Course Update

Out

Current Plan (updated)

Due

- HW 8 (online)
- Mini-project proposal
- HW 9 (online)
- HW 10 (written/prog)
- Midterm 2
- Mini-project

Sun	Mon	Tue	Wed	Thu	Fri	Sat
2	3	4	5 HW8 Proj	6	7	8
9 HW8	10 HW9 HW10	11	12	13	14	15
16	17 HW9	18	19 Prop	20	21	22 HW10
23	24	25	26 MT2	27	28	29
30	1	2	3	4	5 Proj	6

Plan

Last time

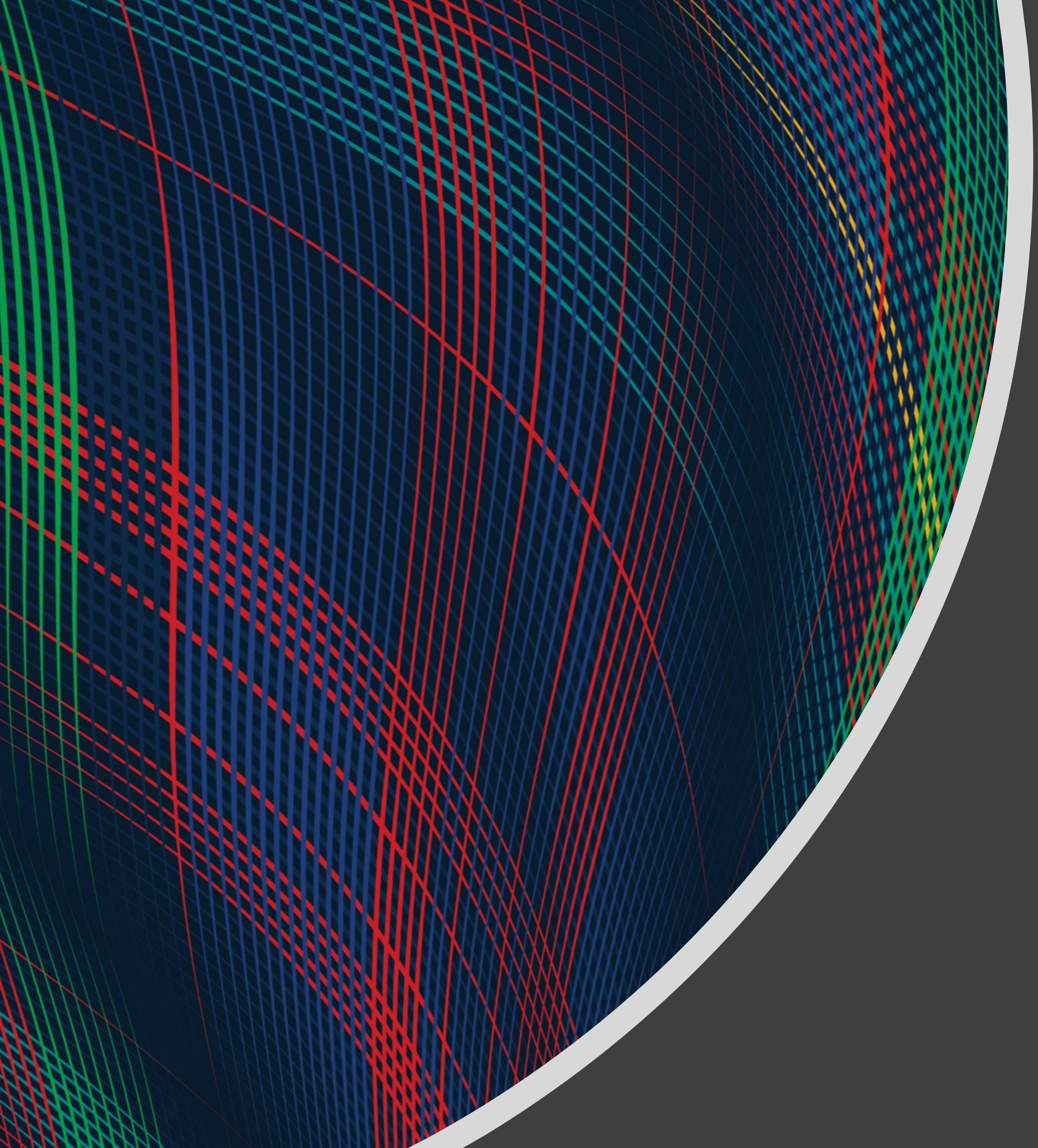
- Unsupervised Learning: Dimensionality Reduction

Today

- Recommender Systems
- Unsupervised Learning: Clustering
  - K-means

Next time

- Unsupervised Learning: Clustering
  - Gaussian mixture models and expectation maximization



10-315 Introduction to ML

Recommender Systems

Instructor: Pat Virtue

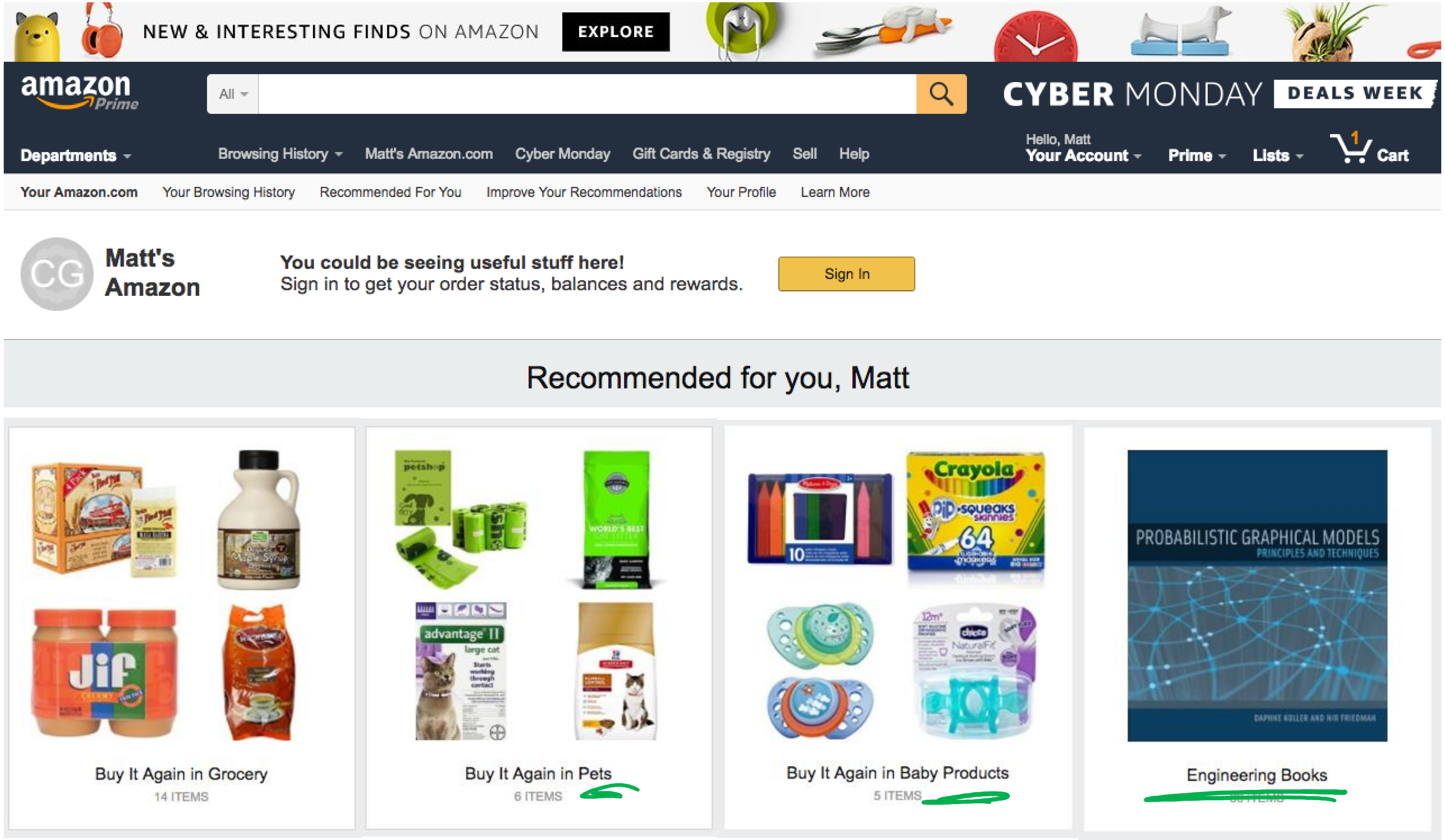
Recommender Systems

A Common Challenge:

- Assume you're a company selling items of some sort: movies, songs, products, etc.
- Company collects millions of ratings from users of their items
- To maximize profit / user happiness, you want to recommend items that users are likely to want

Slide credit: CMU MLD Matt Gormley

Recommender Systems



NEW & INTERESTING FINDS ON AMAZON EXPLORE

Matt's Amazon

You could be seeing useful stuff here!
Sign in to get your order status, balances and rewards.

Sign In

Recommended for you, Matt

Buy It Again in Grocery
14 ITEMS

Buy It Again in Pets
6 ITEMS

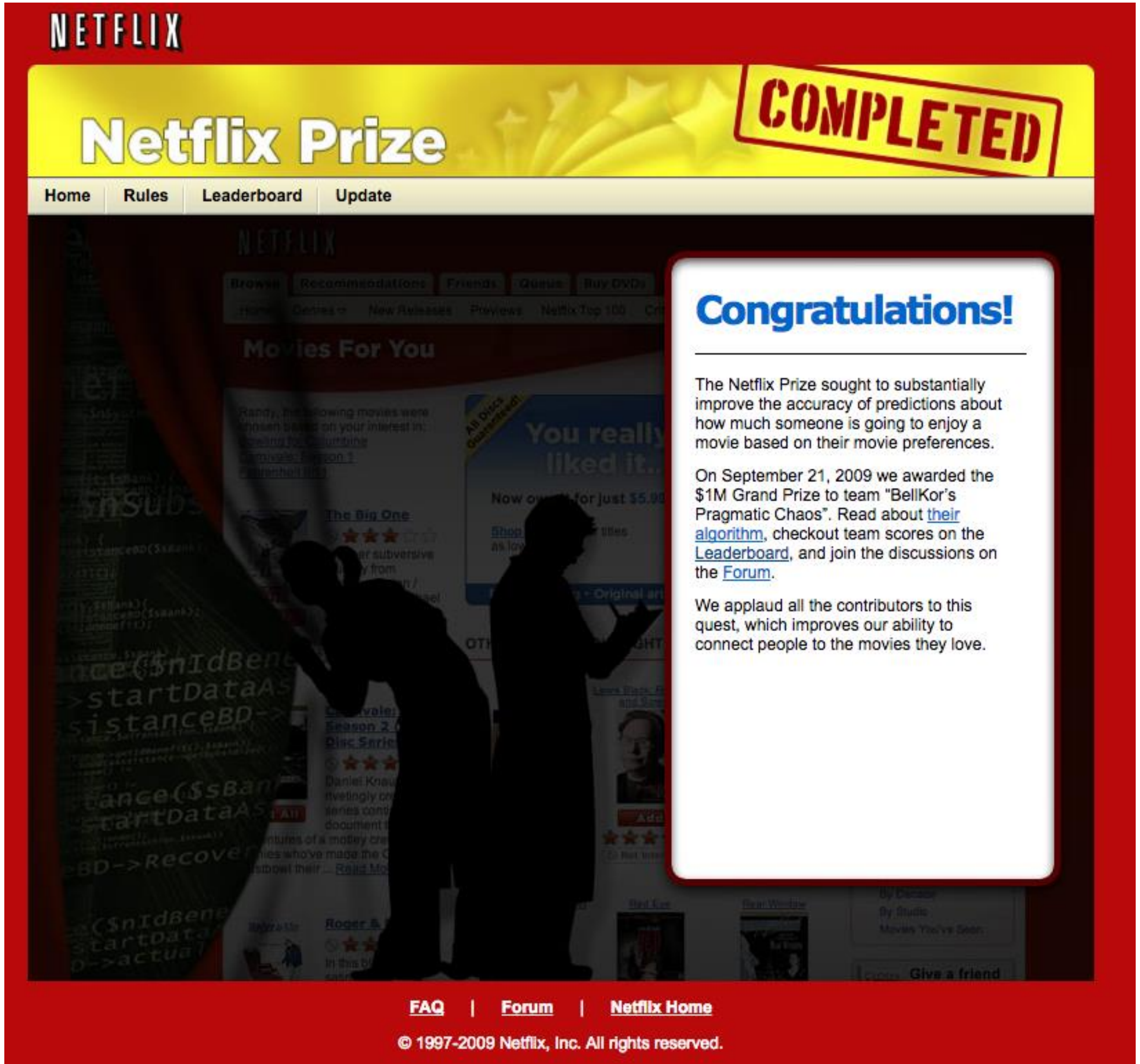
Buy It Again in Baby Products
5 ITEMS

Engineering Books

PROBABILISTIC GRAPHICAL MODELS
PRINCIPLES AND TECHNIQUES
DAPHNE KOLLER AND NIR FRIEDMAN
Crayola
64

Slide credit: CMU MLD Matt Gormley

Recommender Systems



Congratulations!

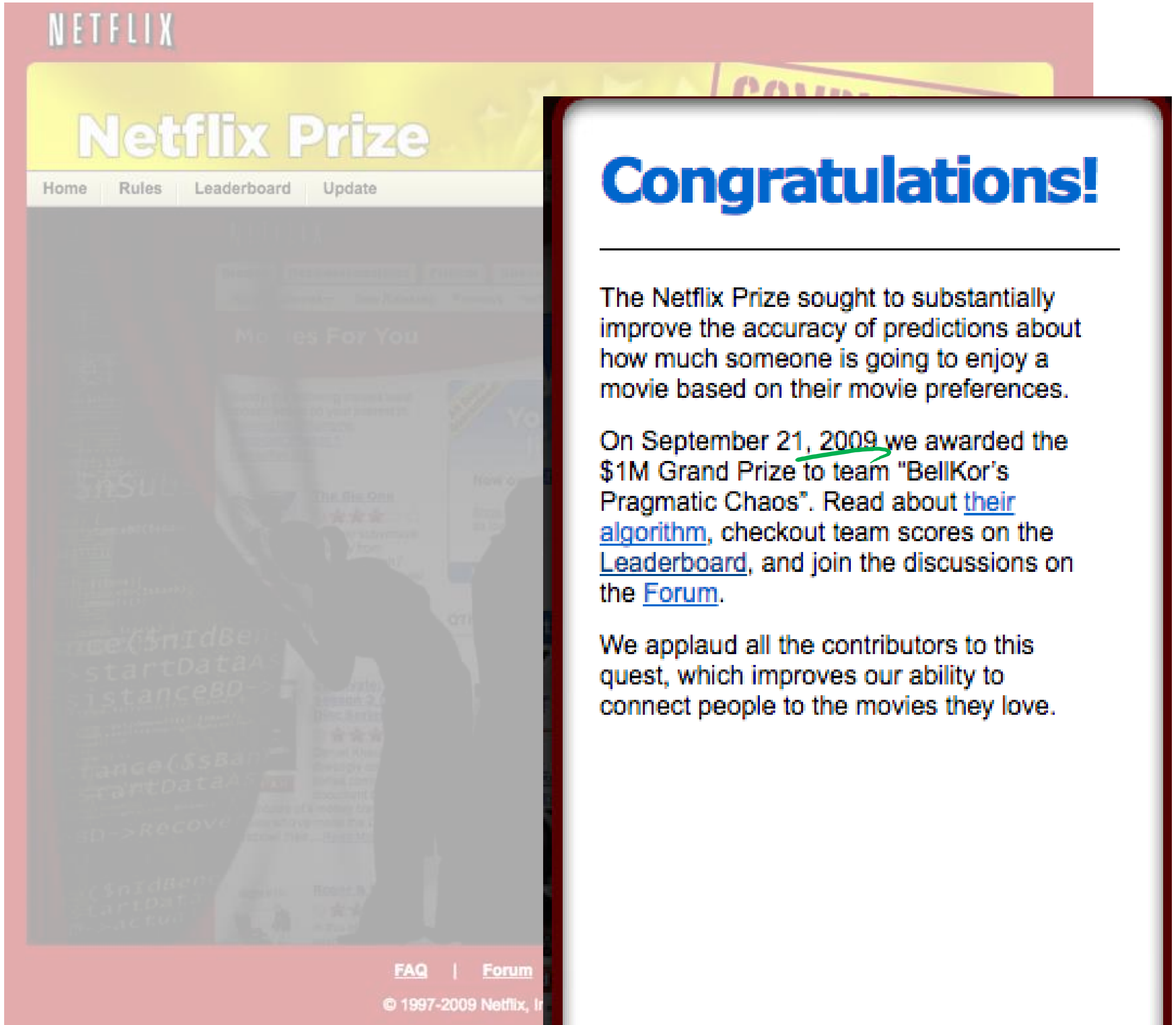
The Netflix Prize sought to substantially improve the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences.

On September 21, 2009 we awarded the $1M Grand Prize to team "BellKor's Pragmatic Chaos". Read about their algorithm, checkout team scores on the Leaderboard, and join the discussions on the Forum.

We applaud all the contributors to this quest, which improves our ability to connect people to the movies they love.

Slide credit: CMU MLD Matt Gormley

Recommender Systems



Congratulations!

The Netflix Prize sought to substantially improve the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences.

On September 21, 2009 we awarded the $1M Grand Prize to team "BellKor's Pragmatic Chaos". Read about their algorithm, checkout team scores on the Leaderboard, and join the discussions on the Forum.

We applaud all the contributors to this quest, which improves our ability to connect people to the movies they love.

© 1997-2009 Netflix, Ir

$1M

Slide credit: CMU MLD Matt Gormley

AI System Design: Movie Recommendation

Task

$h(\text{input}) \to \text{output}$

user item

rating

1 - 5

Experience

Data (training)

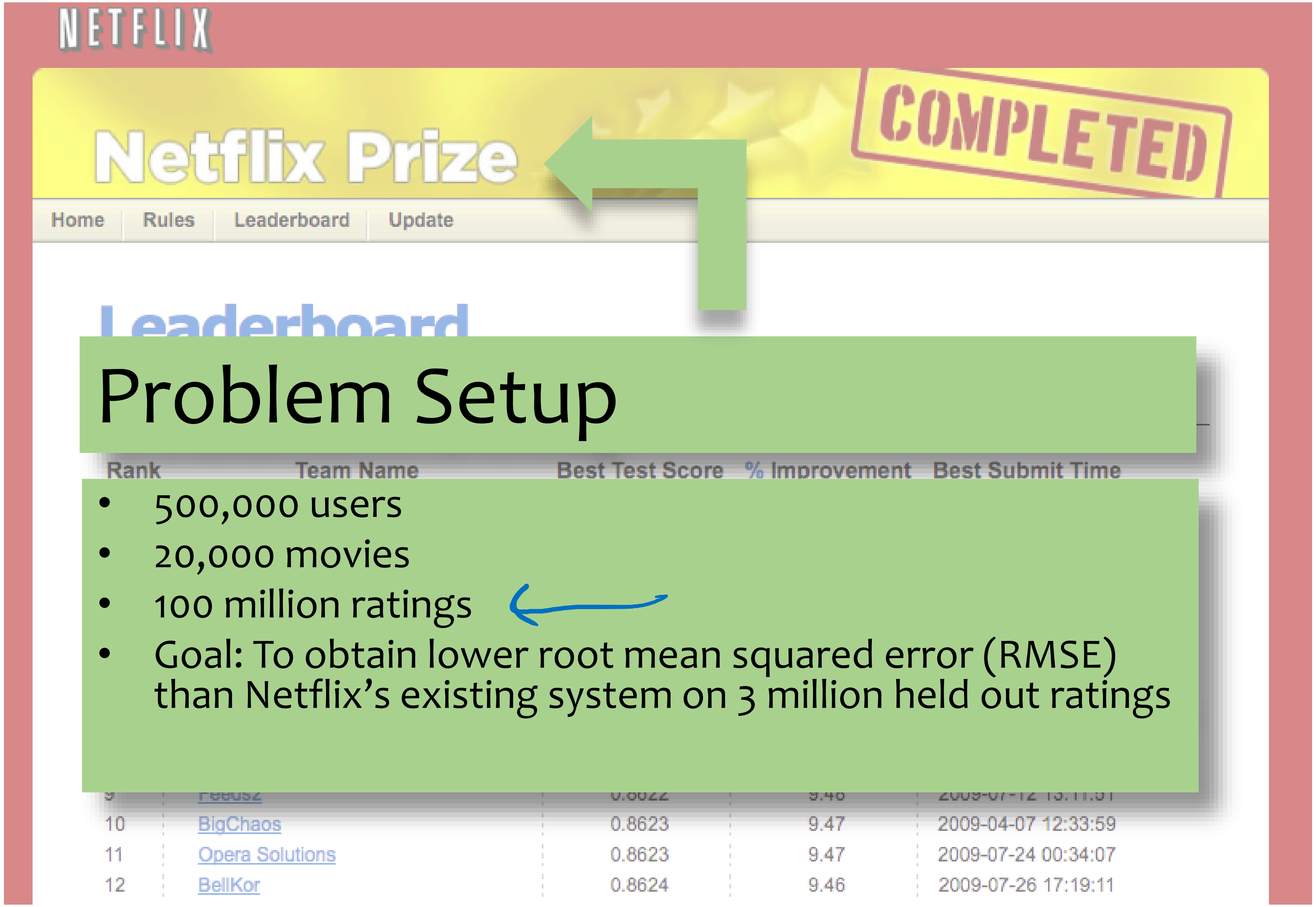
N (i, j, r)

Performance measure

output

$\hat{y}$ y

Recommender Systems



Problem Setup

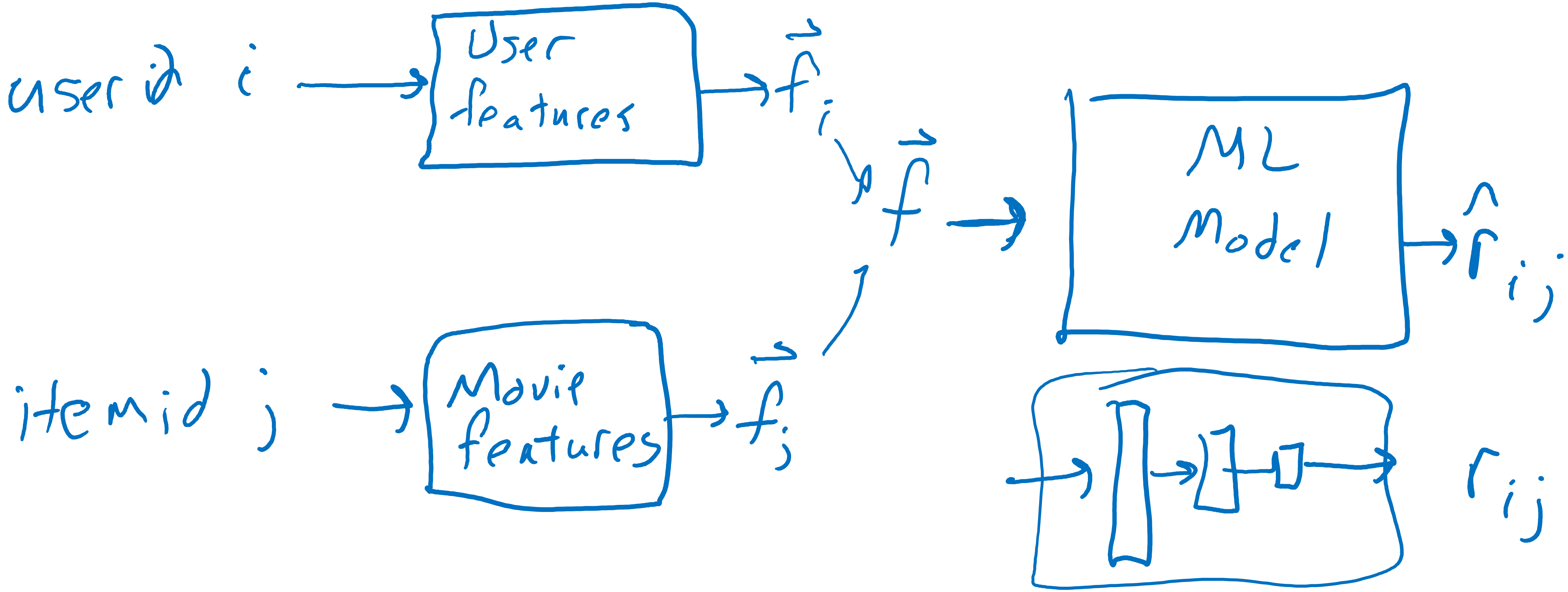
- 500,000 users
- 20,000 movies
- 100 million ratings
- Goal: To obtain lower root mean squared error (RMSE) than Netflix's existing system on 3 million held out ratings

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
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

Slide credit: CMU MLD Matt Gormley

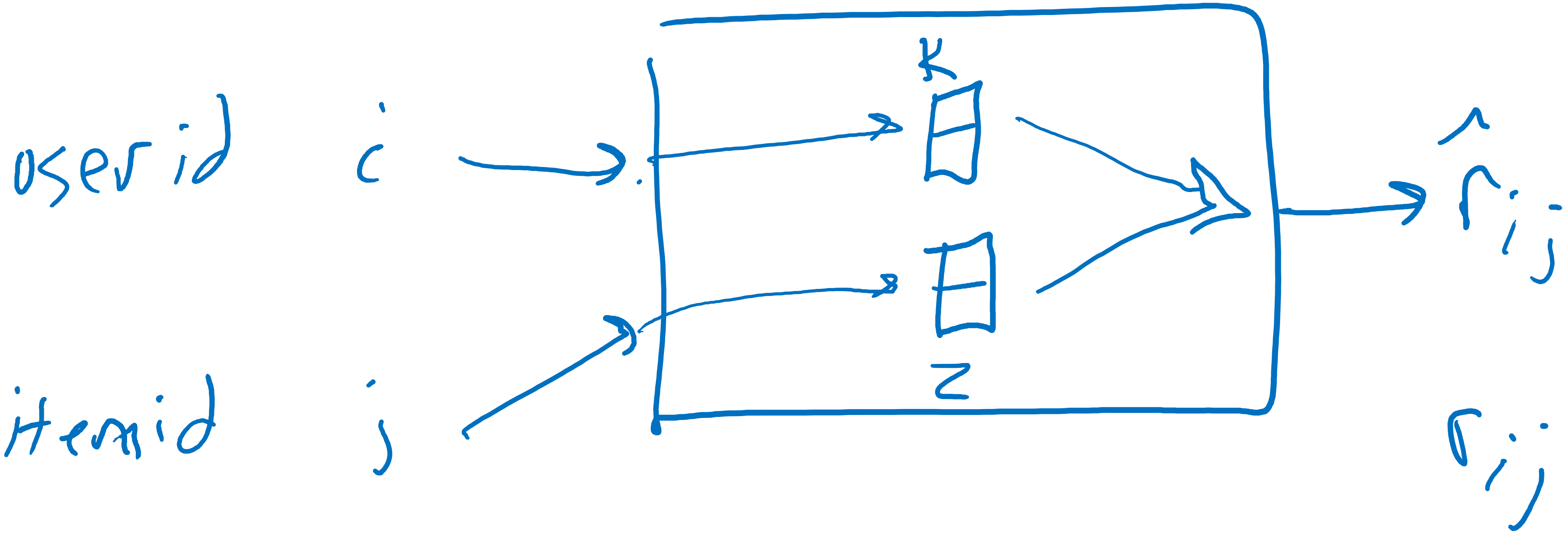
ML System Design: Movie Recommendation

Model



ML System Design: Movie Recommendation

Model



Recommender Systems

Setup:

- Items:
  movies, songs, products, etc.
  (often many thousands)
- Users:
  watchers, listeners, purchasers, etc.
  (often many millions)
- Feedback:
  5-star ratings, not-clicking ‘next’, purchases, etc.

Challenge:

- Users only rate a small number of items
  (the user/item rating data is sparse)

	Doctor Strange	Star Trek: Beyond	Zootopia
Alita	1		5
BB-8	3	4	
C-3P0	3	5	2
Pat			5

Slide credit: CMU MLD Matt Gormley

Different Approaches

Item-based (*Content filtering*)

- Features about each item
- Given an item, other “close” items have similar values
- e.g. Pandora.com, music genome project

Different Approaches

Item-based (*Content filtering*)

- Features about each item
- Given an item, other “close” items have similar values
- e.g. Pandora.com, music genome project

User-based

- Features about each user
- Given a user, other “close” users have similar preferences
- *Market segmentation*

Learning user-item relationship

- Can be done without features on either user or item
- Collaborative filtering techniques

Collaborative Filtering

Collaborative Filtering

Everyday Examples of Collaborative Filtering...

- Bestseller lists
- Top 40 music lists
- The "recent returns" shelf at the library
- Unmarked but well-used paths thru the woods ←
- The printer room at work
- "Read any good books lately?"
- …

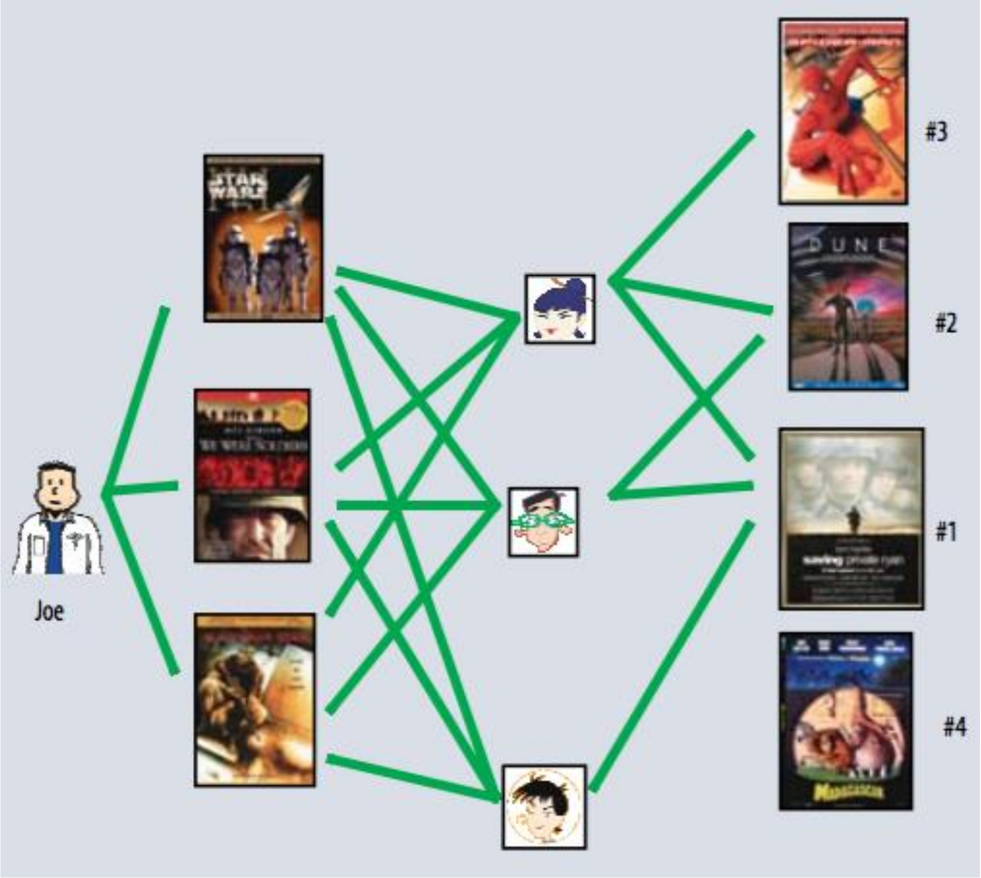
Common insight: personal tastes are correlated

- If Alita and BB-8 both like X and Alita likes Y then BB-8 is more likely to like Y
- especially (perhaps) if BB-8 knows Alita

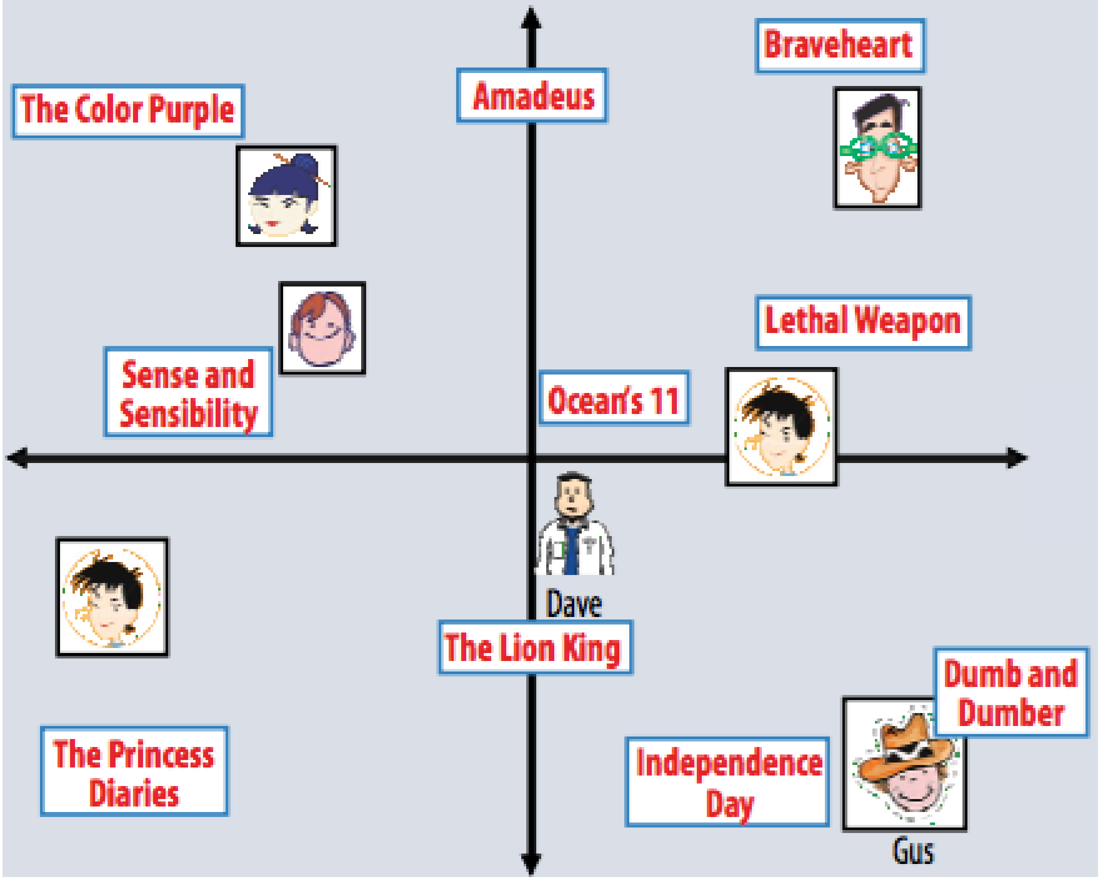
Slide from William Cohen

Two Types of Collaborative Filtering

1. Neighborhood Methods



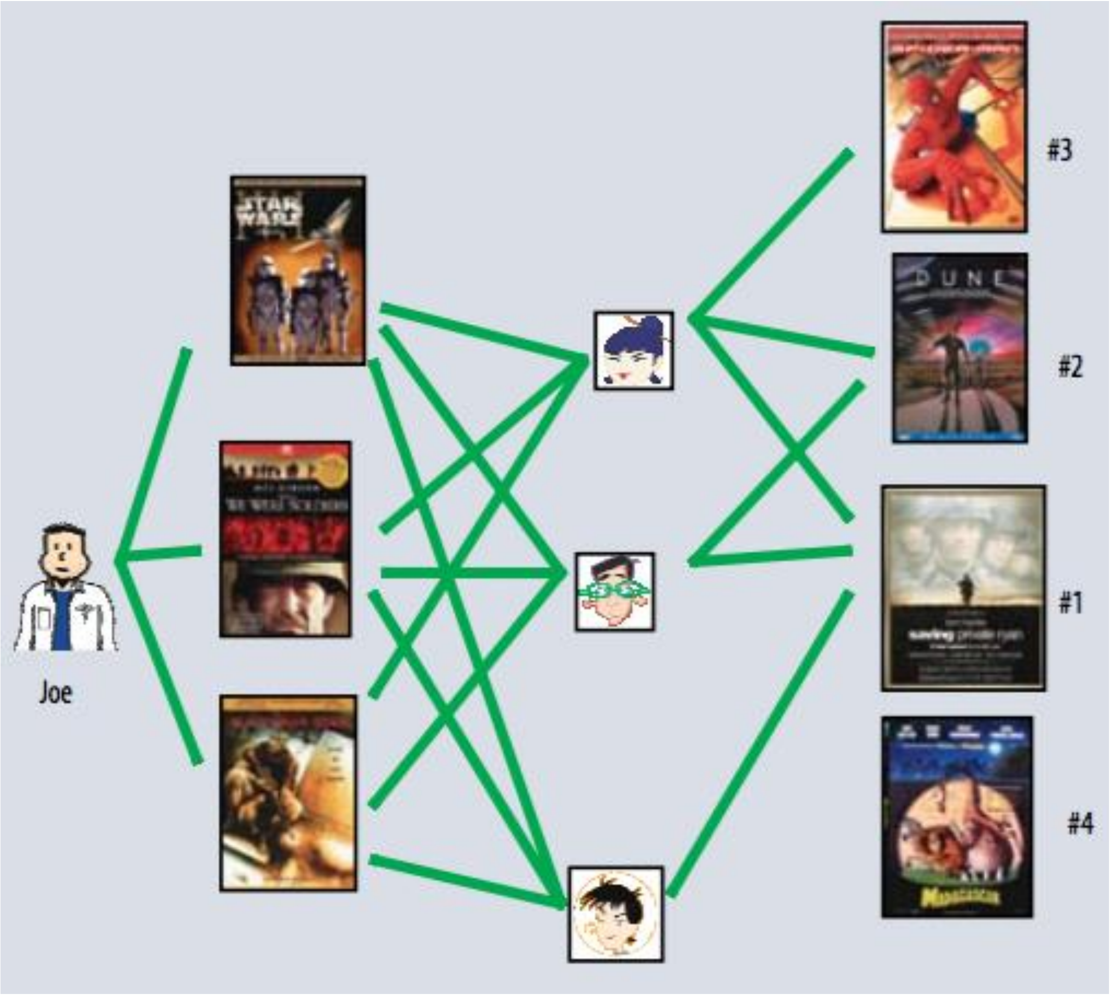
2. Latent Factor Methods



Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

1. Neighborhood Methods



In the figure, assume that a green line indicates the movie was **watched**

Algorithm:

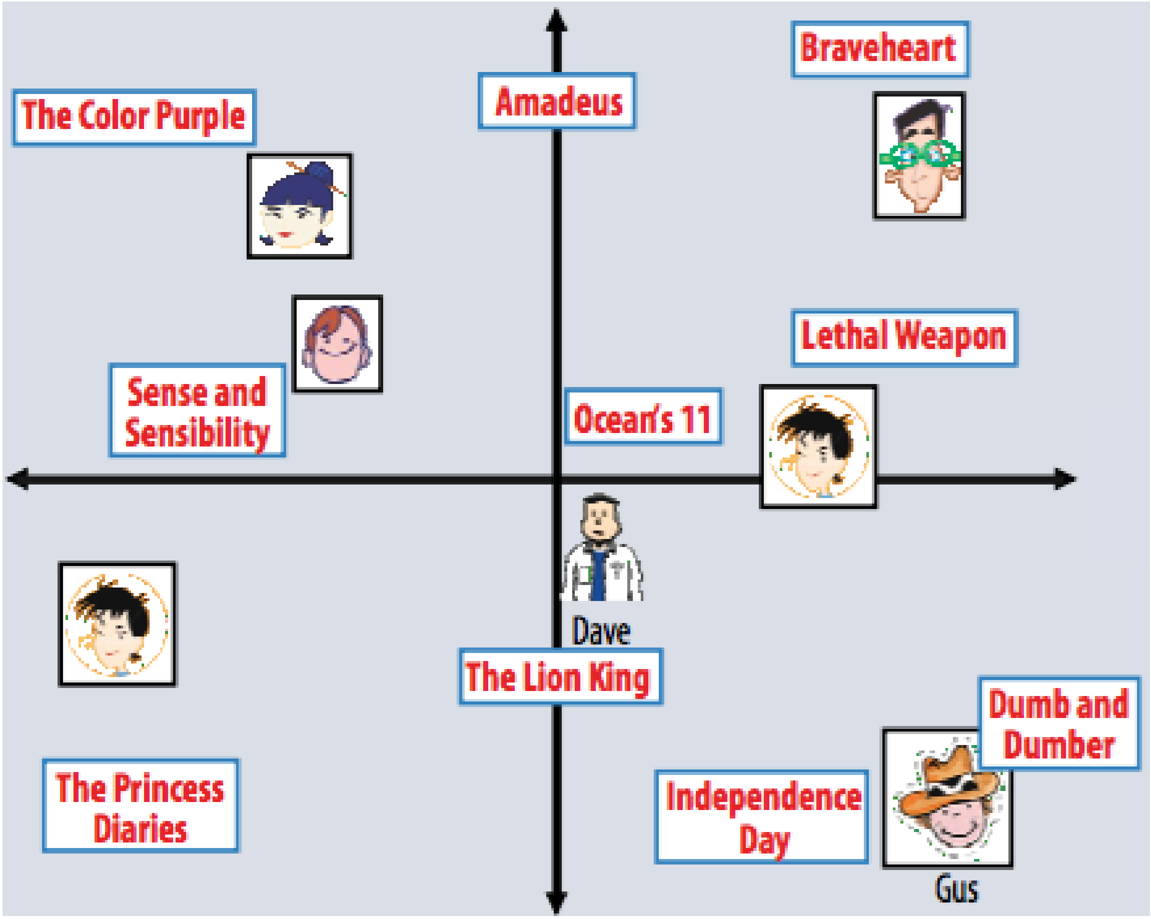
1. **Find neighbors** based on similarity of movie preferences
2. **Recommend** movies that those neighbors watched

Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

2. Latent Factor Methods

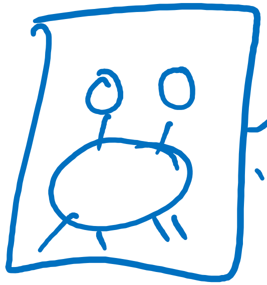
- Assume that both movies and users can be mapped to the same **feature space**
- **Recommend** a movie based on its **proximity** to the user in the feature (latent) space
- **Example algorithm**: Matrix Factorization

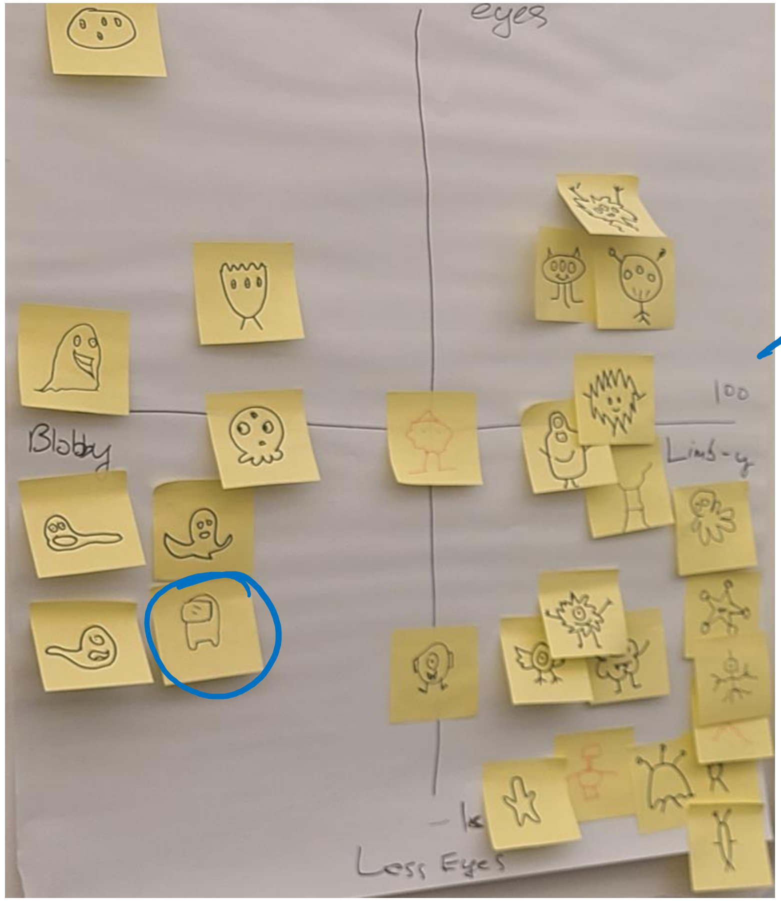


Figures from Koren et al. (2009)

Background: Learn a Feature Space

$\vec{x}$



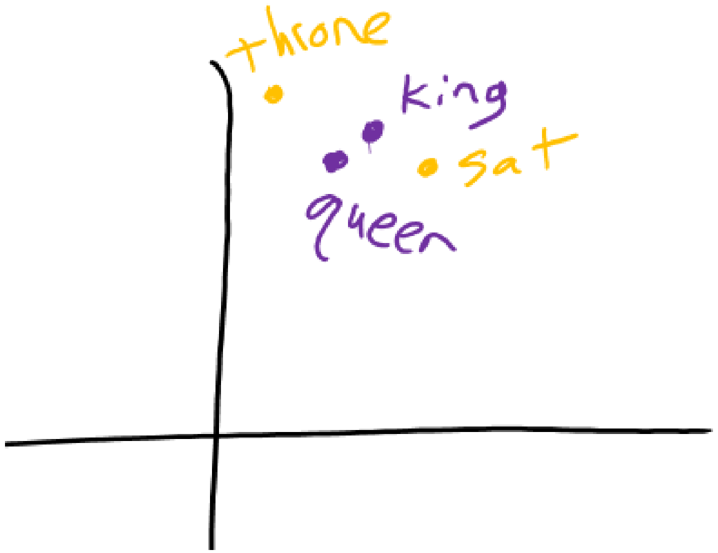


$\vec{z} \in \mathbb{R}^2$

Background: Learn a Feature Space

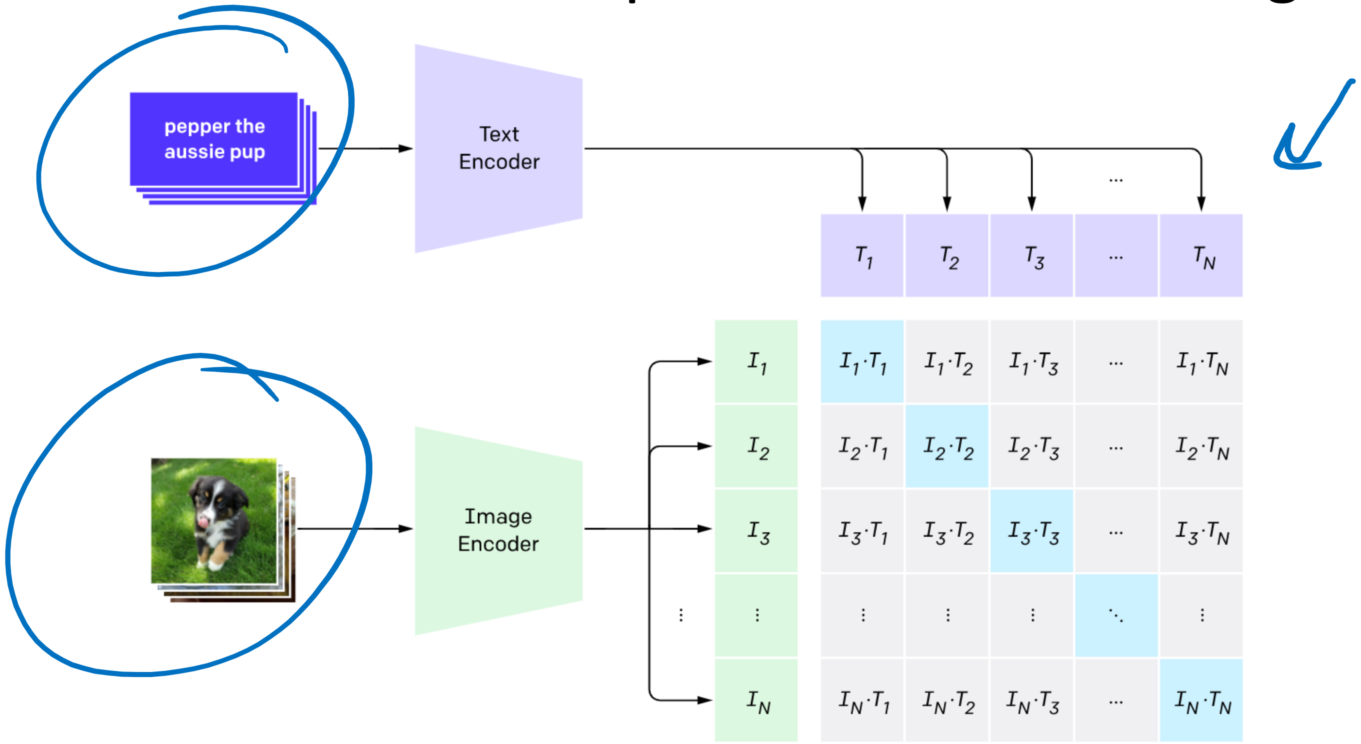
Word2vec:

Feature space for words



CLIP:

Common feature space for text and images

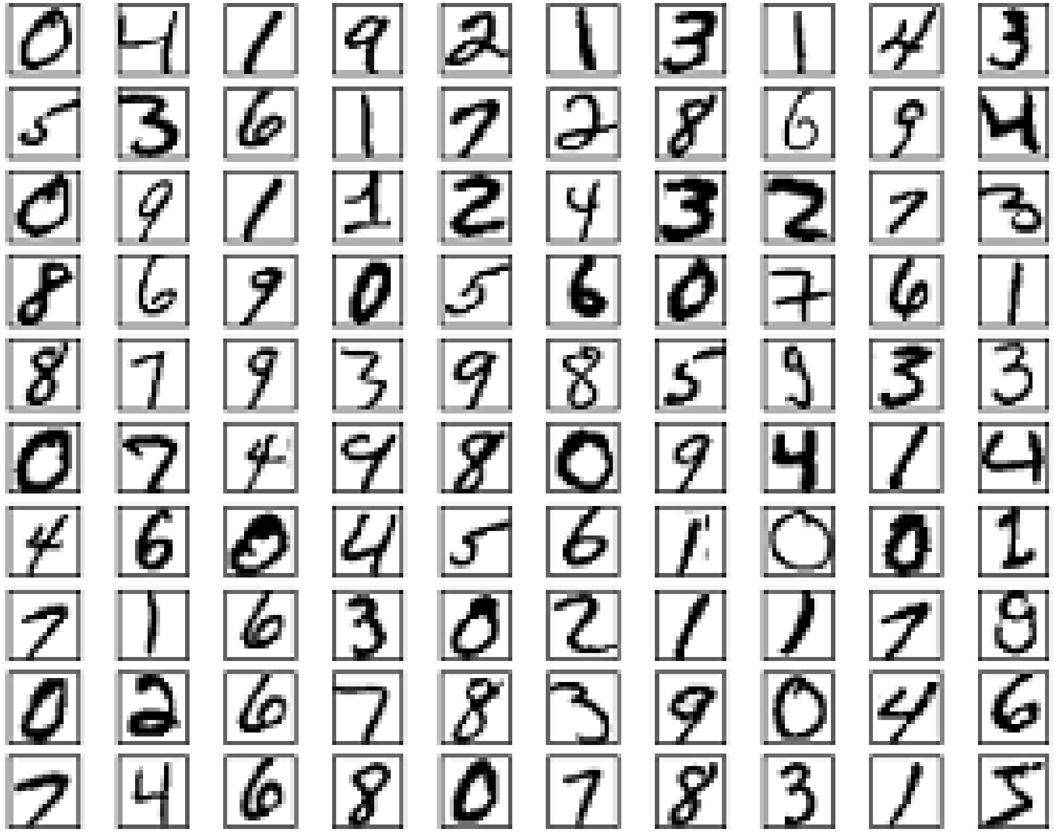


	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Background: Learn a Feature Space

Why might low dimensional embeddings be useful?

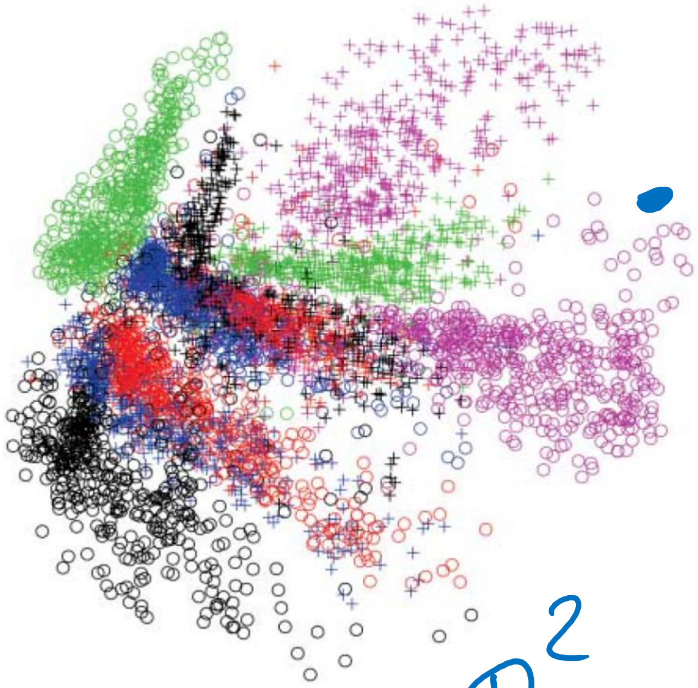
- Example: MNIST digit classification with nearest neighbor



Background: Learn a Feature Space

Why might low dimensional embeddings be useful?

- Example: MNIST digit classification with nearest neighbor



$z \in \mathbb{R}^2$

Image: Hinton & Salakhutdinov. *Science* 313.5786 (2006): 504-507.

Background: Measure of Similarity

$\|u-v\|_2$

We’ve been using Euclidean distance

- $d(\mathbf{u},\mathbf{v}) = \|\mathbf{u}-\mathbf{v}\|_2$

Cosine similarity

- To vectors are similar if the angle between them is small
- $\underset{s}{\cancel{d}}(\mathbf{u},\mathbf{v}) = \mathbf{u}^T\mathbf{v}$

$$= \|\vec{u}\|_2 \|\vec{v}\|_2 \cos\theta$$

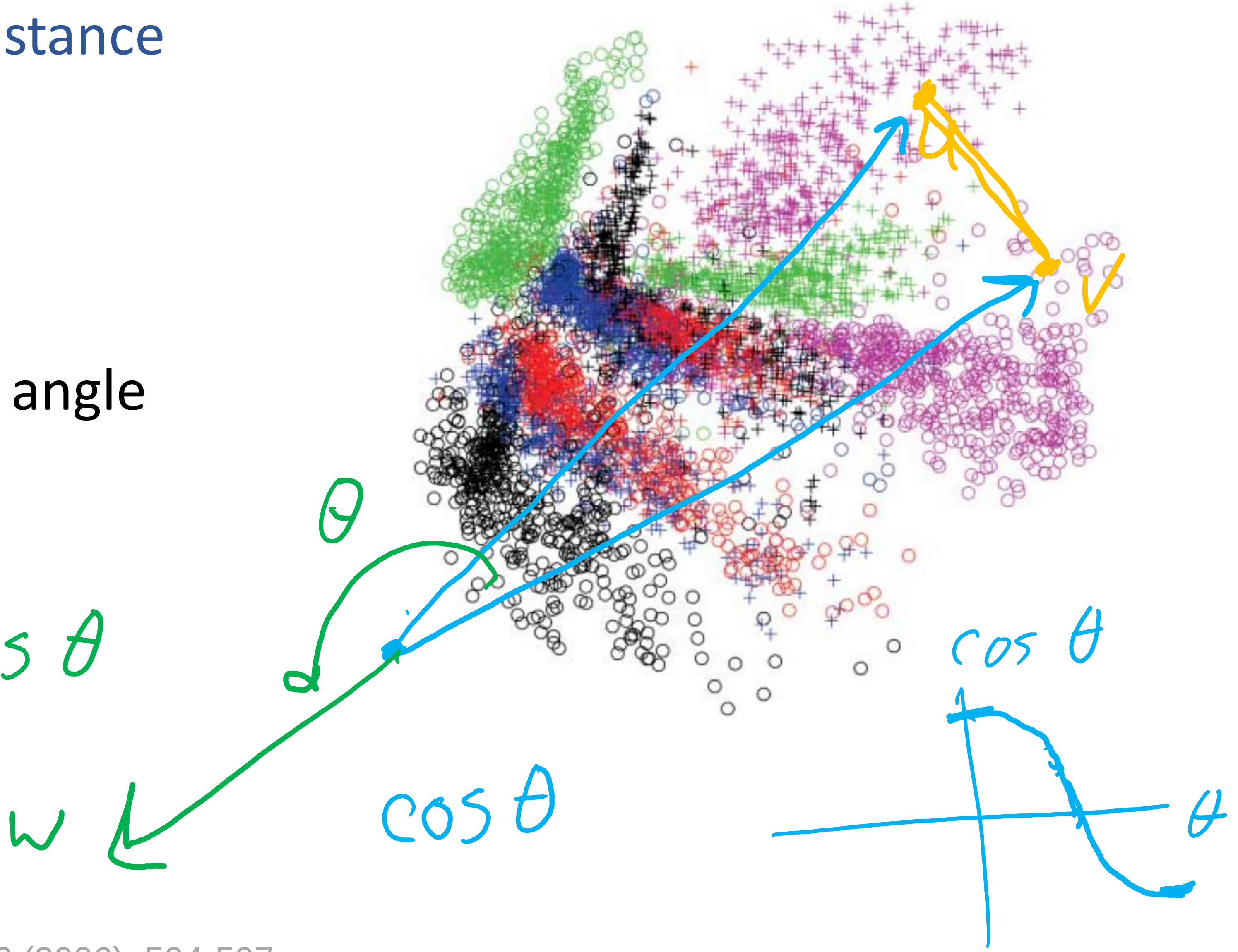
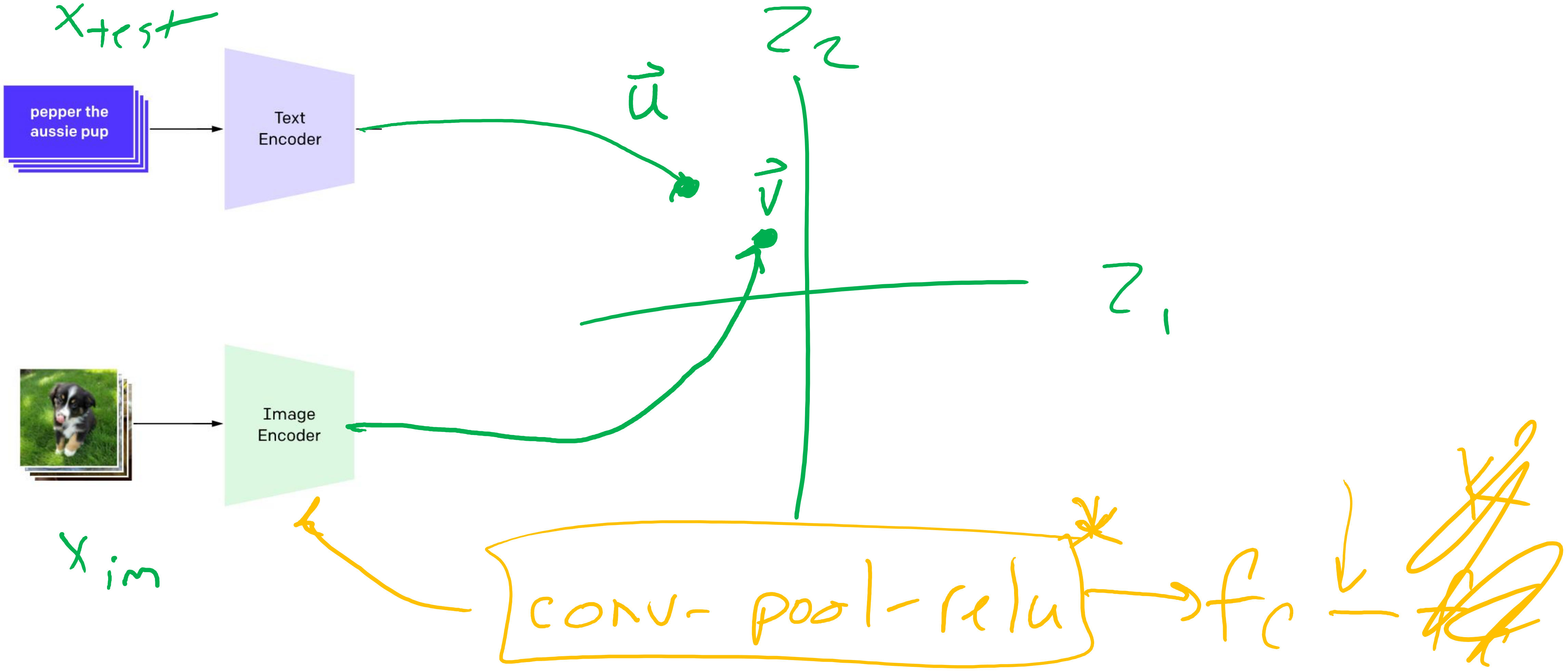


Image: Hinton & Salakhutdinov. *Science* 313.5786 (2006): 504-507.

Background: Learn a Feature Space

CLIP: Common feature space for text and images



pepper the aussie pup

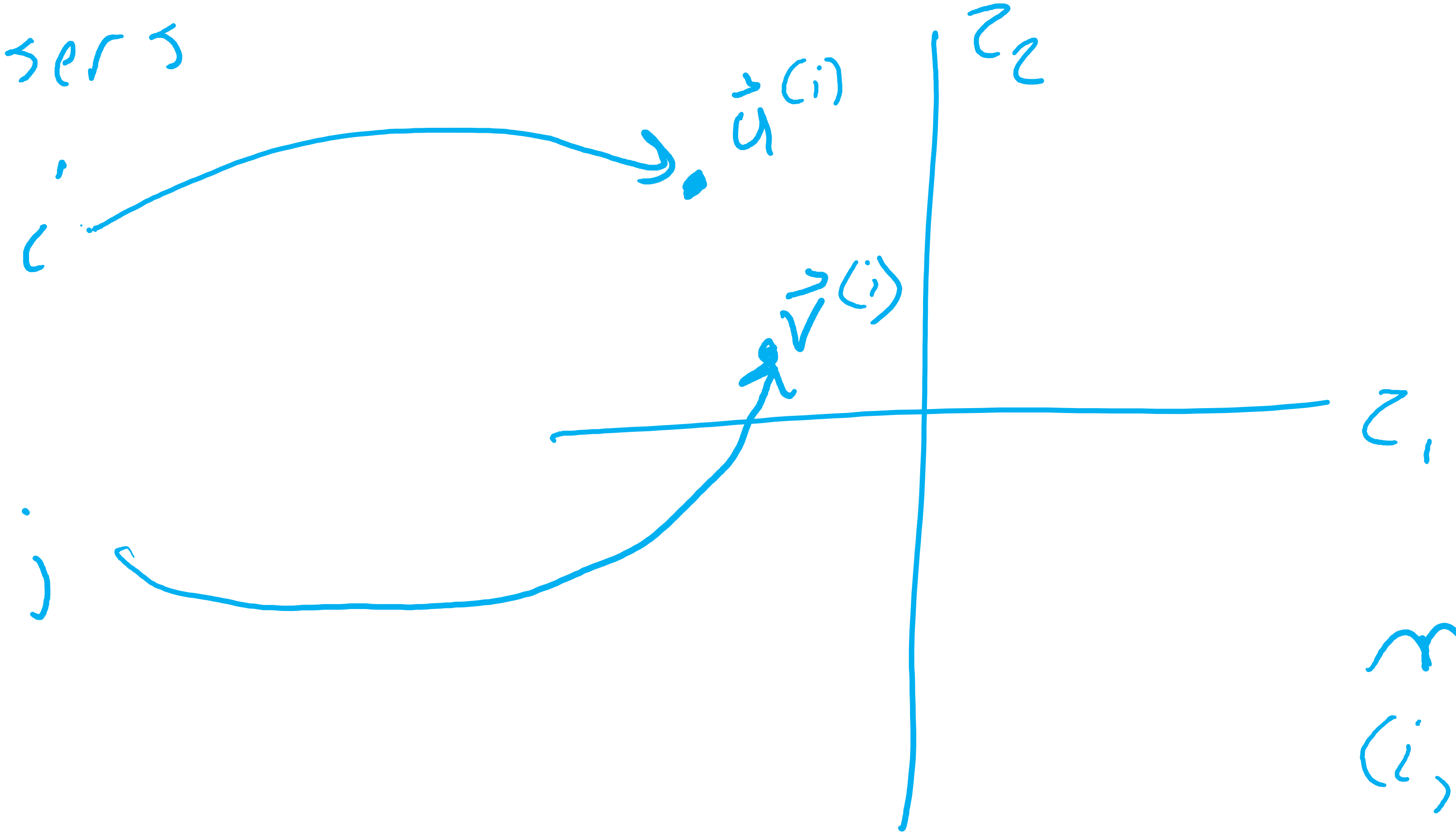
Text Encoder

Image Encoder

Recommender System: Matrix Factorization

Learning to map items and users to the same feature space

N users



$$\vec{u}^{(i)T}\vec{v}^{(j)} = \hat{r}_{ij}$$

$$\min_{(i,j,r)} \sum \left(r_{ij} - \hat{r}_{ij}\right)^2$$

M items

Recommender System: Matrix Factorization

Learning to map items and users to the same feature space

Recommender System: Matrix Factorization

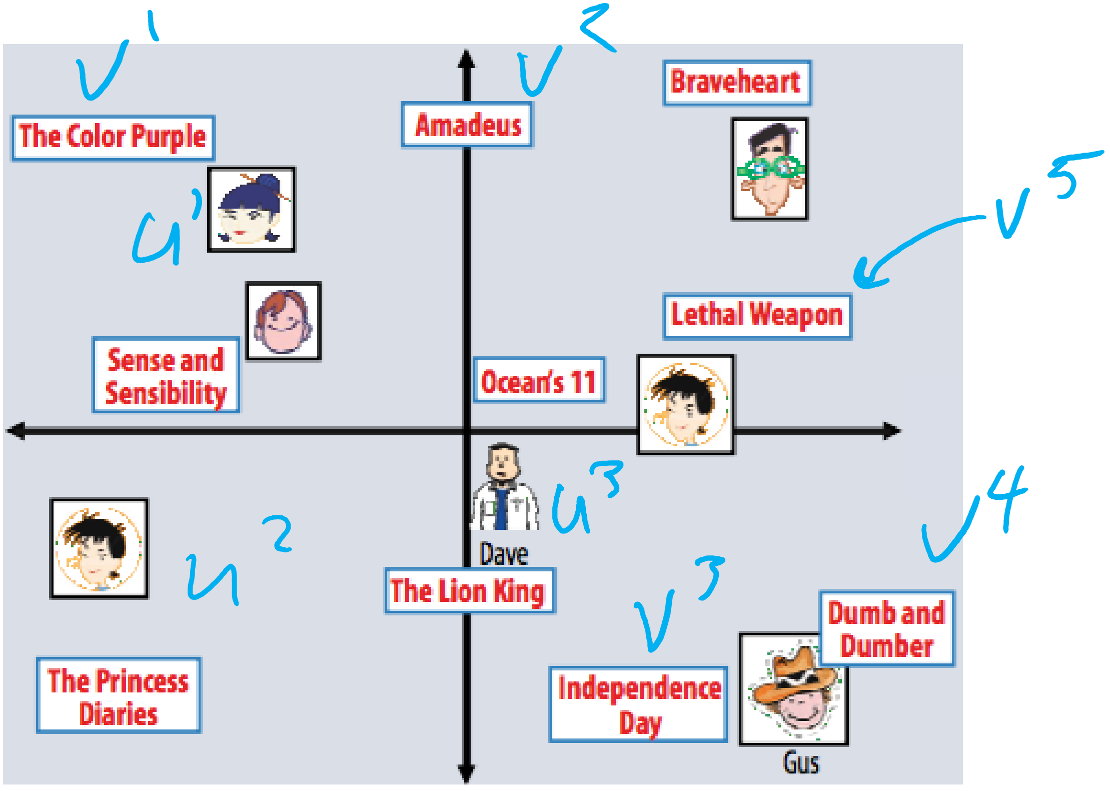
Learning to map items and users to the same lower dimensional space

N user

M item

$$U \in \mathbb{R}^{\underline{N} \times K}$$

$$V \in \mathbb{R}^{M \times K}$$



$K=2$

Figures from Koren et al. (2009)

Recommender System: Matrix Factorization

Optimization: Objective function using only the labels we have

$$\underset{U,V}{\operatorname{argmin}} \sum_{(i,j,r_{ij})} \left(r_{ij} - \underline{u^{(i)T} v^{(j)}}\right)^2$$

Poll 1

Is the following optimization a quadratic optimization?

$$\min_{\boldsymbol{U}\cancel{,\boldsymbol{V}}} \sum_{i,j \in \mathcal{S}} \left(r_{ij} - \mathbf{u}^{(i)^T} \underline{\mathbf{v}^{(j)}} \right)^2$$

A. ✗ A|V

B. Yes

C. No

D. ✗ A|V

Matrix Factorization

Method of *alternating minimization*

$$\min_{U,V} J(U,V) \qquad J(U,V) = \sum_{i,j\in S}\left(r_{ij} - {\mathbf{u}^{(i)}}^T \mathbf{v}^{(j)}\right)^2$$

do

① $\min_{U} J(U,V)$

② $\min_{V} J(U,V)$

$$U^{(t+1)} \leftarrow U^{(t)} - \alpha \nabla_U J(U^t, V^t)$$

$$V^{(t+1)} \leftarrow V^{(t)} - \alpha \nabla_V J(U^{t+1}, V^t)$$

Matrix Factorization

Method of *alternating minimization*

$$\min_{U,V} J(U,V) \qquad J(U,V) = \sum_{i,j\in\mathcal{S}} \left(r_{ij} - \mathbf{u}_i^T \mathbf{v}_j\right)^2$$

1) $\underset{U}{\operatorname{argmin}}\, J(U,V)$
2) $\underset{V}{\operatorname{argmin}}\, J(U,V)$

Matrix Factorization

K=2

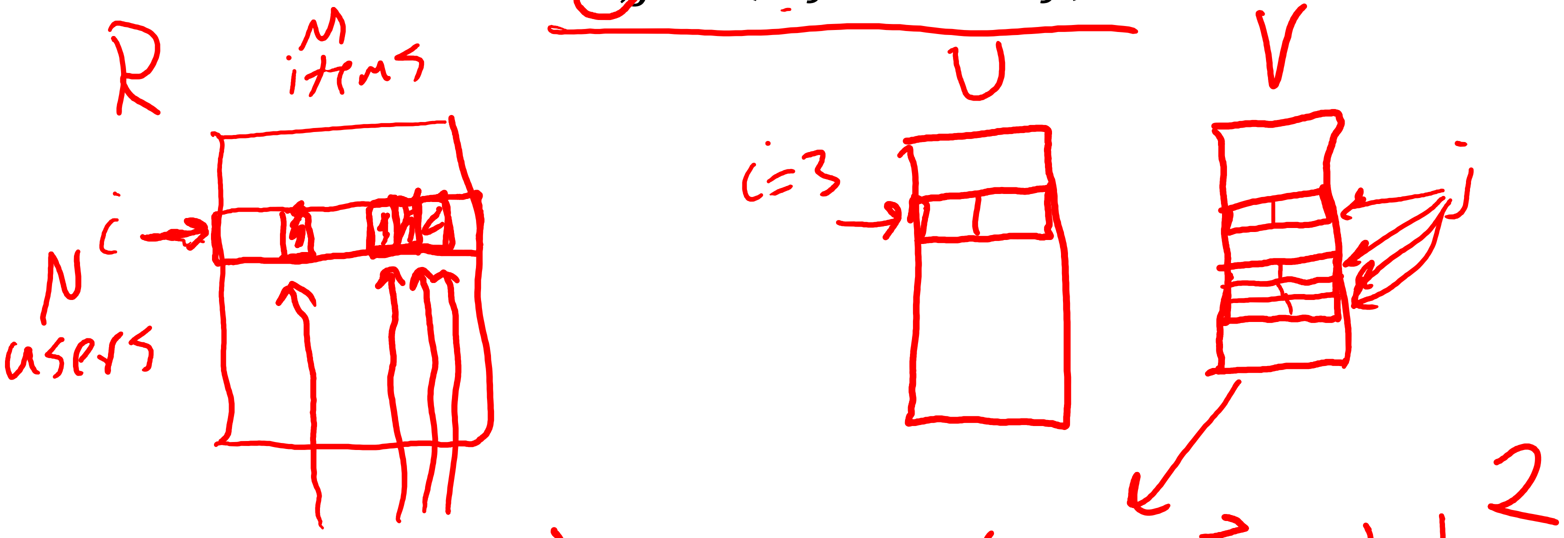
$i=3$ update $\vec{u}_i$

Method of *alternating minimization*

$$\min_{U,V} J(U,V) \qquad J(U,V) = \underline{\sum_{i,j\in\mathcal{S}} \left(r_{ij} - \mathbf{u}_i^T\mathbf{v}_j\right)^2}$$

1) $\underset{U}{\operatorname{argmin}}\, J(U,V)$

2) $\underset{V}{\operatorname{argmin}}\, J(U,V)$



1) For all i, $\underset{\mathbf{u}_i}{\operatorname{argmin}}\, J(U,V)$

2) For all j, $\underset{\mathbf{v}_j}{\operatorname{argmin}}\, J(U,V)$

$$\left\| \underset{y}{\vec{r}_{sub}} - \underset{X}{V_{sub}} \;@\; \underset{\theta}{\vec{u}_i} \right\|_2^2$$

Matrix Factorization

Method of *alternating minimization*

$$\min_{U,V} J(U,V) \qquad J(U,V) = \sum_{i,j\in\mathcal{S}} \left(r_{ij} - \mathbf{u}_i^T \mathbf{v}_j\right)^2$$

$$\left\| \underset{b}{\vec{r}_{sub}} - \underset{A}{V_{sub}}\, \vec{u}_i \right\|_2^2$$

$$(X^T X)^{-1} X^T y \rightarrow \hat{\theta}$$

1) For all i, $\operatorname*{argmin}_{\mathbf{u}_i} J(U,V)$ $\qquad \mathbf{u}_i^{(t+1)} = (A^T A)^{-1} A^T \boldsymbol{b}$

2) For all j, $\operatorname*{argmin}_{\mathbf{v}_j} J(U,V)$ $\qquad \mathbf{v}_j^{(t+1)} = (C^T C)^{-1} C^T \mathbf{d}$

Matrix Factorization

Method of *alternating minimization*

$$\min_{U,V} J(U,V) \qquad\qquad J(U,V) = \sum_{i,j\in\mathcal{S}} \left(r_{ij} - \mathbf{u}_i^T \mathbf{v}_j\right)^2$$

1) For all i, $\underset{\mathbf{u}_i}{\operatorname{argmin}} J(U,V)$ $\qquad \mathbf{u}_i^{(t+1)} = (A^T A)^{-1} A^T \boldsymbol{b}$

2) For all j, $\underset{\mathbf{v}_j}{\operatorname{argmin}} J(U,V)$ $\qquad \mathbf{v}_j^{(t+1)} = (C^T C)^{-1} C^T \mathbf{d}$

Matrix Factorization

First solve:

$$\min_{U,V} J(U,V) \qquad J(U,V) = \sum_{i,j\in\mathcal{S}} \left(r_{ij} - \mathbf{u}^{(i)^T}\mathbf{v}^{(j)}\right)^2$$

$U \quad V$

What then?

predict

$$h(i,j) \longrightarrow \hat{r}_{ij}$$

$$u^{(i)T} v^{(j)} \longrightarrow \hat{r}_{ij}$$

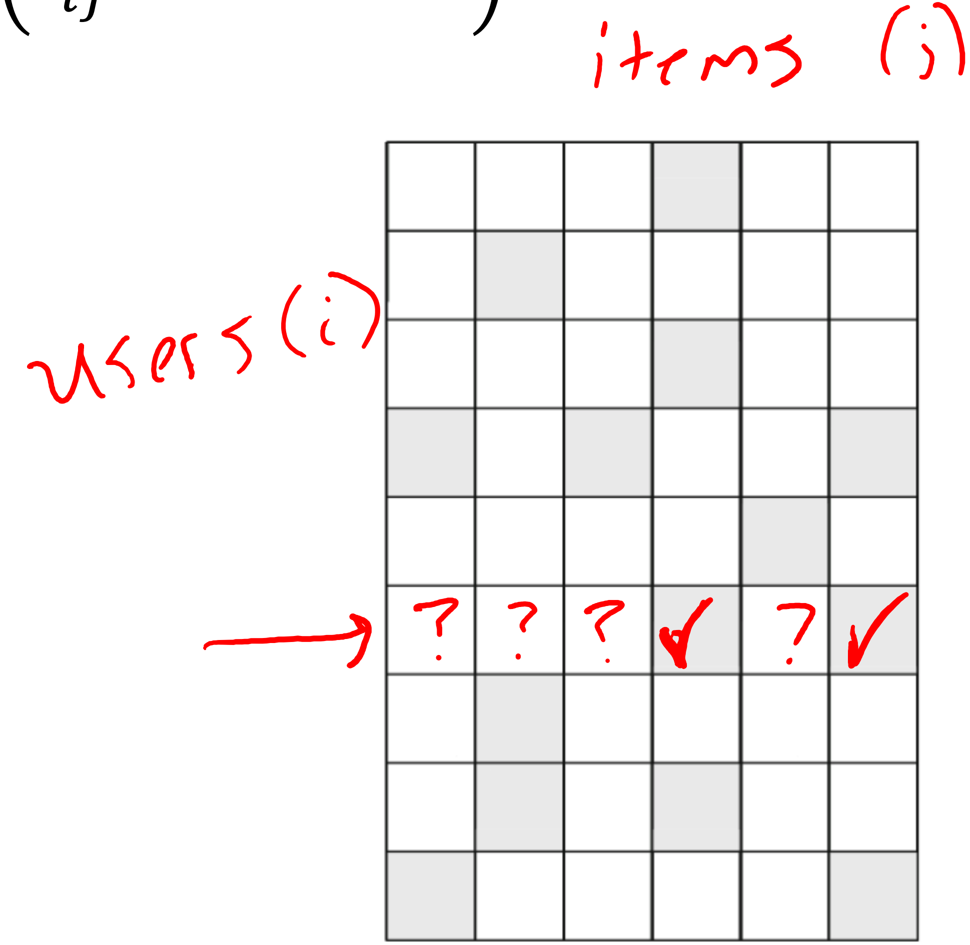
	Doctor Strange	Star Trek: Beyond	Zootopia
Alita	1	6,4	5
BB-8	3	4	2,5
C-3Po	3	5	2

Matrix Factorization

First solve:

$$\min_{U,V} J(U,V) \qquad J(\underline{U,V}) = \sum_{i,j\in\mathcal{S}}\left(r_{ij} - {\mathbf{u}^{(i)}}^T\mathbf{v}^{(j)}\right)^2$$

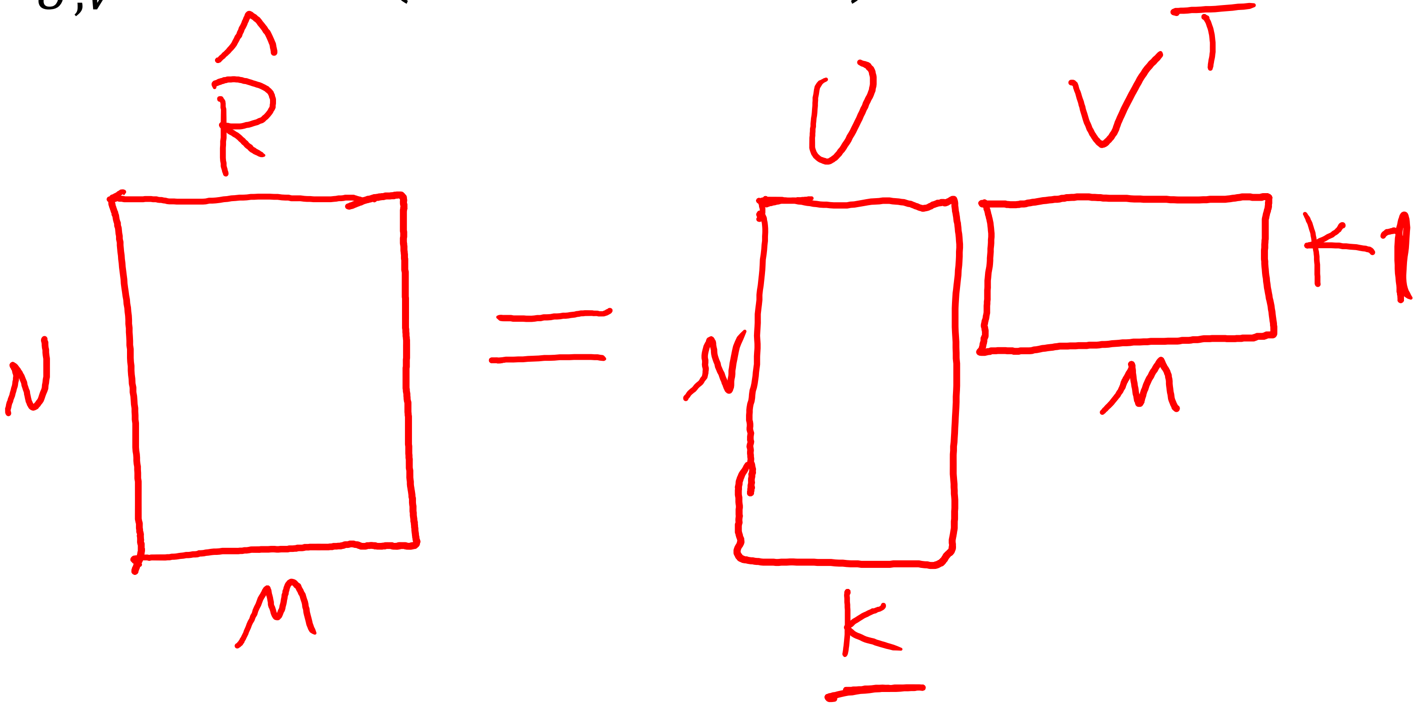
What then?



Recommender System: Matrix Factorization

Why is it called matrix factorization?

$$\min_{U,V} \sum_{i,j\in\mathcal{S}} \left(r_{ij} - \mathbf{u}^{(i)^T}\mathbf{v}^{(j)}\right)^2$$



SVD

$X = USV^T$

$R = U \underbrace{V^T}$

matrix factorization

Recommender System: Matrix Factorization

Why is it called matrix factorization?

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{R} - \boldsymbol{U}\boldsymbol{V}^T\|_F^2$$

Sparse labels ☹

	Doctor Strange	Star Trek: Beyond	Zootopia
Alita	1		5
BB-8	3	4	
C-3Po	3	5	2

Matrix Factorization

Add regularization to avoid overfitting

$$\min_{U,V} J(U,V) + \lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_F^2$$

$$J(U,V) = \sum_{i,j \in \mathcal{S}} \left(r_{ij} - \mathbf{u}^{(i)^T} \mathbf{v}^{(j)} \right)^2$$

Bias in Recommender Systems

What high-level problems can occur from recommender systems?

Summary

Recommender systems solve many **real-world** (*large-scale) **problems**

Collaborative filtering by matrix factorization (MF) is an **efficient** and **effective** approach

(Sparse matrix makes MF more challenging)

MF is just another example of a **common recipe**:

1. define a model
2. define an objective function
3. optimize

Optimization

- Use alternating minimization
- Add regularization to avoid overfitting