Recommender Systems: Latent Factor Models

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

The Netflix Prize

- Movie recommender system
- Training data:
  - 100 million ratings, 480,000 users, 17,770 movies
  - 6 years of data: 2000-2005
- Test data
  - Last few ratings of each user (2.8 million)

The Netflix Utility Matrix *R*

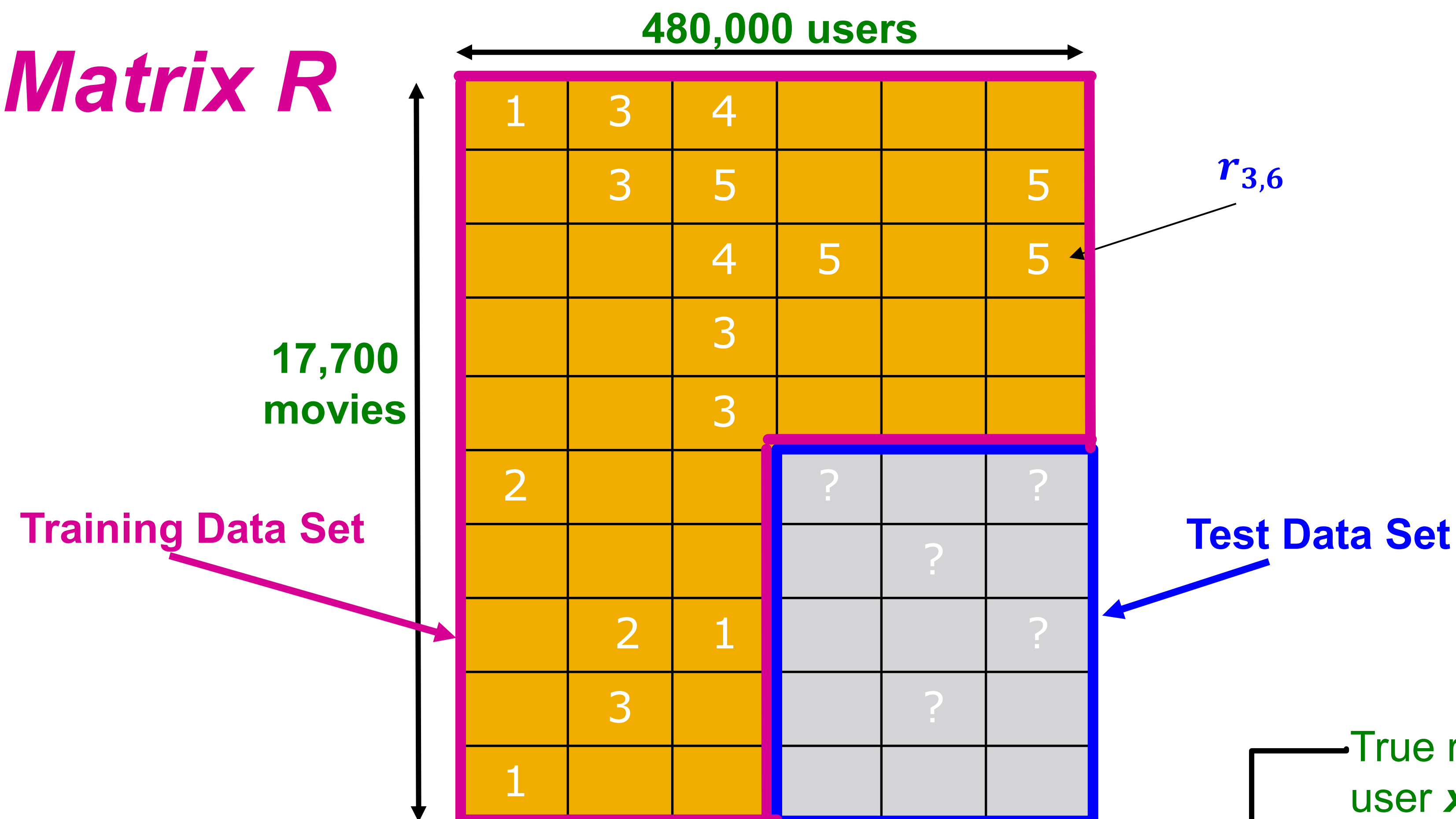
Matrix R

480,000 users

17,700 movies

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

Utility Matrix *R*: Evaluation



$$\textbf{RMSE} = \frac{1}{|R|}\sqrt{\sum_{(i,x)\in R}(\hat{r}_{xi} - r_{xi})^2}$$

True rating of user ***x*** on item ***i***

Predicted rating

The Netflix Prize

- **Given the training data**
  - 100 million ratings
- **Predict last few ratings of each user**
  - **Evaluation criterion:** Root Mean Square Error (**RMSE**)

$$= \frac{1}{|R|}\sqrt{\sum_{(i,x)\in R} (\hat{r}_{xi} - r_{xi})^2}$$

  - **Netflix's system RMSE: 0.9514**
- **Competition**
  - 2,700+ teams
  - **$1 million** prize for 10% improvement on Netflix

A Modern Recommender System

- **Multi-scale modeling of the data:** Combine top level, “regional” modeling of the data, with a refined, local view:
  - **Global:**
    - Overall deviations of users/movies
  - **Factorization:**
    - Addressing “regional” effects
  - **Collaborative filtering:**
    - Extract local patterns

Modeling Local & Global Effects

- **Global:**
  - Mean movie rating: **3.7 stars**
  - *The Sixth Sense* is **0.5** stars above avg.
  - Joe rates **0.2** stars below avg.

    ⇒ **Baseline estimation:**

    ***Joe* will rate *The Sixth Sense* 4 stars**

- **Local neighborhood (CF/NN):**
  - *Joe* didn't like related movie *Signs*
  - ⇒ **Final estimate:**

    ***Joe* will rate *The Sixth Sense* 3.8 stars**

Recap: Collaborative Filtering (CF)

- **Earliest and most popular collaborative filtering method**
- Derive unknown ratings from those of "**similar**" movies (item-item variant)
- Define similarity measure s_{ij} of items i and j
- Select k-nearest neighbors, compute the rating
  - $N(i; x)$: items most similar to i that were rated by x

$$\hat{r}_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij} … similarity of items i and j
r_{uj} … rating of user x on item j
$N(i;x)$ … set of items similar to item i that were rated by x

Modeling Local & Global Effects

- **In practice we get better estimates if we model deviations:**

$$\hat{r}_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij}}$$

baseline estimate for r_{xi}

$$\boldsymbol{b_{xi} = \mu + b_x + b_i}$$

μ = overall mean rating
b_x = rating deviation of user ***x***
= (*avg. rating of* user ***x***) – ***μ***
b_i = (*avg. rating of* movie ***i***) – ***μ***

Problems/Issues:

1) Similarity measures are "arbitrary"
2) Pairwise similarities neglect interdependencies among users
3) Taking a weighted average can be restricting

RMSE of Various Methods