Recommender Systems

Hsiang-Hsuan Hung (HHH)

Hsiang.Hung2015@gmail.com

> 3000 products

Enhance customers’ satisfaction, so more purchase

Machine Learning and Business

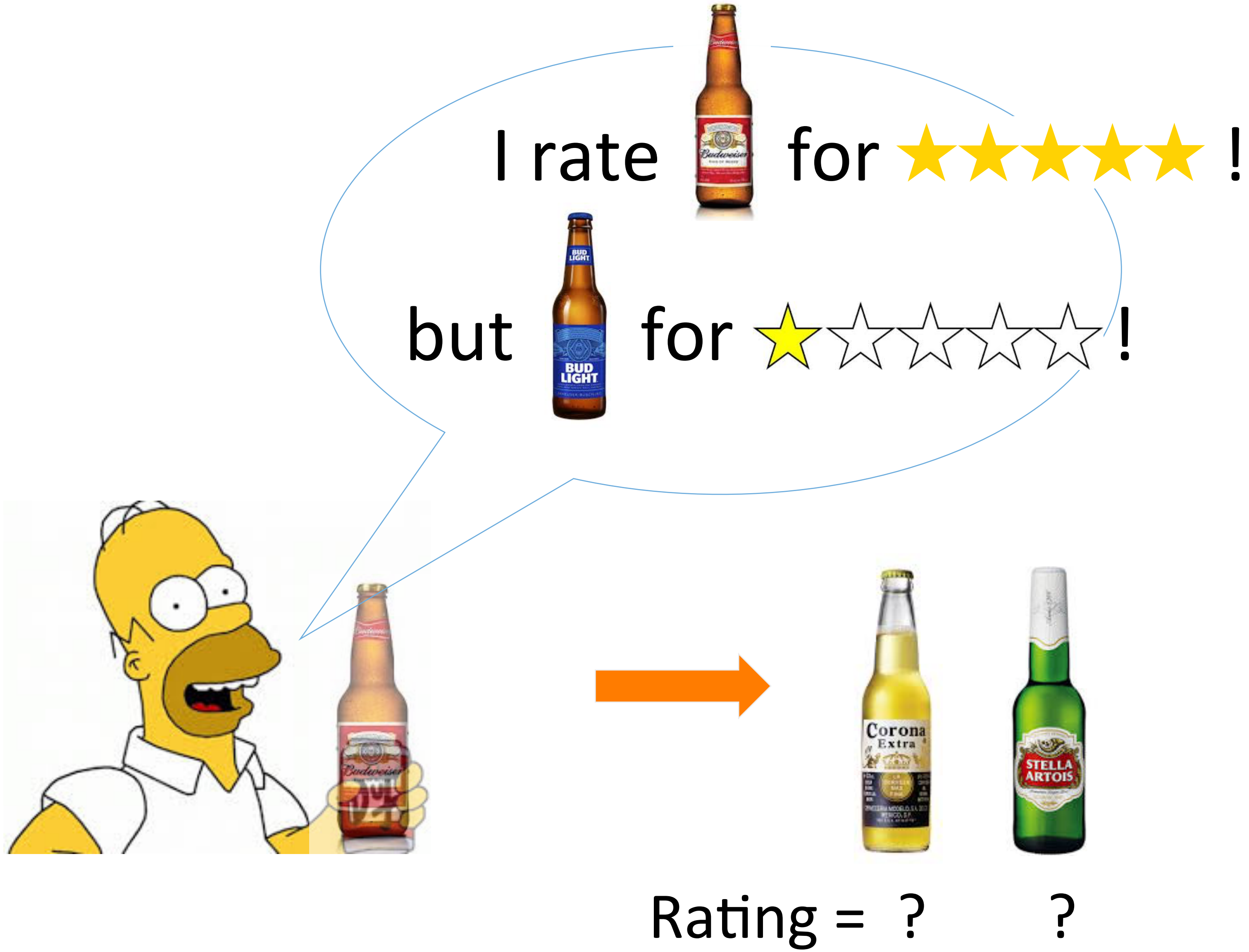
- Always give good recommendation

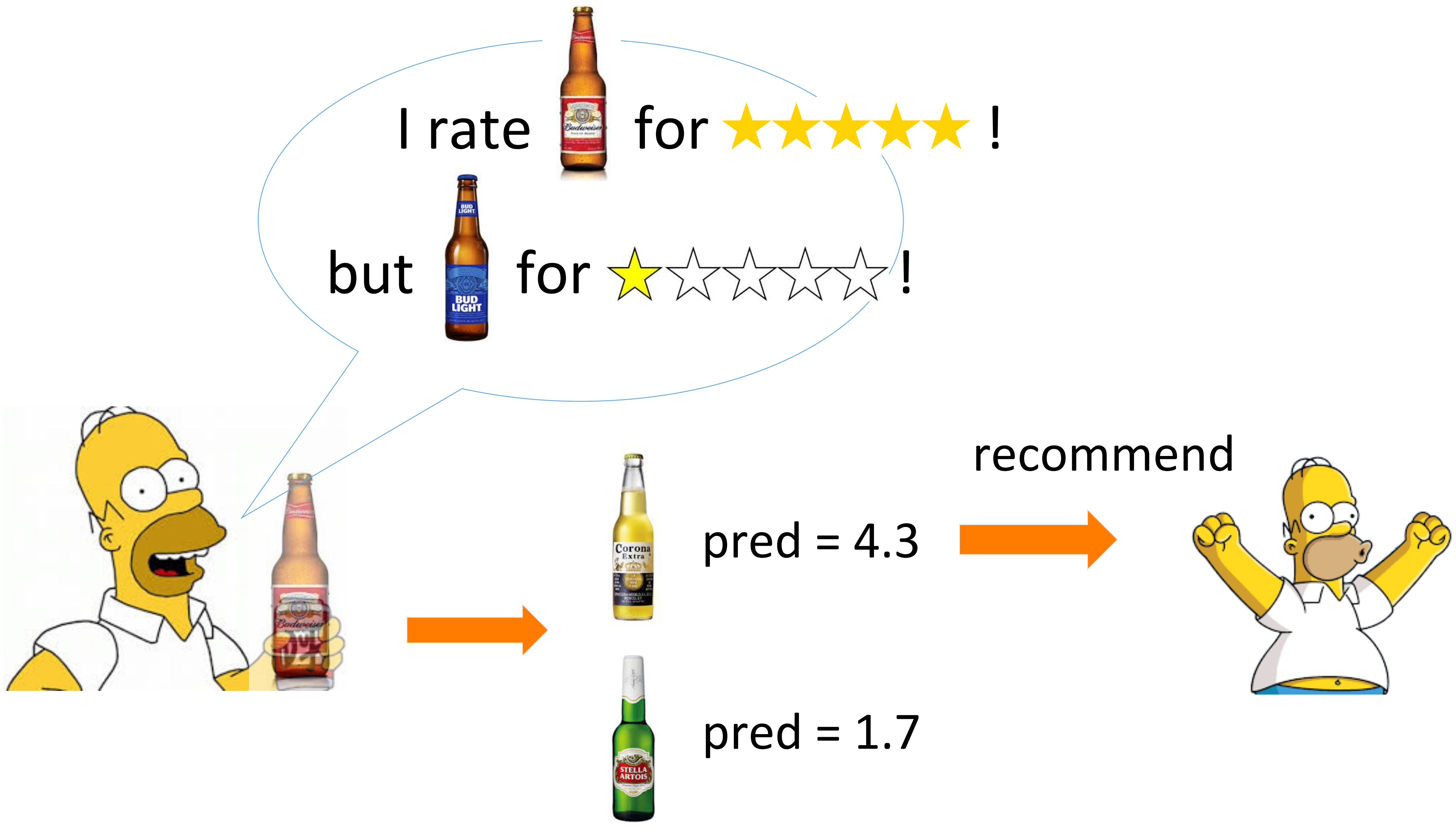
3 February, 2012

Recommendations help drive 27.9% holiday sales growth at John Lewis

By David Moth @ Econsultancy

- Understand customers' demographic characteristics


I rate for !
but for !
Rating = ? ?



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

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

  Collaborative filtering (Netflix, Spotify)

Recommendation Engine UI

When Mr. Simpsons (id=7) is browsing the web:

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 Mr. Simpsons (id=7) is browsing the web:

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.
List Price: $105.00 Save:
Want it tomorrow, Oct. 29? Order within 3 hrs 1 min and choose Saturday Delivery at checkout. Details
Add to Cart
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

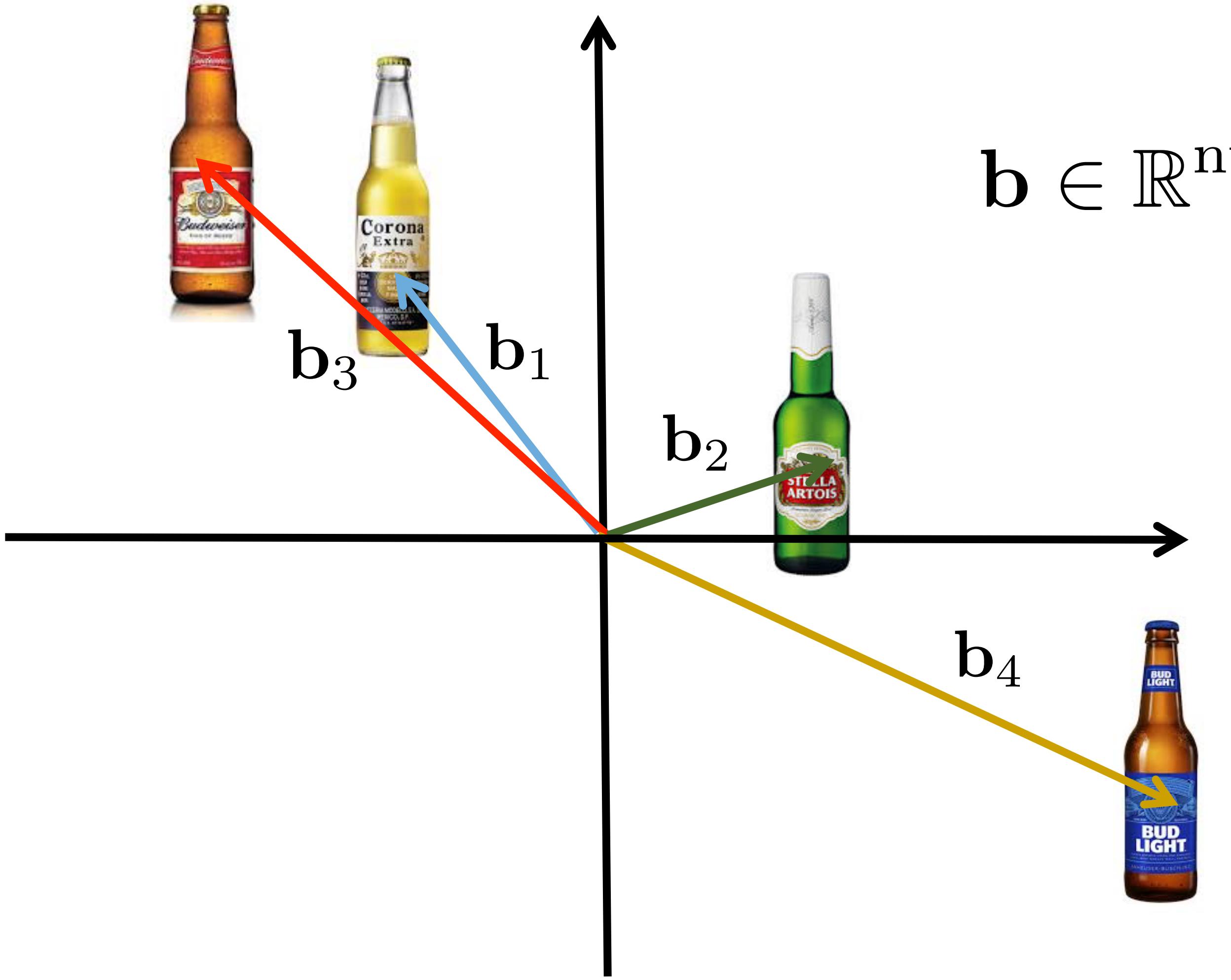
Beer id

Beer vectors are represented by features:

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

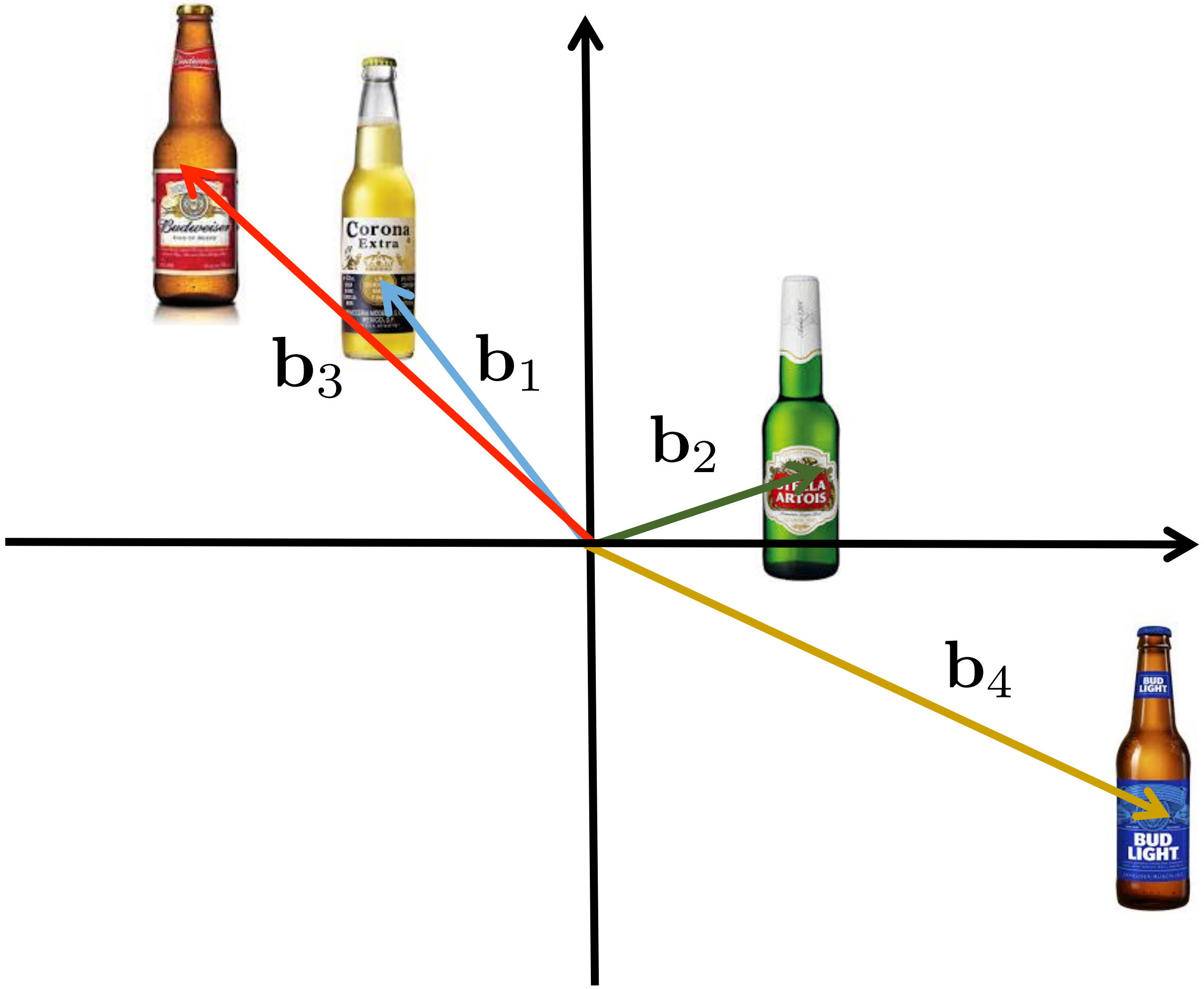
$$\mathbf{b}_2 = (131, 4.2, 4.22, 4.27, 4.1, 0, 0, \cdots)$$

Beer Vector Space



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

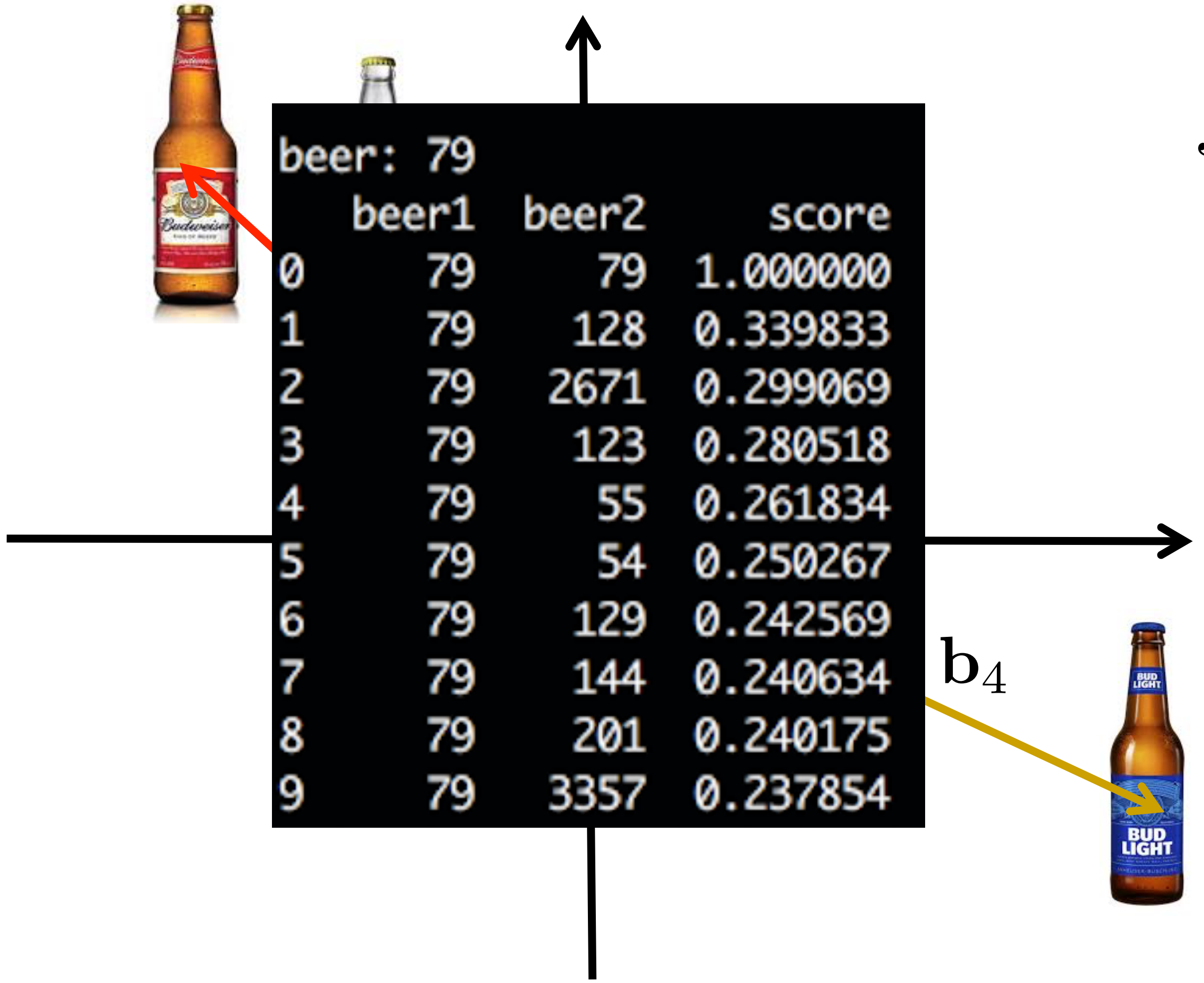
Cosine Similarity



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

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

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

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

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


is more similar

to  than 

Recommendation Engine UI

When Mr. Simpsons (id=7) is browsing the web:

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

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				

Ratings as Features of Beer Vectors

beers

users

	5	6	7	8	9	10	11	12
7	5	4		1	5	5	4	
33			1			1		
34	1	1		5	1	2	2	
42			3					3
45	4	5		1	5	5	3	1
47					4	4		3
48			4	4		1		

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

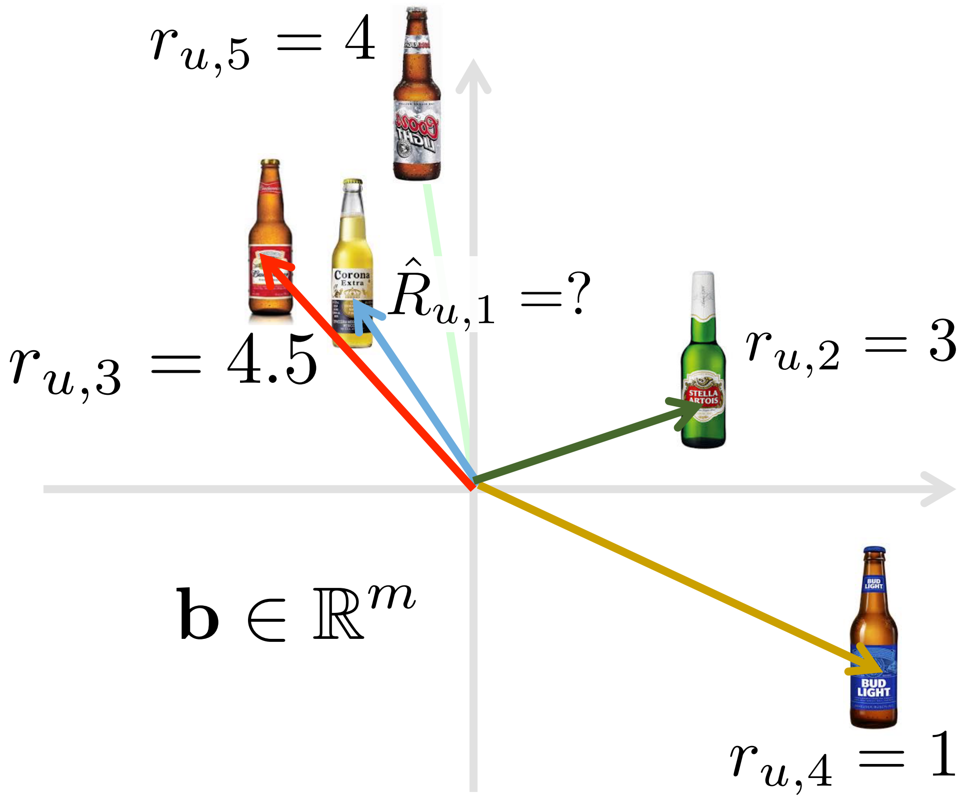
$$\mathbf{b}_9 = (5, -, 1, -, 5, 4, -)$$

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

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

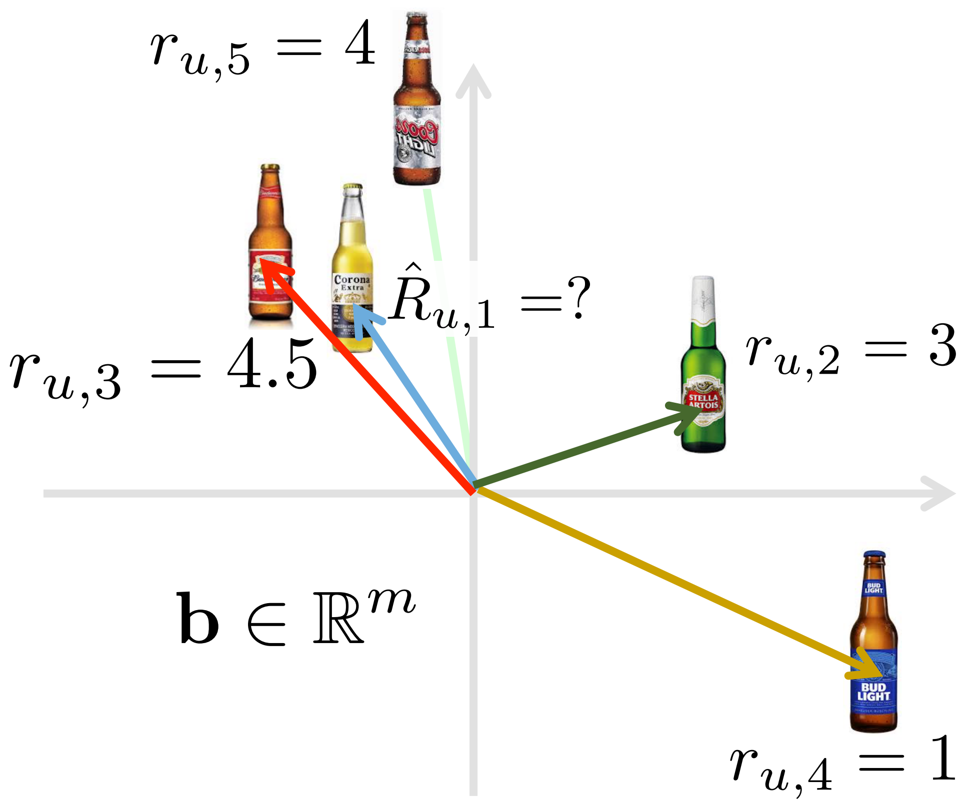
Neighborhood Model (item-based)

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



Neighborhood Model (item-based)

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

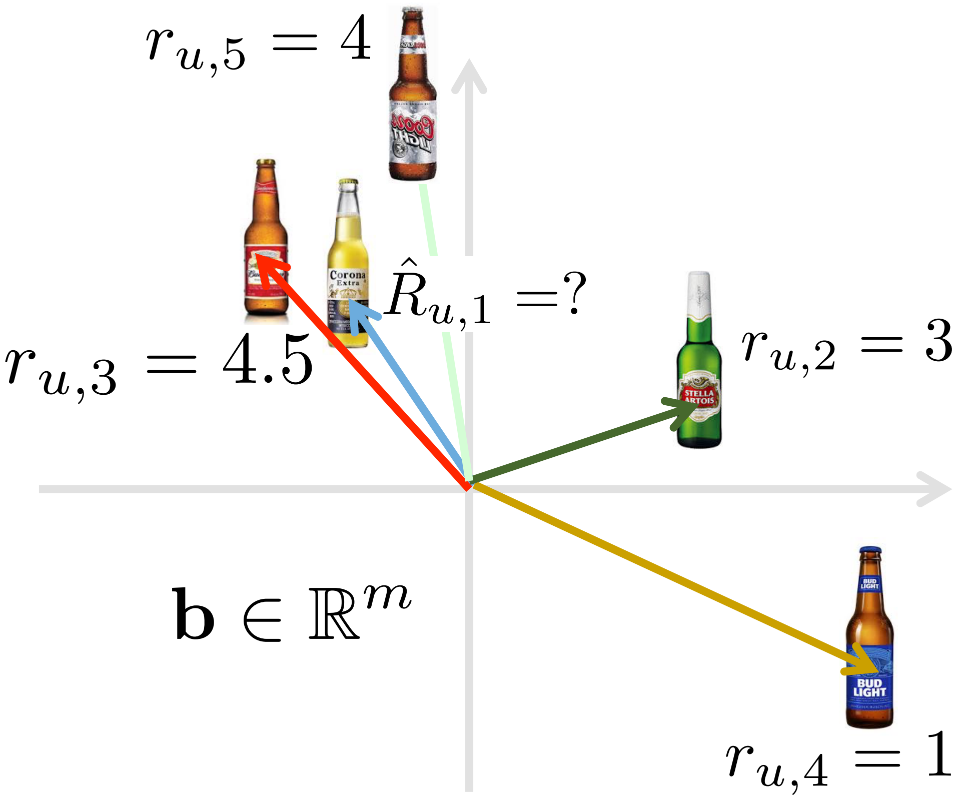


k-nearest-neighbor:

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

Neighborhood Model (item-based)

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



Model = kNN + weight:

weight

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

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

Ratings as Features of Users 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
47					4	4		3
48			4	4		1		

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

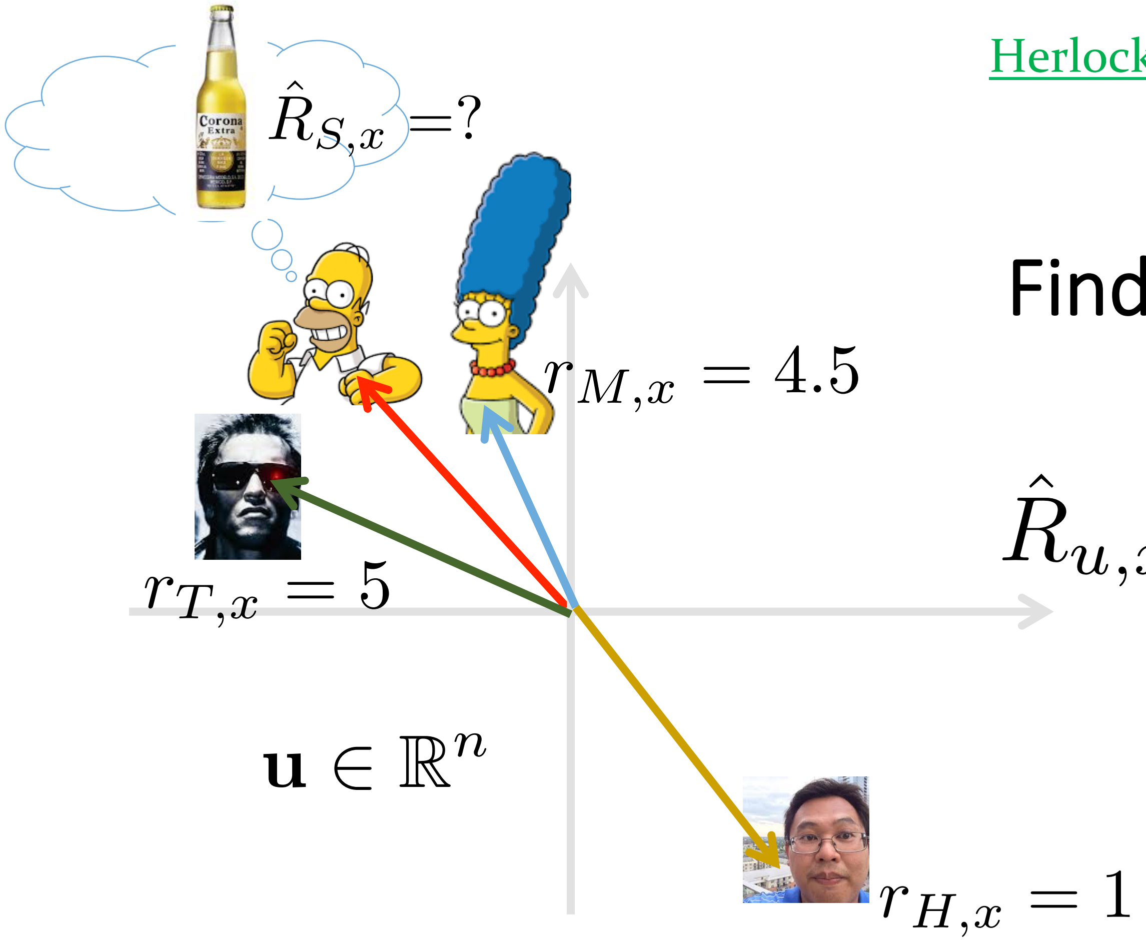
$$\mathbf{u}_{34} = (1, 1, -, 5, 1, 2, 2, -)$$

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

$$s_{45,7} > s_{45,34}$$

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)

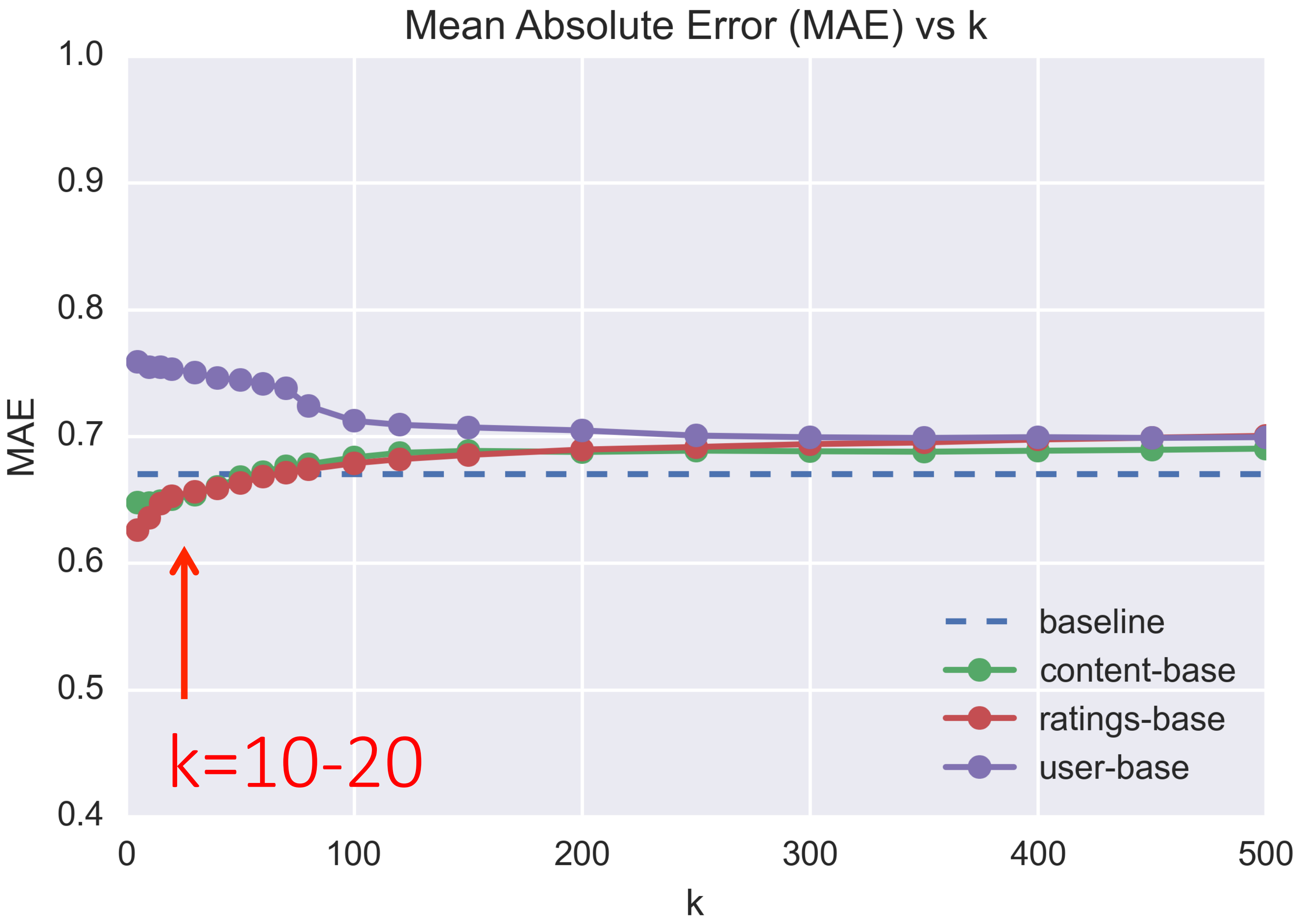
a) item-based:

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

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

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

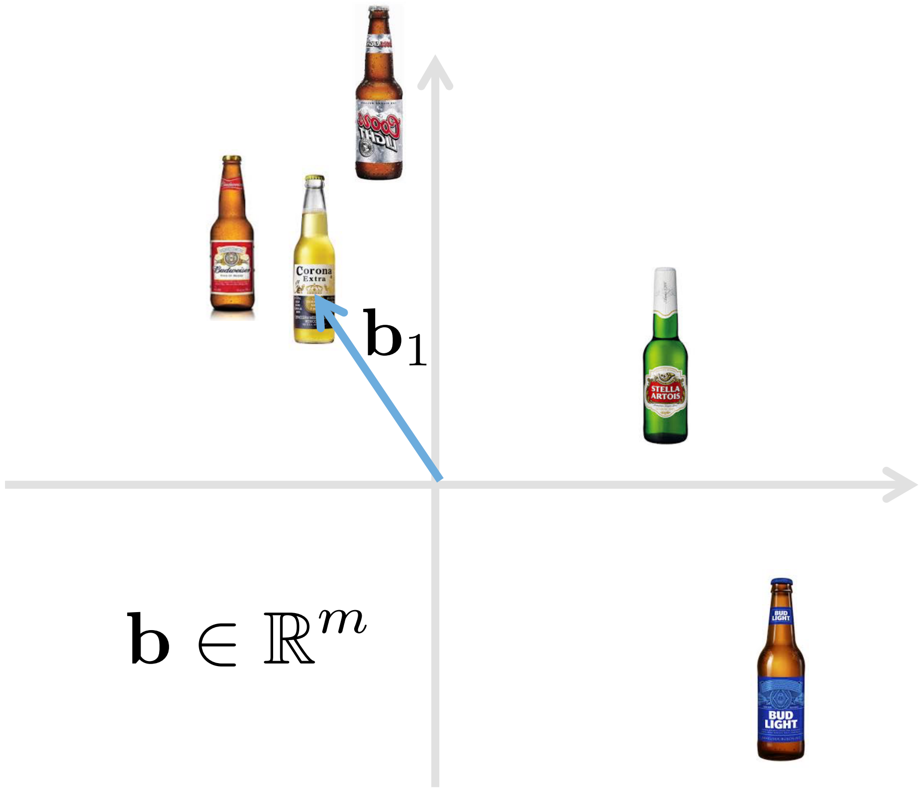
Model Performance

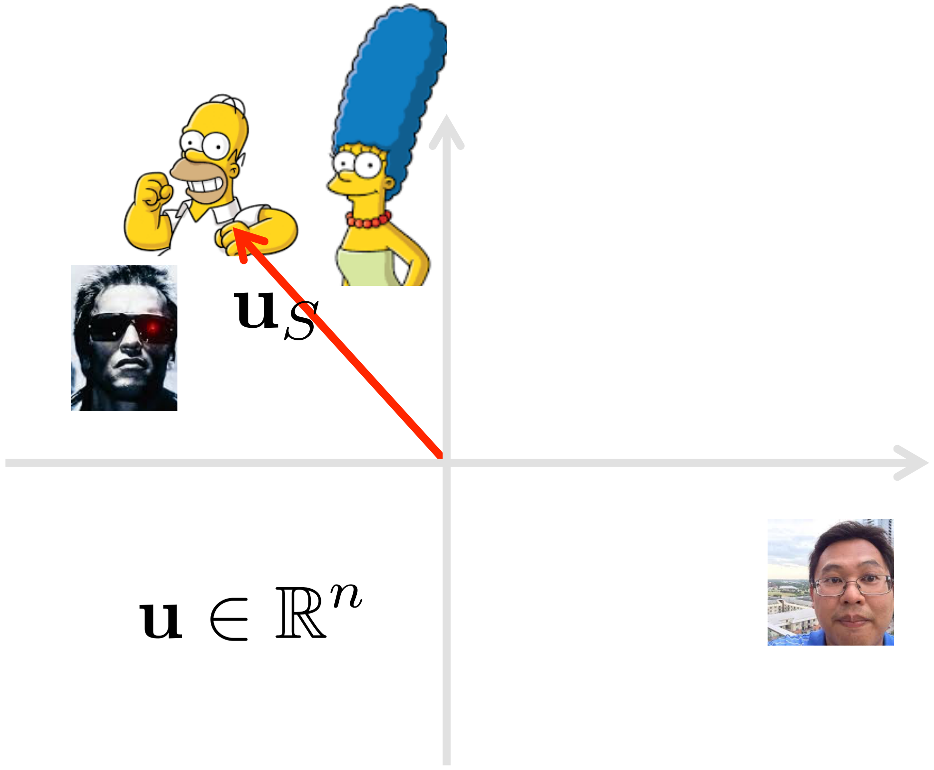


Disadvantages of the Model

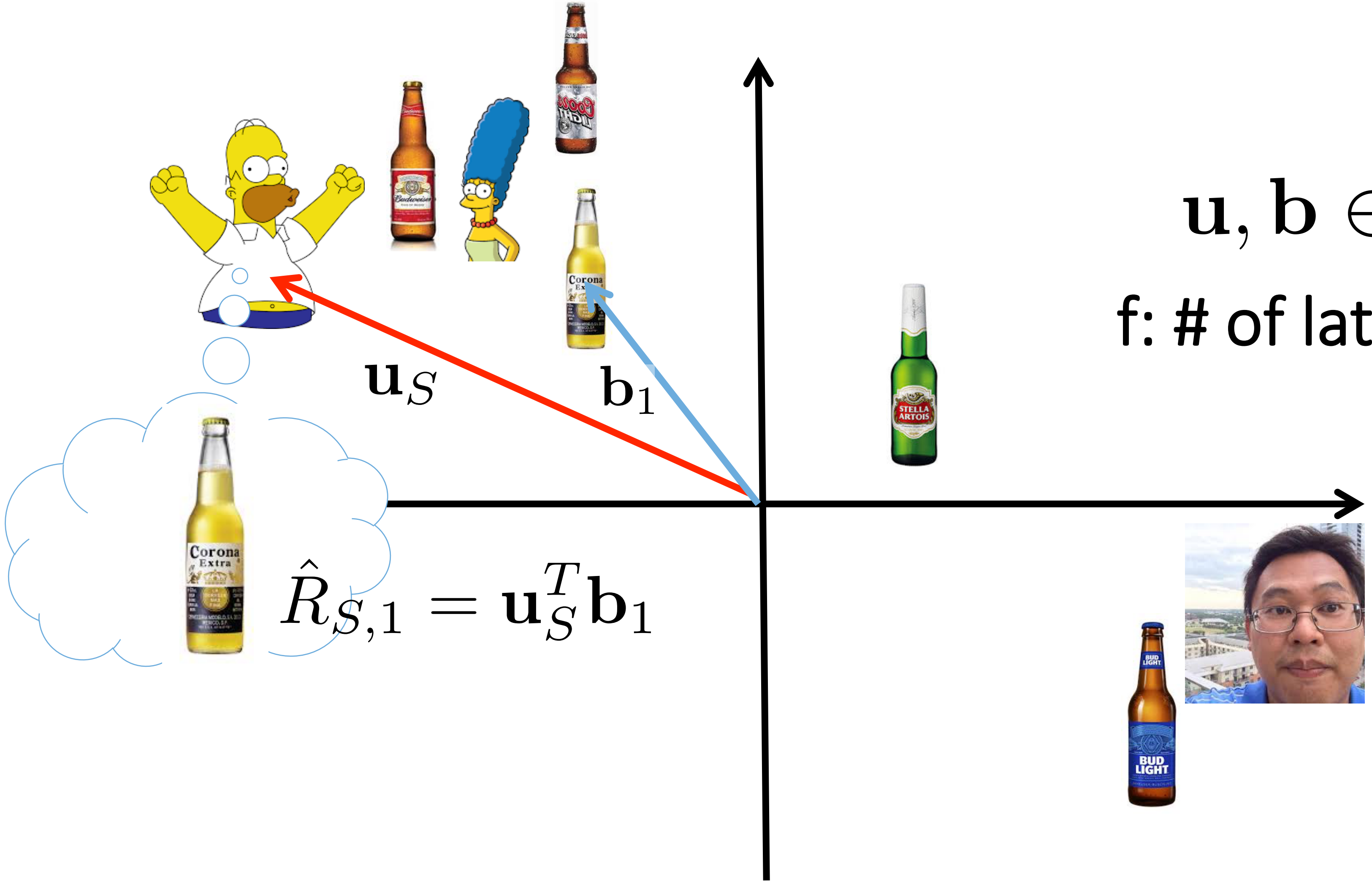
- NOT scalable.
- Not easy to incorporate additional information, e.g. purchase, browse history (implicit data).
- Alternative: **latent-factor model**!

Vector Spaces





Customer-Beer Vector Space

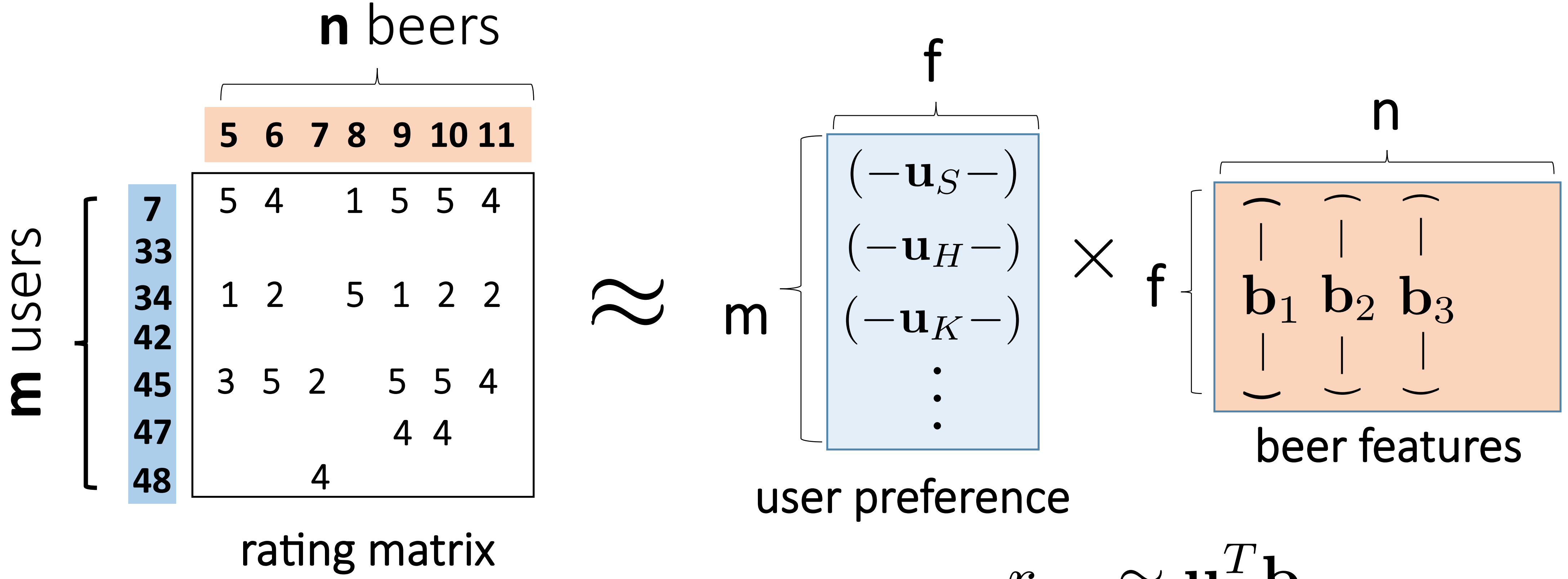


$\mathbf{u}, \mathbf{b} \in \mathbb{R}^f$

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?

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

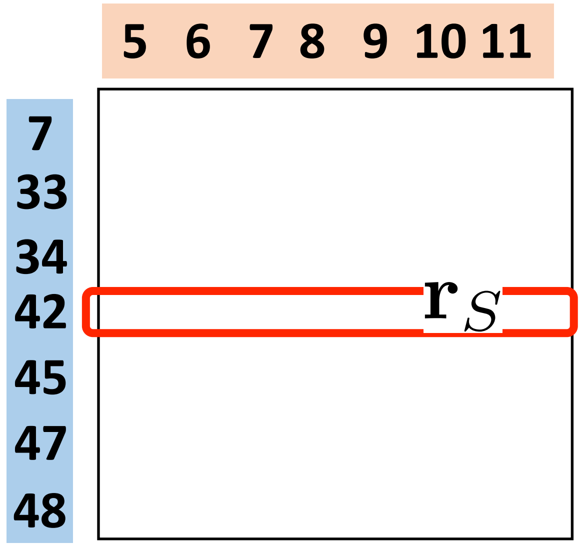
- At each step, fix one variable, and solve minimization:

  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

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

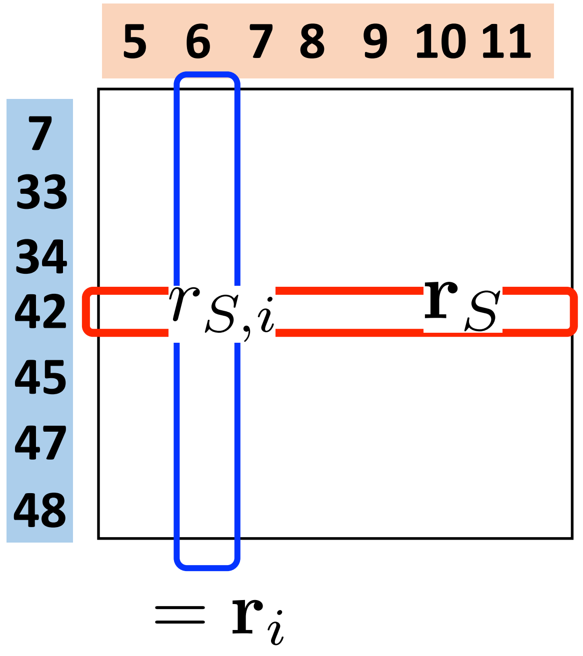


More Detail: Normal Equations

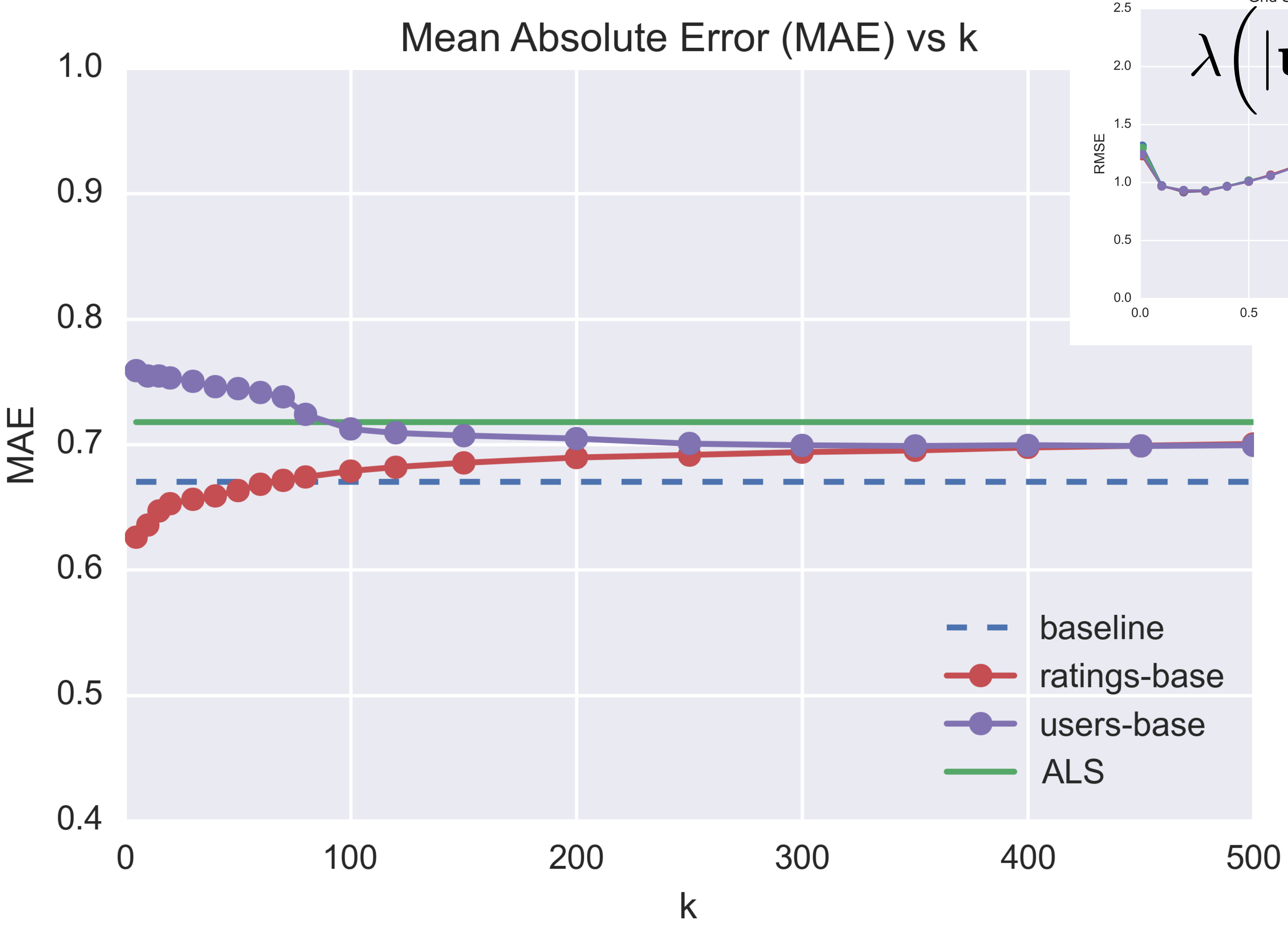
Hu, Koren and Volinsky, 2008

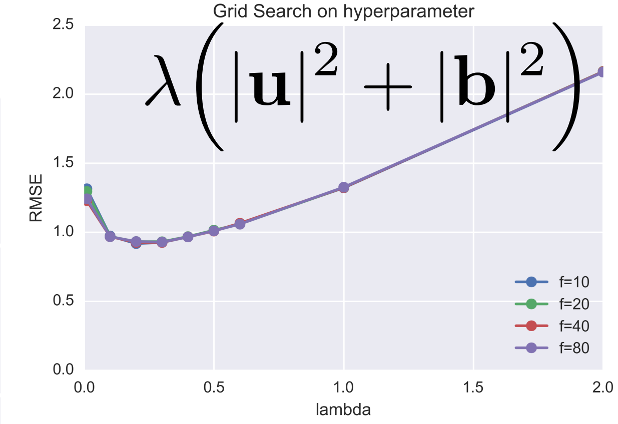
$$\mathbf{u}_S = \begin{pmatrix} u_{S,1} \\ u_{S,2} \\ \vdots \\ u_{S,f} \end{pmatrix} = \left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I}\right)^{-1}\mathbf{B}^T \begin{pmatrix} r_{S,1} \\ r_{S,2} \\ \vdots \\ r_{S,n} \end{pmatrix}$$

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



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