	5
6	7	57	5
7	7	79	5
8	7	81	5
9	7	106	5

beer: 144

	beer1	beer2	score
0	144	144	1.000000
1	144	2671	0.352322
2	144	3467	0.281549
3	144	120	0.251629
4	144	129	0.251033
5	144	123	0.249825
6	144	79	0.240634
7	144	32	0.225036
8	144	6959	0.217816
9	144	95	0.214369

item-item reco


Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series) 1st Edition
by Kevin P. Murphy (Author)
62 customer reviews
Look inside
Machine Learning
A Probabilistic Perspective
Kevin P. Murphy
eTextbook
$74.92
Hardcover
$95.70
Other Sellers
from $66.92
Buy new
In Stock.
Ships from and sold by Amazon.com. Gift-wrap available.
Want it tomorrow, Oct. 29? Order within 3 hrs 1 min and choose Saturday Delivery at checkout. Details
List Price: $105.00 Save:
34 New f
FREE
Add to Cart
Turn on 1-Click orderi
Customers Who Bought This Item Also Bought
Pattern Recognition and Machine Learning (Information Science and…
Christopher Bishop
125
Hardcover
$75.96 Prime
The Elements of Statistical Learning: Data Mining, Inference, and…
Trevor Hastie
77
#1 Best Seller in Bioinformatics
Hardcover
$73.35 Prime
Probabilistic Graphical Models: Principles and Techniques (Adaptive…
Daphne Koller
36
Hardcover
$108.00 Prime
Learning From Data
Yaser S. Abu-Mostafa
116
Hardcover
Fundamentals of Machine Learning for Predictive Data Analytics:…
John D. Kelleher
14
Hardcover
$74.00 Prime
Python Machine Learning
Sebastian Raschka
77
#1 Best Seller in Data Processing
Paperback
$40.49 Prime

Data Source: BeHoppy (SQL)

The Beer Content Data

```
beers.head()
```

Out[799]:

	id	country	flavor	style	average_rating	number_of_ratings	average_aroma_rating	average_appearance_rating	average_taste_ratin
0	5	BEL	Amadeirado	Lager	3.75481	208	3.45238	3.69531	3.61417
1	6	BEL	Doce	Ale	4.22901	131	4.20430	4.22680	4.26804
2	7	MEX	Doce	Lager	3.90435	230	3.39583	3.88194	3.68750
3	8	BRA	Doce	Bohemian Pilsener	3.79474	190	3.16239	3.38136	3.57627
4	9	EUA	Doce	Lager	3.61244	209	3.13492	3.34646	3.36800

The Beer Content Data

```
beers.head()
```

Out[799]:

Beer id

	id	country	flavor	style	average_rating	number_of_ratings	average_aroma_rating	average_appearance_rating	average_taste_ratin
0	5	BEL	Amadeirado	Lager	3.75481	208	3.45238	3.69531	3.61417
1	6	BEL	Doce	Ale	4.22901	131	4.20430	4.22680	4.26804
2	7	MEX	Doce	Lager	3.90435	230	3.39583	3.88194	3.68750
3	8	BRA	Doce	Bohemian Pilsener	3.79474	190	3.16239	3.38136	3.57627
4	9	EUA	Doce	Lager	3.61244	209	3.13492	3.34646	3.36800

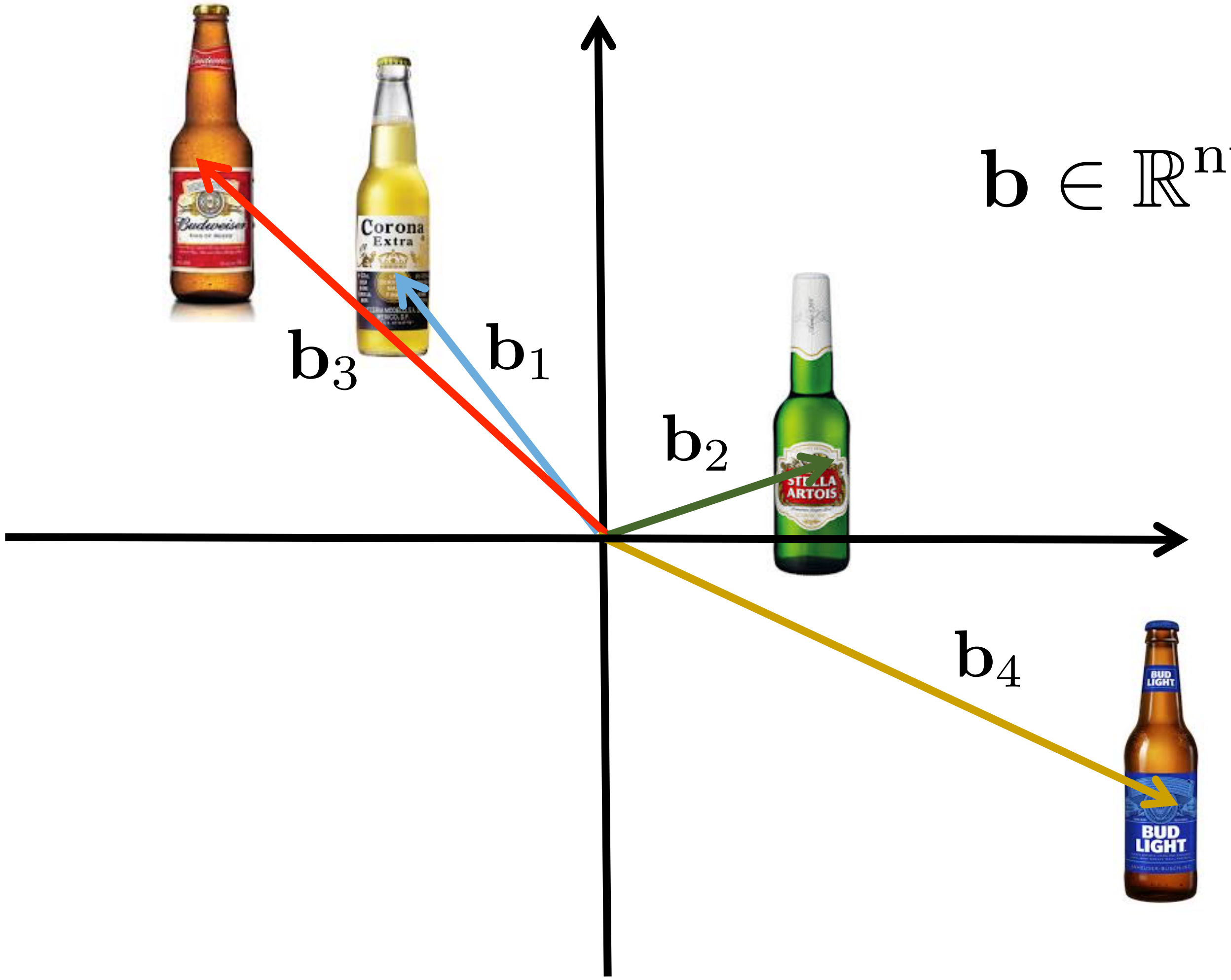
Beer vectors are represented by features:

$$\mathbf{b}_1 = (208, 3.45, 3.69, 3.61, 3.44, 0, 0, \cdots)$$

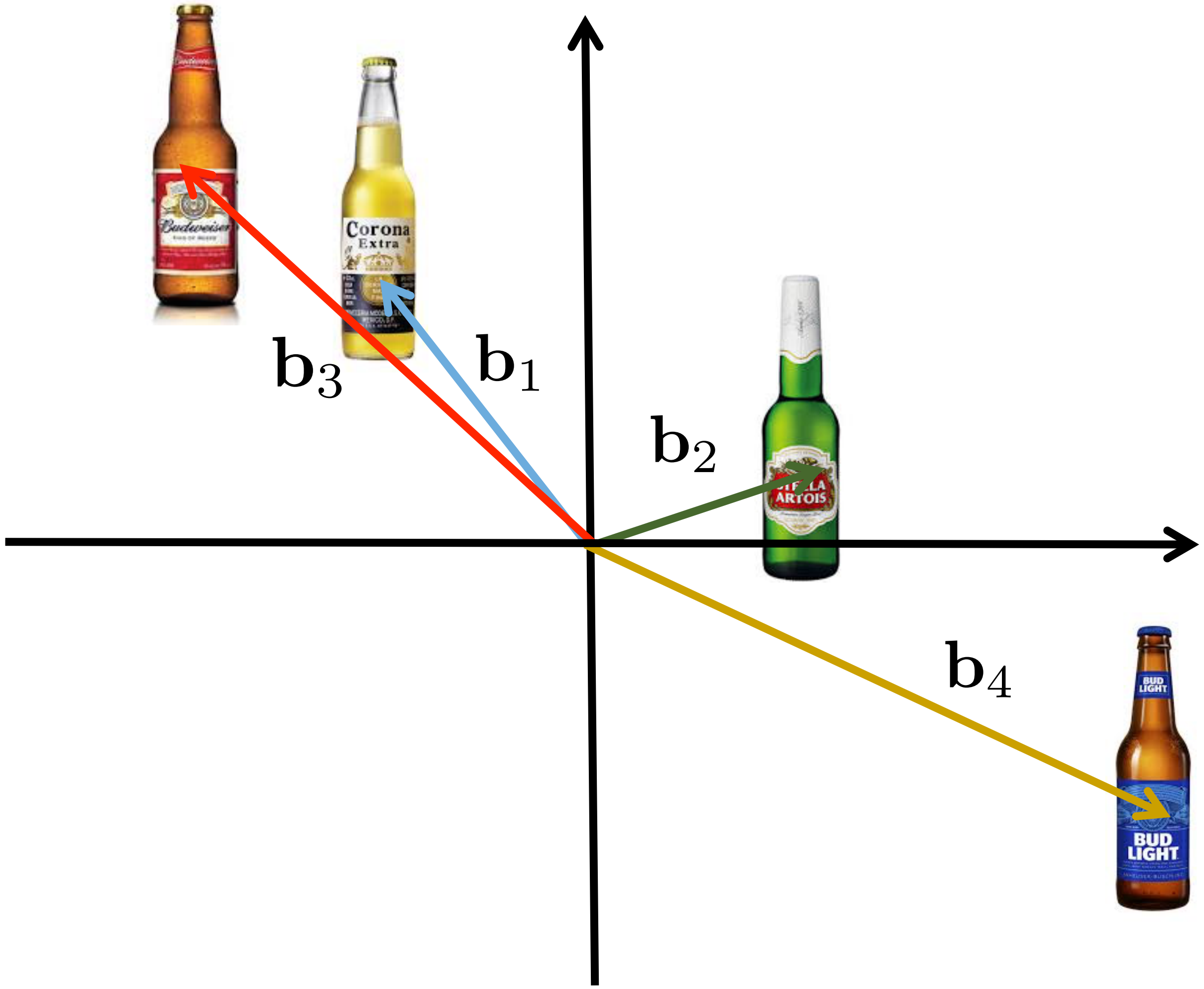
$$\mathbf{b}_2 = (131, 4.2, 4.22, 4.27, 4.1, 0, 0, \cdots)$$

Beer Vector Space

$\mathbf{b} \in \mathbb{R}^{\text{num of features}}$



Cosine Similarity



$$s_{A,B} = \frac{\mathbf{b}_A \cdot \mathbf{b}_B}{|\mathbf{b}_A||\mathbf{b}_B|}$$

$$s_{31} > s_{32} > s_{34}$$


is more similar

to  than 

Cosine Similarity

beer: 79

	beer1	beer2	score
0	79	79	1.000000
1	79	128	0.339833
2	79	2671	0.299069
3	79	123	0.280518
4	79	55	0.261834
5	79	54	0.250267
6	79	129	0.242569
7	79	144	0.240634
8	79	201	0.240175
9	79	3357	0.237854

$\mathbf{b}_4$

$$s_{A,B} = \frac{\mathbf{b}_A \cdot \mathbf{b}_B}{|\mathbf{b}_A||\mathbf{b}_B|}$$

$$s_{31} > s_{32} > s_{34}$$

is more similar

to  than 

Recommendation Engine UI

When Simpsons (id=7) is browsing the beer page:

user_id= 7

	user_id	beer_id	pred_rating
0	7	146	5
1	7	20	5
2	7	27	5
3	7	31	5
4	7	45	5
5	7	54	5
6	7	57	5
7	7	79	5
8	7	81	5
9	7	106	5

user-item reco

beer: 144

	beer1	beer2	score
0	144	144	1.000000
1	144	2671	0.352322
2	144	3467	0.281549
3	144	120	0.251629
4	144	129	0.251033
5	144	123	0.249825
6	144	79	0.240634
7	144	32	0.225036
8	144	6959	0.217816
9	144	95	0.214369

User-Item recommendation

Content-Based Filtering

Content-based reco systems, Pazzaniıand and Billsus

customers' ratings:

	score	user_id	beer_id
0	4	7	6
1	4	48	7
2	4	47	9
3	4	47	10
4	4	50	9

Simpsons

7	254	5
7	251	4
7	210	4
7	200	3

$\mathbf{b}_1$ 

$r_{S,1}$

$\mathbf{b}_2$ 

$r_{S,2}$

Content-Based Filtering

Content-based reco systems, Pazzaniıand and Billsus

customers' ratings:

	score	user_id	beer_id
0	4	7	6
1	4	48	7
2	4	47	9
3	4	47	10
4	4	50	9

Simpsons

7	254	5
7	251	4
7	210	4
7	200	3

HHH

50	38	4
50	5	5

$\mathbf{b}_1$ 

$r_{S,1}$ $r_{H,1}$

$\mathbf{b}_2$ 

$r_{S,2}$ $r_{H,2}$

Content-Based Filtering

Each user is a regression problem:

7	254	5
7	251	4
7	210	4
7	200	3

$\mathbf{u}_7 = ?$

$$5 \simeq \mathbf{u}_7^T \mathbf{b}_{254}, \quad 4 \simeq \mathbf{u}_7^T \mathbf{b}_{251}, \quad 3 \simeq \mathbf{u}_7^T \mathbf{b}_{200} \quad \cdots$$

$$\min_{\mathbf{u}_7} \sum_{i=\text{beers rated by id=7 user}} \left(r_{7,i} - \mathbf{u}_7^T \mathbf{b}_i \right)^2 \Rightarrow \mathbf{u}_7 = (0.1, 0.35, 0, \cdots)$$

Content-Based Filtering

Each user is a regression problem:

$\mathbf{u}_7 = ?$

7	254	5
7	251	4
7	210	4
7	200	3

$$5 \simeq \mathbf{u}_7^T \mathbf{b}_{254}, \quad 4 \simeq \mathbf{u}_7^T \mathbf{b}_{251}, \quad 3 \simeq \mathbf{u}_7^T \mathbf{b}_{200} \quad \cdots$$

$$\min_{\mathbf{u}_7} \sum_{i=\text{beers rated by id=7 user}} \left(r_{7,i} - \mathbf{u}_7^T \mathbf{b}_i\right)^2 \rightarrow \mathbf{u}_7 = (0.1, 0.35, 0, \cdots)$$

$\mathbf{u}_{50} = ?$

50	38	4
50	5	5

$$4 \simeq \mathbf{u}_{50}^T \mathbf{b}_{38}, \quad 5 \simeq \mathbf{u}_{50}^T \mathbf{b}_5, \rightarrow \mathbf{u}_{50} = (0, 0.05, 0.7, \cdots)$$

Predict

$$r_{7,1} = \mathbf{u}_7^T \mathbf{b}_1, \quad r_{50,1} = \mathbf{u}_{50}^T \mathbf{b}_1$$

Cold Start: Spare Rating Data

number of users: 1728
number of beers: 1854
number of ratings: 12360
sparsity = 0.9962%

D: For most users, little data points, low accuracy:

```
-----------------------------
user= 3556 , n_data= 122
41 0.707317073171
-----------------------------
user= 4328 , n_data= 115
38 0.763157894737
-----------------------------
```

```
-----------------------------
user= 2600 , n_data= 91
31 0.612903225806
-----------------------------
user= 2754 , n_data= 51
17 0.647058823529
-----------------------------
user= 2877 , n_data= 62
21 0.333333333333
```

Collaborative Filtering (CF)

- No content features are needed.
- Consider user-item interactions

rating table

	score	user_id	beer_id
0	4	7	6
1	4	48	7
2	4	47	9
3	4	47	10
4	4	50	9

beers

users	5	6	7	8	9	10	11
7	5	4		1	5	5	4
33			1				
34	1	2		5	1	2	2
42							
45	3	5	2		5	5	4
47					4	4	
48			4				

Collaborative Filtering (CF)

- No content features are needed.
- Consider user-item interactions
- Neighborhood model
  - item based (use item similarity)
  - user based (use user similarity)
- Latent-factor-model

Vector Space by Beer Rating Vectors

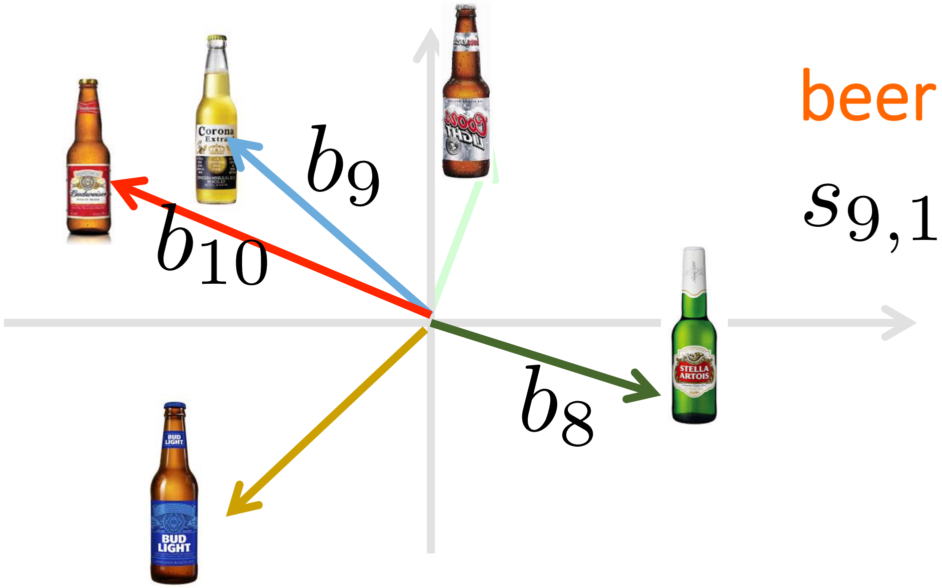
beers

users	..7	8	9	10	11 ..
7		1	5	5	4
33	1			1	
34		5	1	2	2
42	3				
45		1	5	5	3
47			4	4	
48	4	4		1	

$$\mathbf{b}_8 = (1, -, 5, -, 1, -, 4)$$

$$\mathbf{b}_9 = (5, -, 1, -, 5, 4, -)$$

$$\mathbf{b}_{10} = (5, 1, 2, -, 5, 4, 1)$$

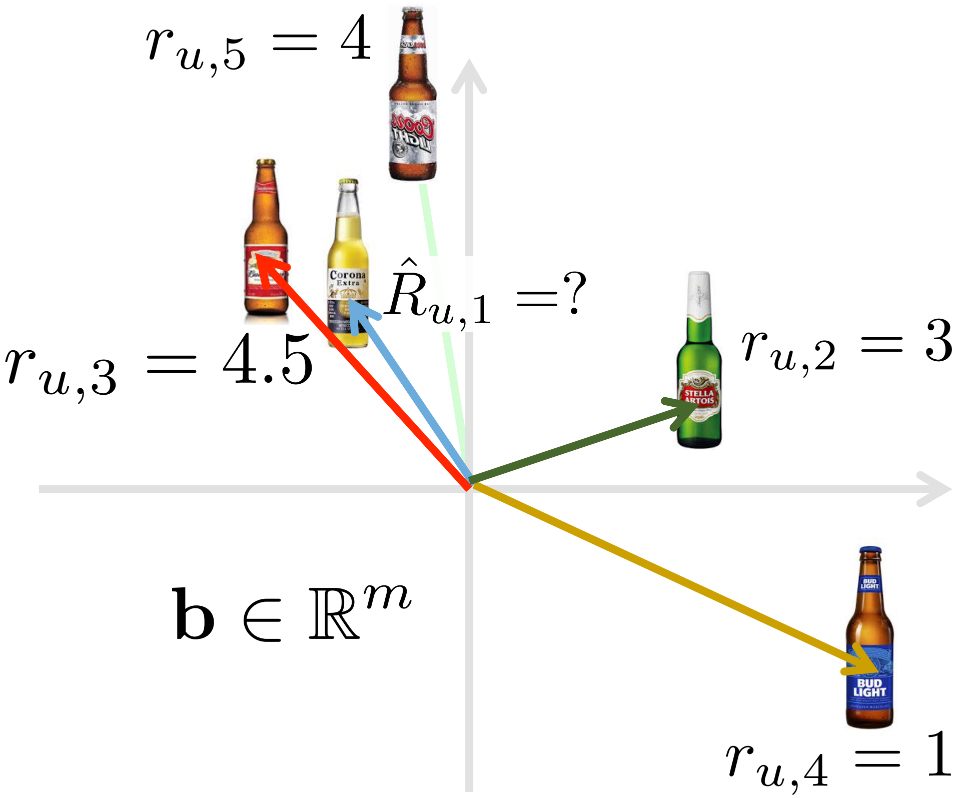


beer similarity:

$$s_{9,10} > s_{9,8}$$

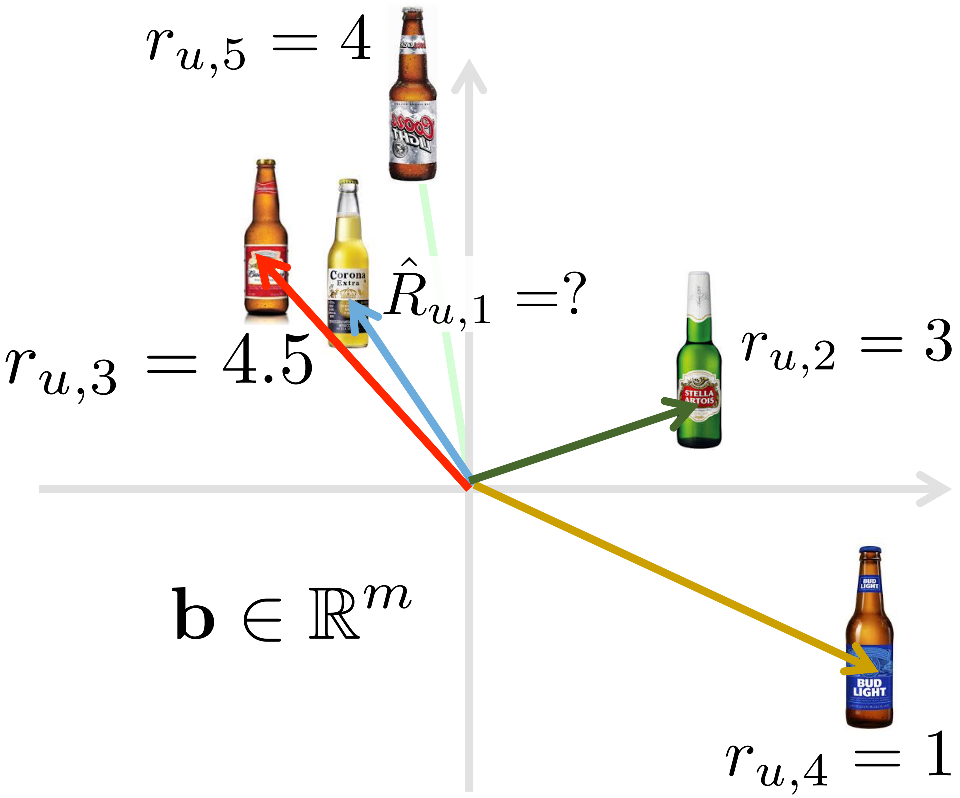
Neighborhood Model – Item Based

Sarwar, Karypis, Konstan, and Riedl , (2001)



Neighborhood Model – Item Based

Sarwar, Karypis, Konstan, and Riedl , (2001)

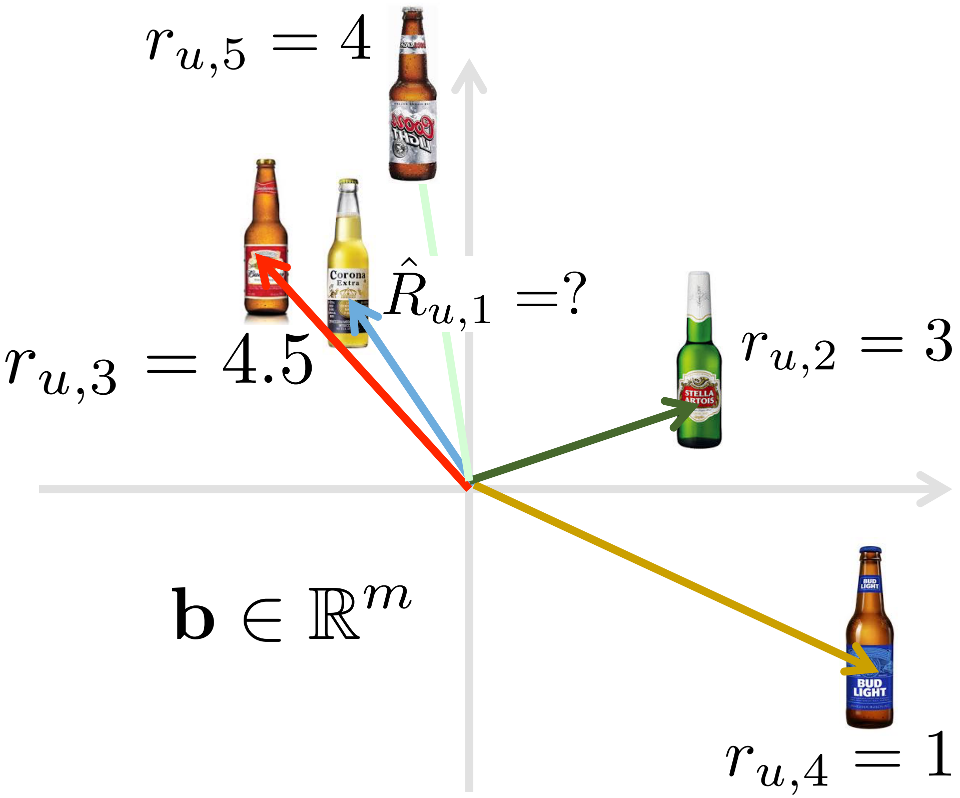


k-nearest-neighbor:

$$\hat{R}_{u,1} = \frac{(4.5 + 4 + 3)}{3}$$

Neighborhood Model – Item Based

Sarwar, Karypis, Konstan, and Riedl , (2001)



Model = kNN + weight:

similarity as weight

$$\hat{R}_{u,i} = \frac{\sum_{j \in S^k} s_{ij} r_{u,j}}{\sum_{j \in S^k} |s_{ij}|}$$

$$= \frac{(0.8 * 4.5 + 0.7 * 4 + 0.2 * 3)}{0.8 + 0.7 + 0.2}$$

Vector Space by User Rating Vectors

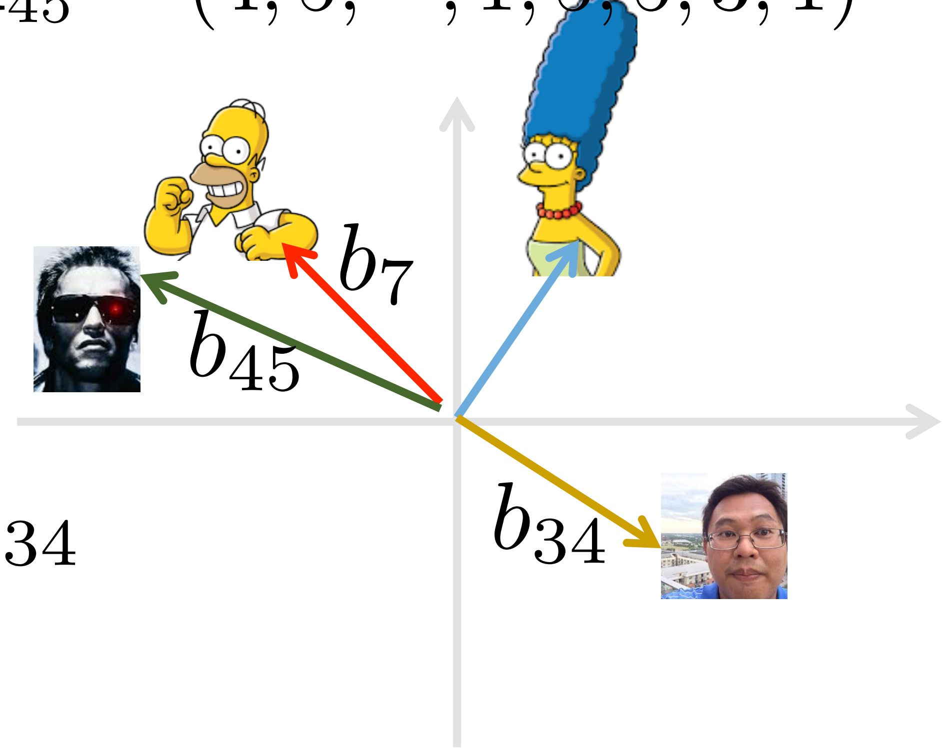
beers

users	5	6	7	8	9	10	11	12
7	5	4		1	5	5	4	
33			1					
34	1	1		5	1	2	2	
42			3					3
45	4	5		1	5	5	3	1

$$\mathbf{u}_7 = (5, 4, -, 1, 5, 5, 4, -)$$

$$\mathbf{u}_{34} = (1, 1, -, 5, 1, 2, 2, -)$$

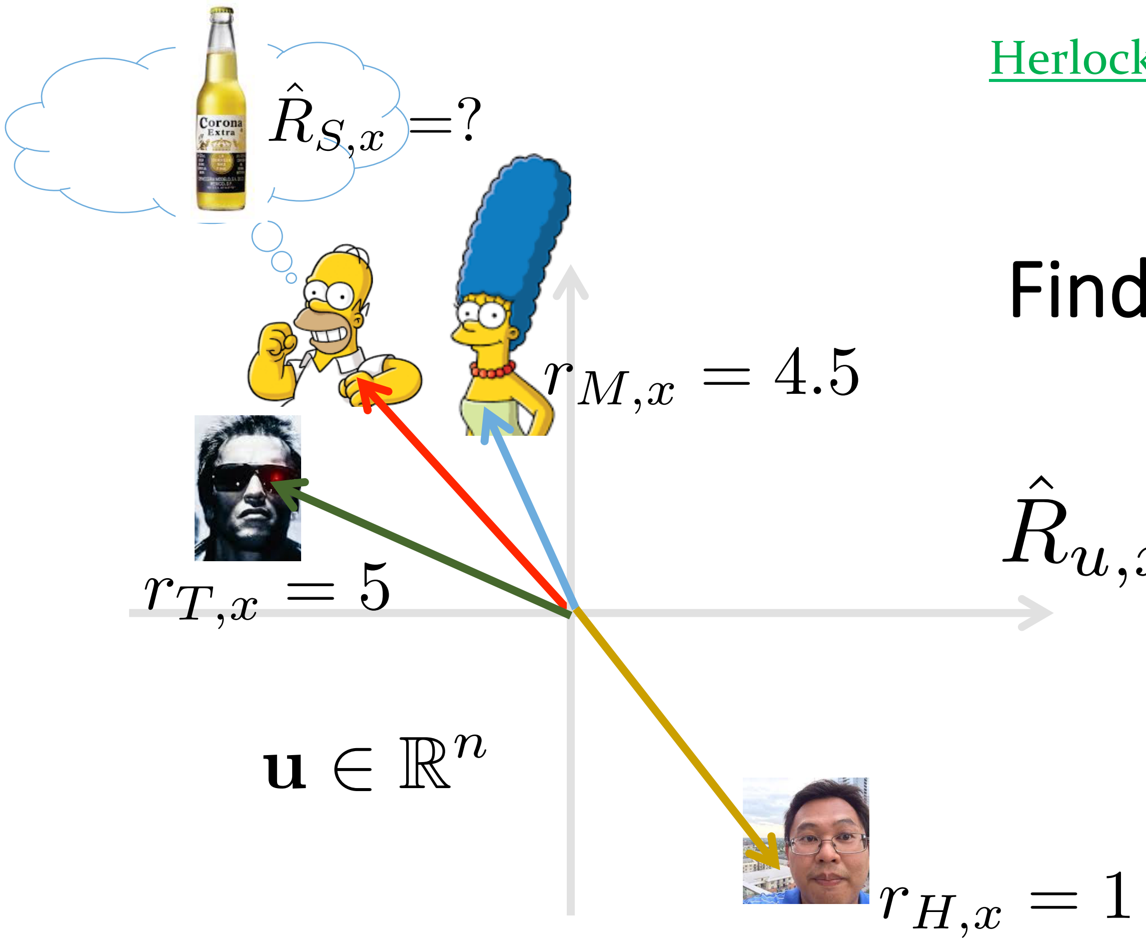
$$\mathbf{u}_{45} = (4, 5, -, 1, 5, 5, 3, 1)$$



user similarity: $s_{45,7} > s_{45,34}$

Neighborhood Model – User Based

Herlocker, Konstan, Borches and Riedl , (1999)



Find similar users:

weight

$$\hat{R}_{u,x} = \frac{\sum_{v \in S^k} s_{uv} r_{v,x}}{\sum_{v \in S^k} |s_{uv}|}$$

Metrics

mean absolute error: $\mathrm{MAE} = \frac{1}{N}\sum_{u,i}|\hat{R}_{u,i} - r_{u,i}|$

Sarwar, Karypis, Konstan, and Riedl , (2001)

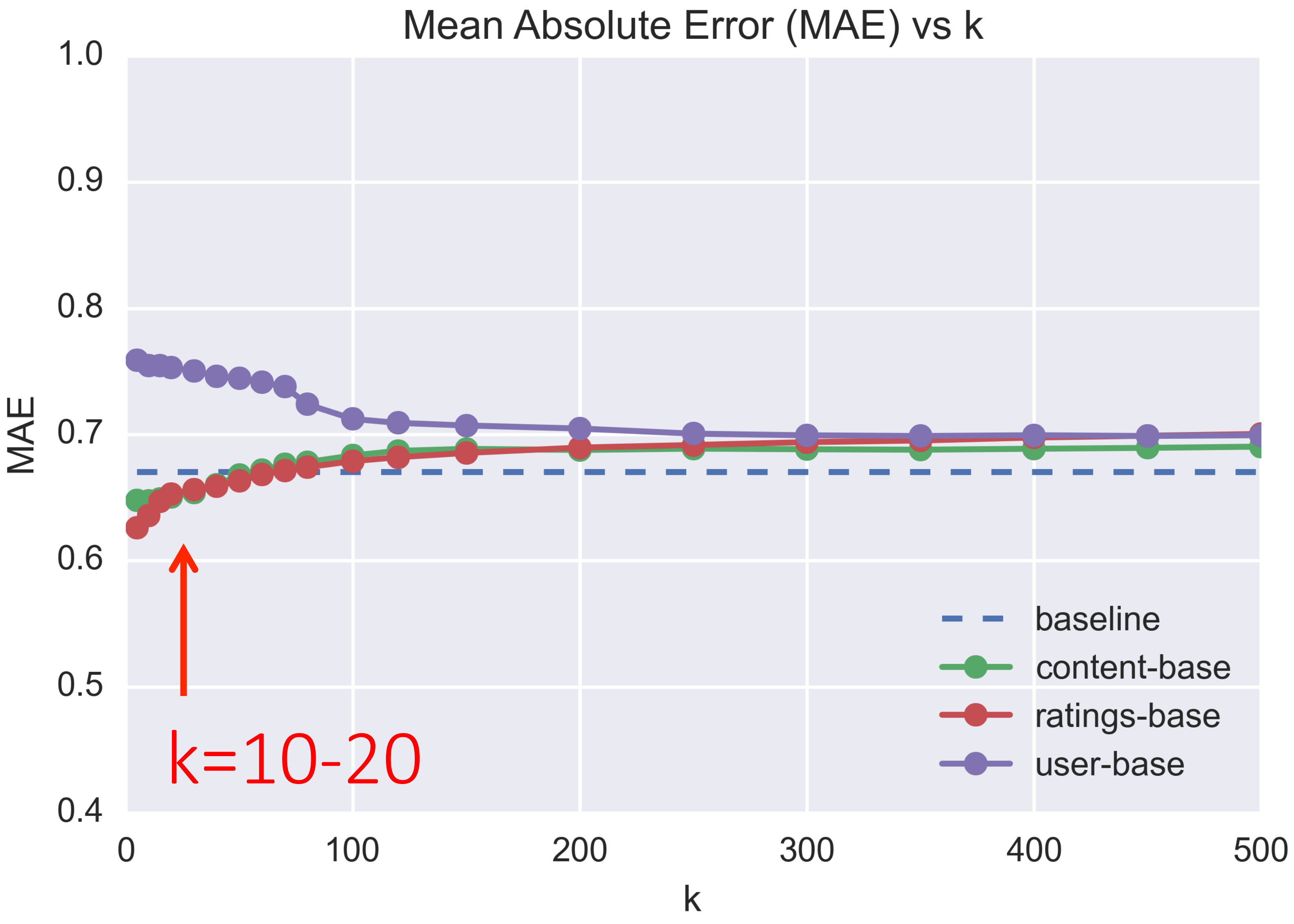
a) item-based:

$\mathbf{b}_i$ = beer content and ratings as features

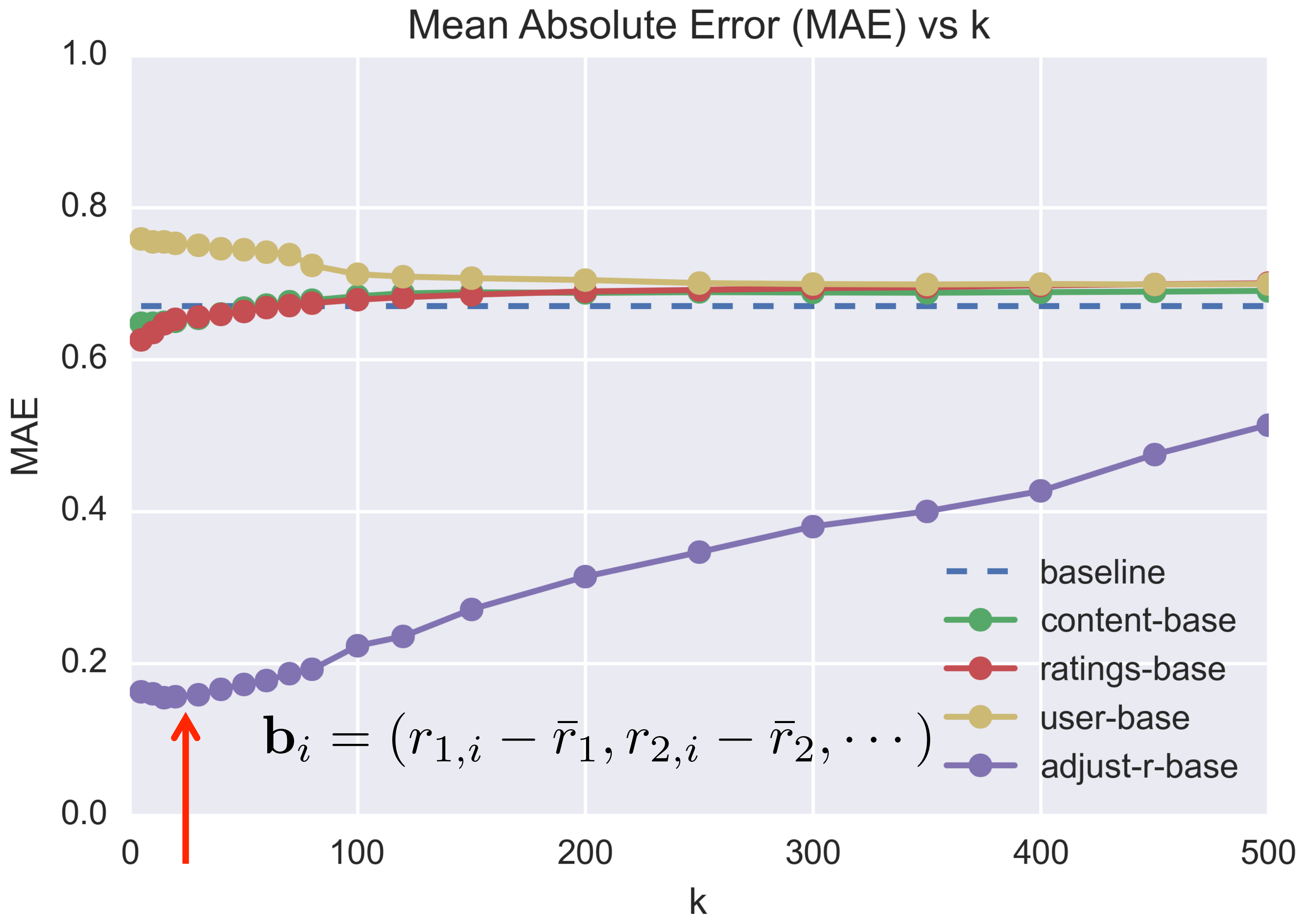
b) user-based: $\mathbf{u}_u$

c) Baseline: $\overline{\mathrm{MAE}} = \frac{1}{N}\sum_{u,i}|\bar{R}_i - r_{u,i}|$

Model Performance



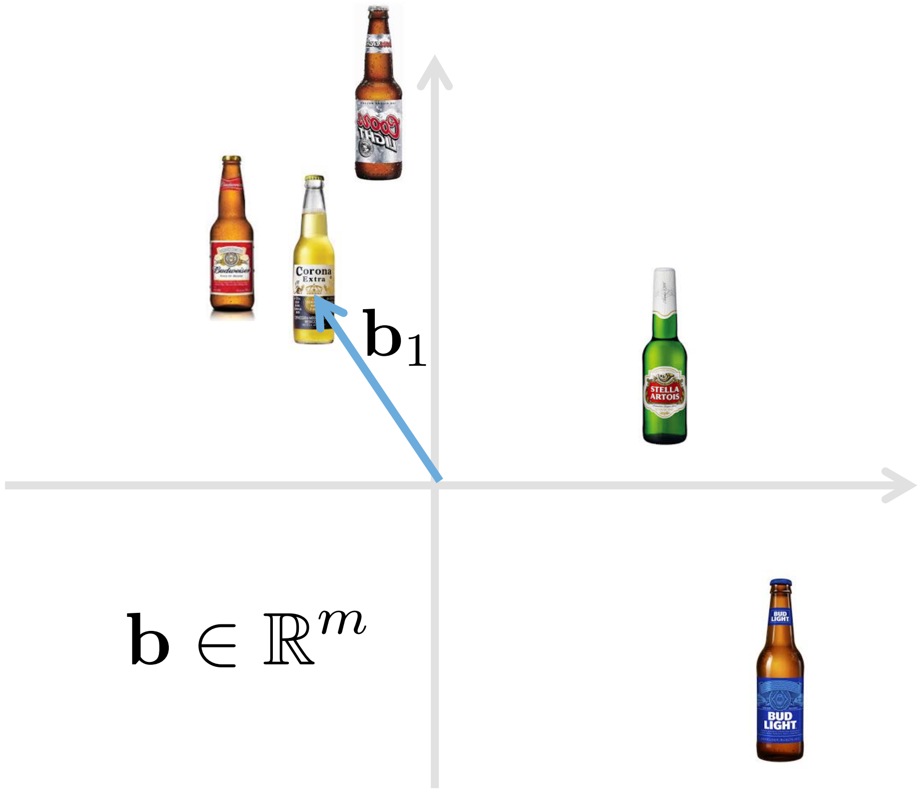
Revised Neighborhood - Adjusted Ratings

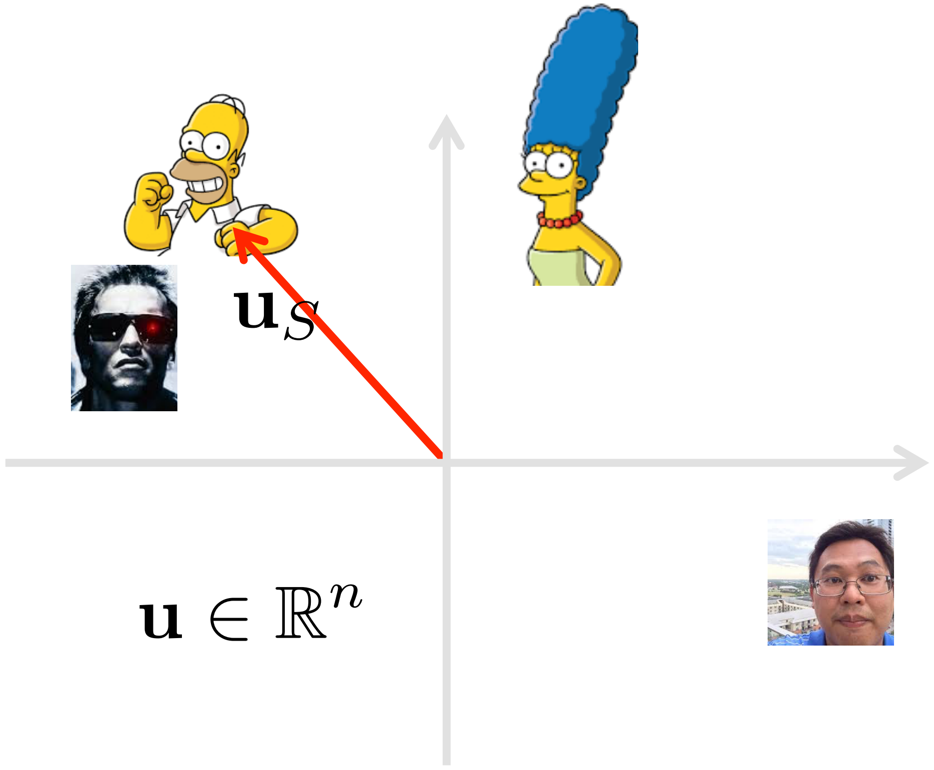


Disadvantages of the Model

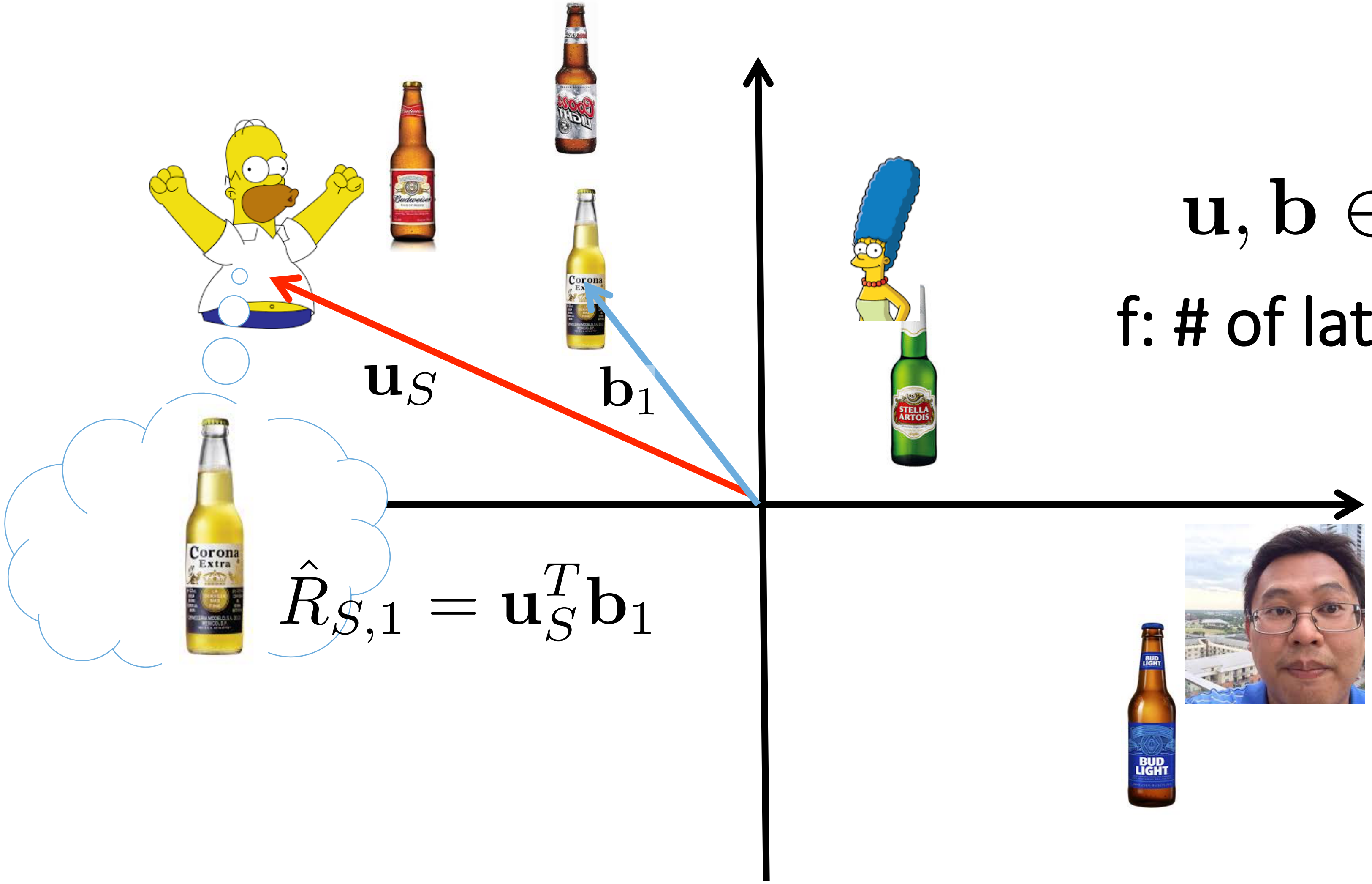
- NOT scalable.
- Not easy to incorporate additional information, e.g. purchase, browse history (implicit data).
- Alternative: **latent-factor model**!

Vector Spaces





Customer-Beer Vector Space

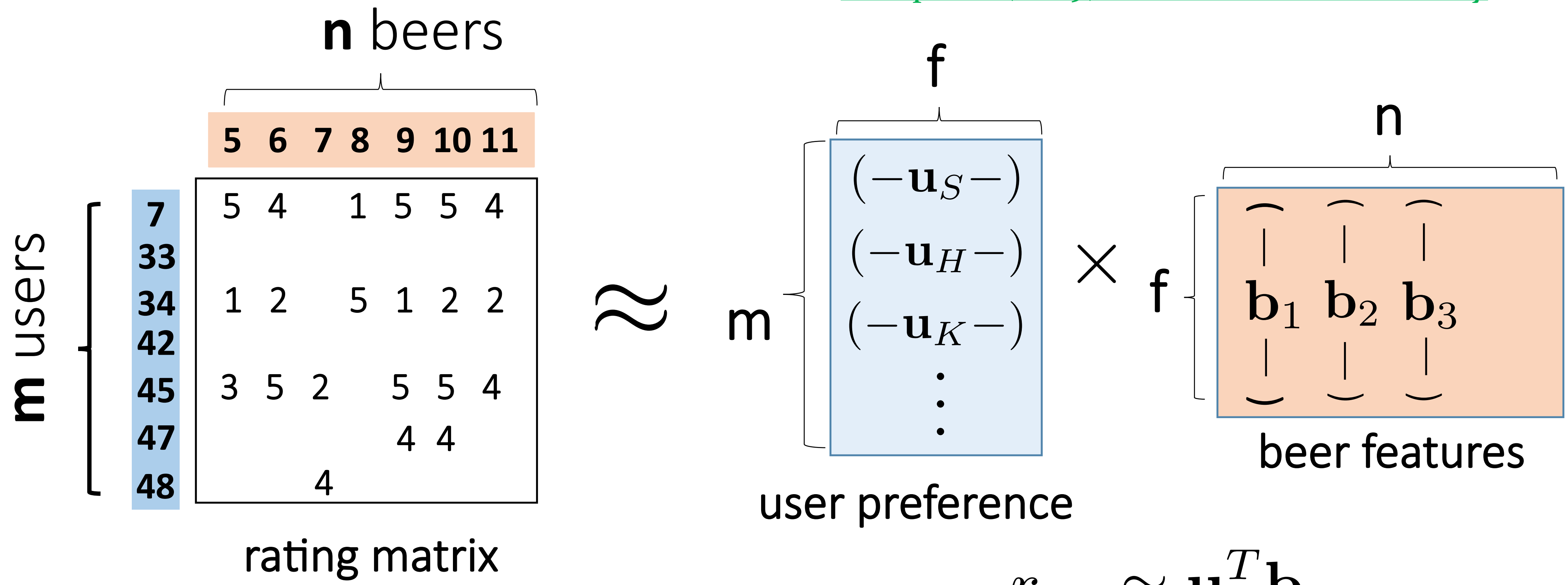


$$\mathbf{u}, \mathbf{b} \in \mathbb{R}^f$$

f: # of latent factors

Matrix-Factorization for Latent-Factor Model

Computer (2009), Koren, Bell and Volinsky



$$r_{u,i} \simeq \mathbf{u}_u^T \mathbf{b}_i$$

Matrix-Factorization $\sim$ Linear Regression

$$\min_{\mathbf{u},\mathbf{b},\xi} \sum_{(u,i)\ \text{if}\ r_{u,i}\neq 0} \Big(r_{u,i} - \underbrace{\mathbf{u}_u^T \mathbf{b}_i}_{\text{user-item interaction}} - \overbrace{\xi_{u,i}}^{\text{bias}}\Big)^2 + \underbrace{\lambda\Big(\sum_u |\mathbf{u}_u|^2 + \sum_i |\mathbf{b}_i|^2\Big)}_{\text{regularization}}$$

- SVD, not scalable
- Gradient descent, not convex problem
- Alternating least square (ALS)!

What is the ALS?

$$\text{Rating matrix} \approx \begin{matrix} (-\mathbf{u}_S-) \\ (-\mathbf{u}_H-) \\ (-\mathbf{u}_K-) \\ \vdots \end{matrix} \times \begin{matrix} \big(& \big(& \big(& \\ | & | & | & \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \cdots \\ | & | & | & \\ \big) & \big) & \big) & \end{matrix} \Rightarrow \mathbf{R} = \mathbf{U}\mathbf{B}^T$$

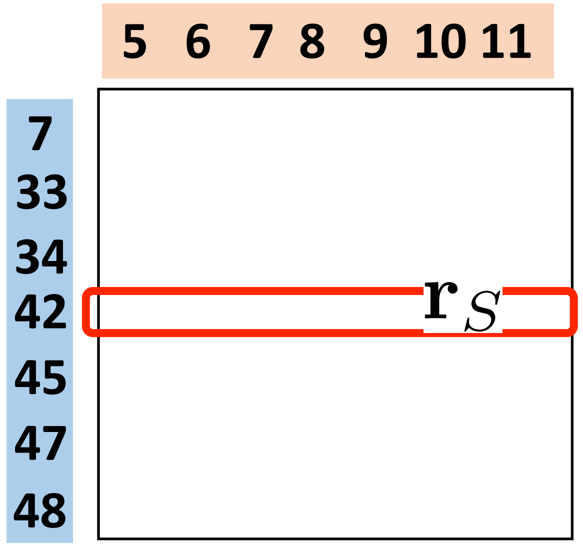
- At each step, fix one variable, and solve minimization:

  fix $\mathbf{u}$, solve $\mathbf{b}$ $\Rightarrow$ fix $\mathbf{b}$, solve $\mathbf{u}$ $\Rightarrow$ fix $\mathbf{u}$, solve $\mathbf{b}$

More Detail: Normal Equations

Hu, Koren and Volinsky, 2008

$$\mathbf{u}_S = \begin{pmatrix} u_{S,1} \\ u_{S,2} \\ \vdots \\ u_{S,f} \end{pmatrix} = \left(\mathbf{B}^T\mathbf{B} + \lambda \mathbf{I}\right)^{-1} \mathbf{B}^T \begin{pmatrix} r_{S,1} \\ r_{S,2} \\ \vdots \\ r_{S,n} \end{pmatrix}$$

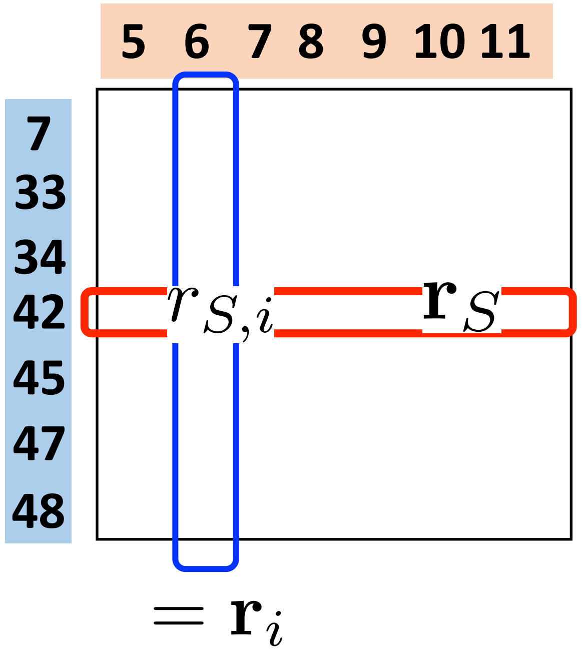


More Detail: Normal Equations

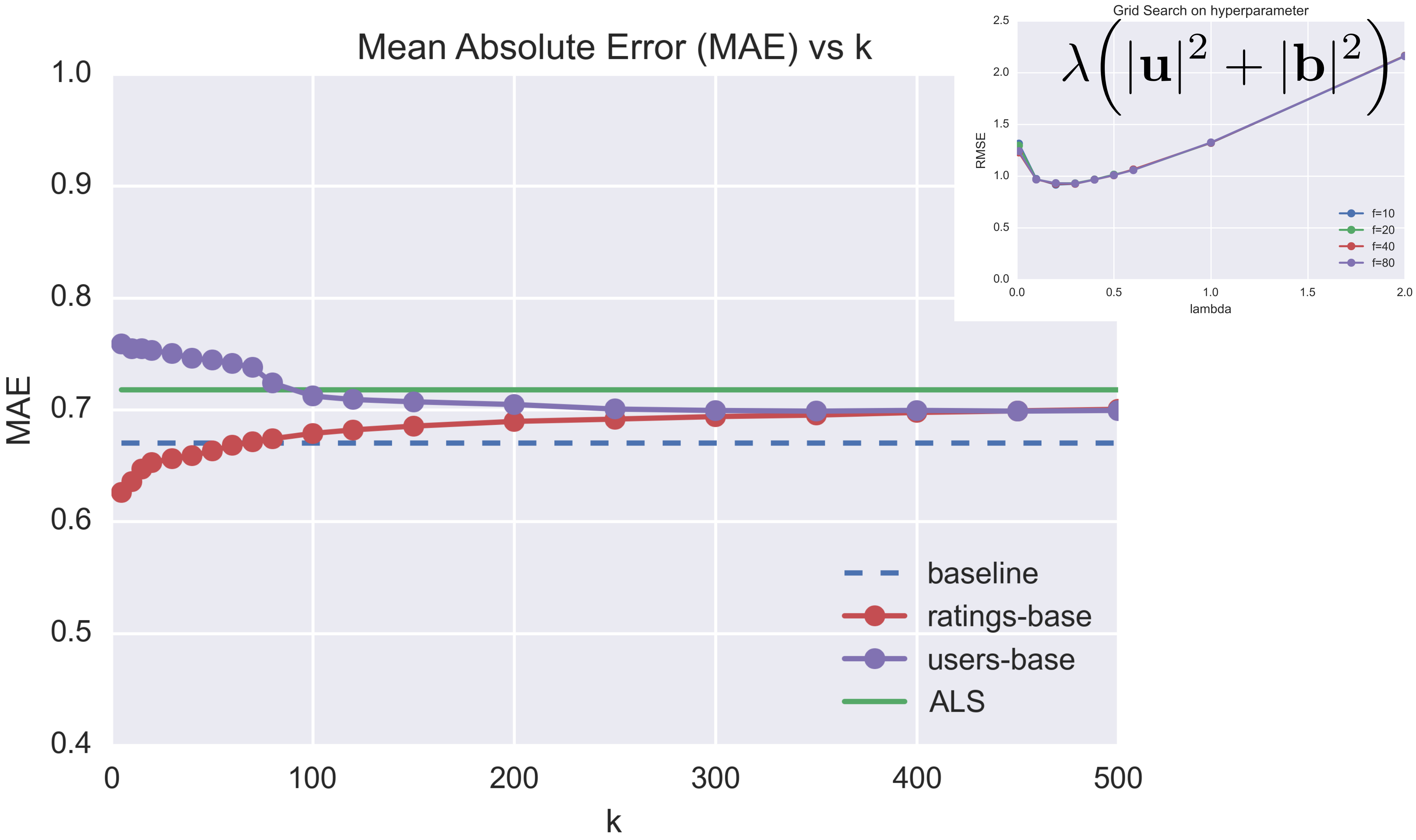
Hu, Koren and Volinsky, 2008

$$\mathbf{u}_S = \begin{pmatrix} u_{S,1} \\ u_{S,2} \\ \vdots \\ u_{S,f} \end{pmatrix} = \left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I}\right)^{-1}\mathbf{B}^T \begin{pmatrix} r_{S,1} \\ r_{S,2} \\ \vdots \\ r_{S,n} \end{pmatrix}$$

$$\mathbf{b}_i = \begin{pmatrix} b_{i,1} \\ b_{i,2} \\ \vdots \\ b_{i,f} \end{pmatrix} = \left(\mathbf{U}^T\mathbf{U} + \lambda\mathbf{I}\right)^{-1}\mathbf{U}^T \begin{pmatrix} r_{1,i} \\ r_{2,i} \\ \vdots \\ r_{m,i} \end{pmatrix}$$



ALS Model Performance (on Spark)



Implicit Data: e-Commerce Data

Explicit feedback
(beHoppy)

	score	user_id	beer_id
0	4	7	6
1	4	48	7
2	4	47	9
3	4	47	10
4	4	50	9

$r_{u,b} = 1 - 5$

ratings

Implicit feedback
(data warehouse)

Hu, Koren and Volinsky, 2008

customer_id	product_id
1	1
1	2
1	1
1	2
1	3
1	191
1	65
2	3
2	3
2	3

$r_{u,b} \in \mathbb{I}$

purchase frequency

CF Using Implicit Data

- **Logistic regression** + confidence weight

Hu, Koren and Volinsky, 2008

$c_{u,i} = f(\alpha, r_{u,i})$

$$\min_{\mathbf{u},\mathbf{b},\xi} \sum_{(u,i)} \overbrace{c_{u,i}}^{\text{confidence}} \Big(p_{u,i} - \underbrace{\mathbf{u}_u^T \mathbf{b}_i}_{\text{user-item interaction}} - \overbrace{\xi_{u,i}}^{\text{bias}}\Big)^2 + \lambda \underbrace{\Big(\sum_u |\mathbf{u}_u|^2 + \sum_i |\mathbf{b}_i|^2\Big)}_{\text{regularization}}$$

preference $p_{u,i} = 0/1 \ (\text{if } r_{u,i} > 0)$

Implicit Data CF Performance

- Metric: percentile-ranking

$$\overline{rank} = \frac{\sum_{u,i} r_{u,i} * rank_{u,i}}{\sum_{u,i} r_{u,i}}$$

- Random: $\overline{rank} = 50\%$ Baseline: $\overline{rank} \sim 29\%$
- CF: $\overline{rank} \sim 16\%$

Recommender Pipeline