# Unsupervised Learning- Matrix Completion

STAT 542 Final Project
Group 11

# Outline

- Dataset Introduction
- Models
- Results Comparison (RMSE)
- Difficulties

# Dataset Introduction

# Dataset Introduction

## Motivation

- Most of the ratings are 3, and could be due to that the students have not tried the restaurants
- The dataset was collected from students, and we want to obtain a similar dataset from students to work on

## Description

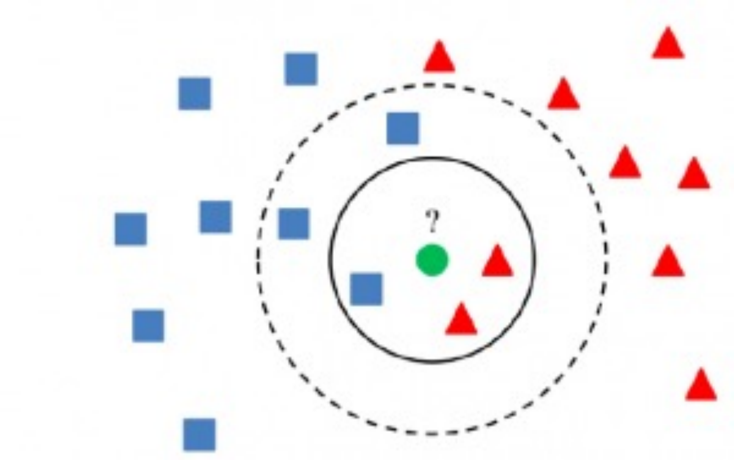
- Created an identical survey with the same 15 restaurants and the same rating scale (1 to 5)
- Collected from 50 students of UIUC
- $(n, p) = (50, 15)$
- Training set: 50% sparsity

# K Nearest Neighbor

# KNN

## Methodology

- Compute similarity of the data points
- Define *k* nearest neighbor



# Similarity Measures

- Pearson correlation:

$$m_{Pearson}(X,Y) = \frac{1}{n-1}\sum_{l=1}^{n}\left(\frac{x_l-\bar{x}}{s_x}\right)\left(\frac{y_l-\bar{y}}{s_y}\right) \text{ , Similarity } s = \frac{m+1}{2}$$

- Cosine Similarity:

$$m_{cosine}(X,Y) = \frac{X \cdot Y}{\|X\|\|Y\|} \text{ , Similarity } s = \frac{m+1}{2}$$

- Euclidean Distance:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \text{ , Similarity } s = \frac{1}{1+d}$$

# User-Based Collaborative Filtering Using KNN

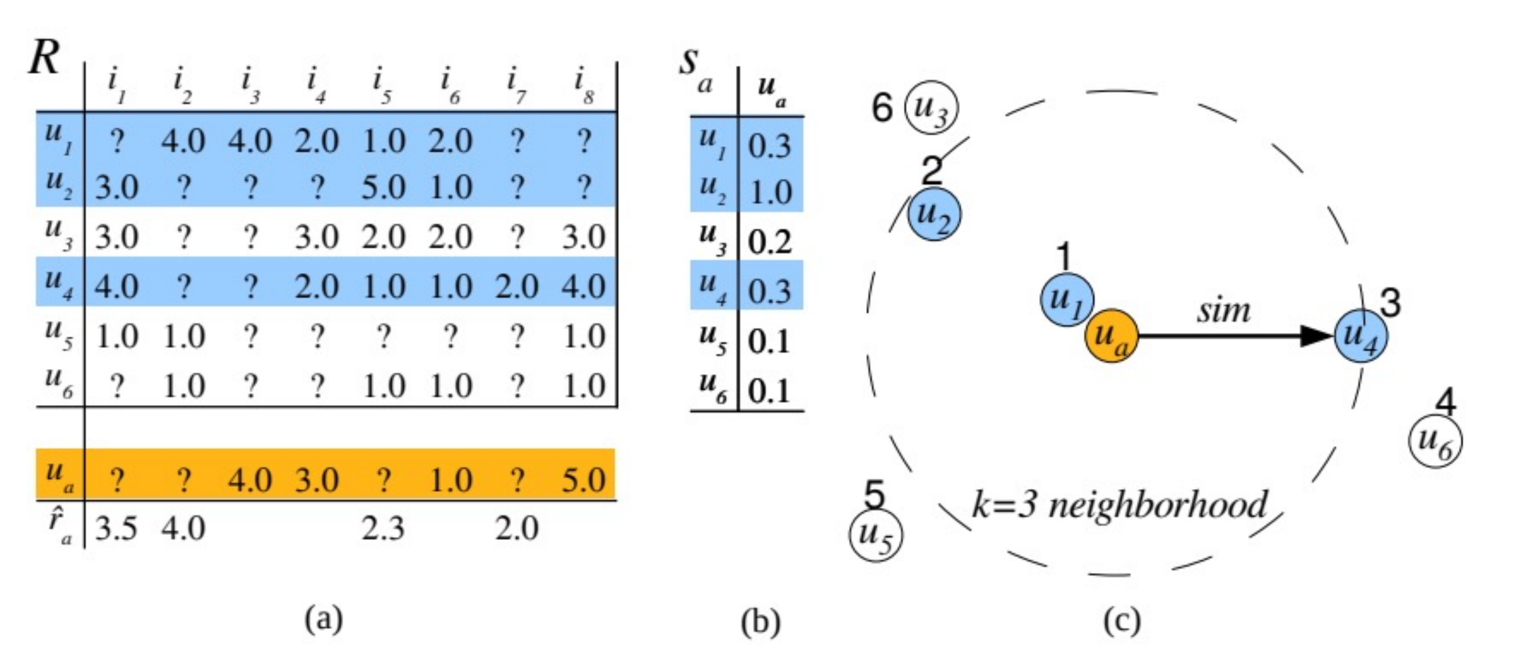
# User-Based Collaborative Filtering

## Methodology

- Find a neighborhood of similar users:
  - Missing ratings are skipped in the calculation.
  - Compute similarity.
  - Define number *k* of nearest neighbors (select highest similarity).
- Predict missing ratings by taking the average rating of users in the *k* nearest neighborhood.

# User-Based Collaborative Filtering

## Methodology



| $R$ | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ |
|---|---|---|---|---|---|---|---|---|
| $u_1$ | ? | 4.0 | 4.0 | 2.0 | 1.0 | 2.0 | ? | ? |
| $u_2$ | 3.0 | ? | ? | ? | 5.0 | 1.0 | ? | ? |
| $u_3$ | 3.0 | ? | ? | 3.0 | 2.0 | 2.0 | ? | 3.0 |
| $u_4$ | 4.0 | ? | ? | 2.0 | 1.0 | 1.0 | 2.0 | 4.0 |
| $u_5$ | 1.0 | 1.0 | ? | ? | ? | ? | ? | 1.0 |
| $u_6$ | ? | 1.0 | ? | ? | 1.0 | 1.0 | ? | 1.0 |
| $u_a$ | ? | ? | 4.0 | 3.0 | ? | 1.0 | ? | 5.0 |
| $\hat{r}_a$ | 3.5 | 4.0 | | | 2.3 | | 2.0 | |

(a)

| $S_a$ | $u_a$ |
|---|---|
| $u_1$ | 0.3 |
| $u_2$ | 1.0 |
| $u_3$ | 0.2 |
| $u_4$ | 0.3 |
| $u_5$ | 0.1 |
| $u_6$ | 0.1 |

(b)

(c)

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

# User-Based CF using KNN

Results

| Similarity measure | Optimal K | RMSE |
|---|---|---|
| Pearson | 11 | 0.6640 |
| Cosine | 26 | 0.6442 |
| Euclidean | 22 | 0.6415 |

# Item-Based Collaborative Filtering Using KNN

# Item-Based Collaborative Filtering

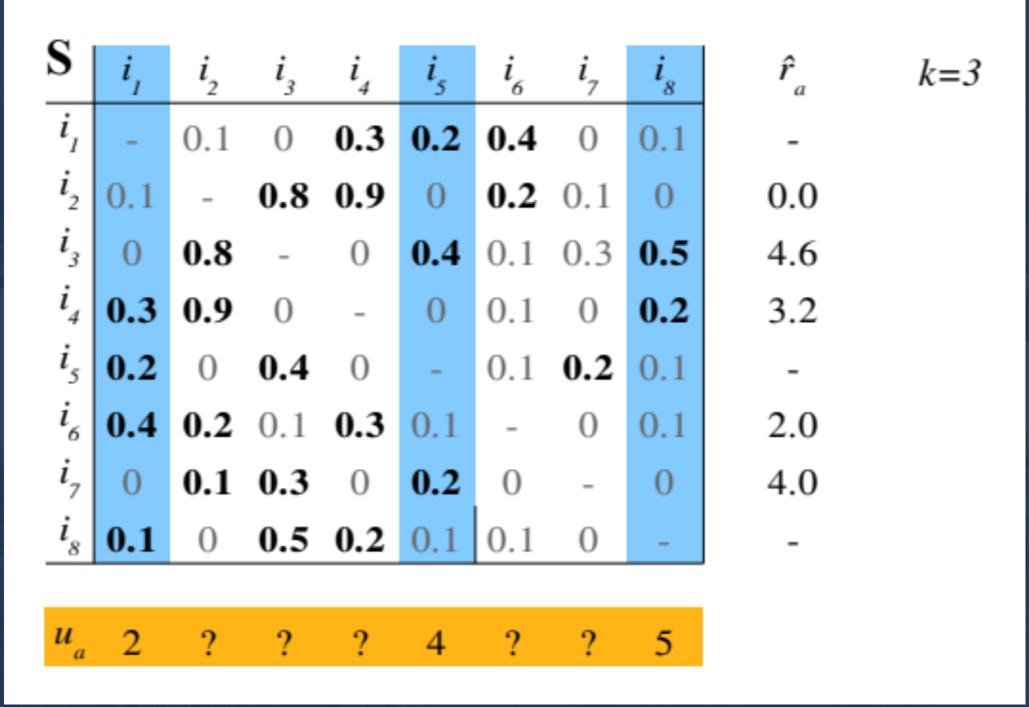
## Methodology

- Find a neighborhood of similar items:
  - Missing ratings are skipped.
  - Compute similarity.
  - Define number $k$ of nearest neighbors (select highest similarity).

- Predict missing ratings by taking the weighted-average rating of items in the $k$ nearest neighborhood.
  - Weight: similarity
  - Rating: user's rating matched similar items

# Item-Based Collaborative Filtering

## Methodology



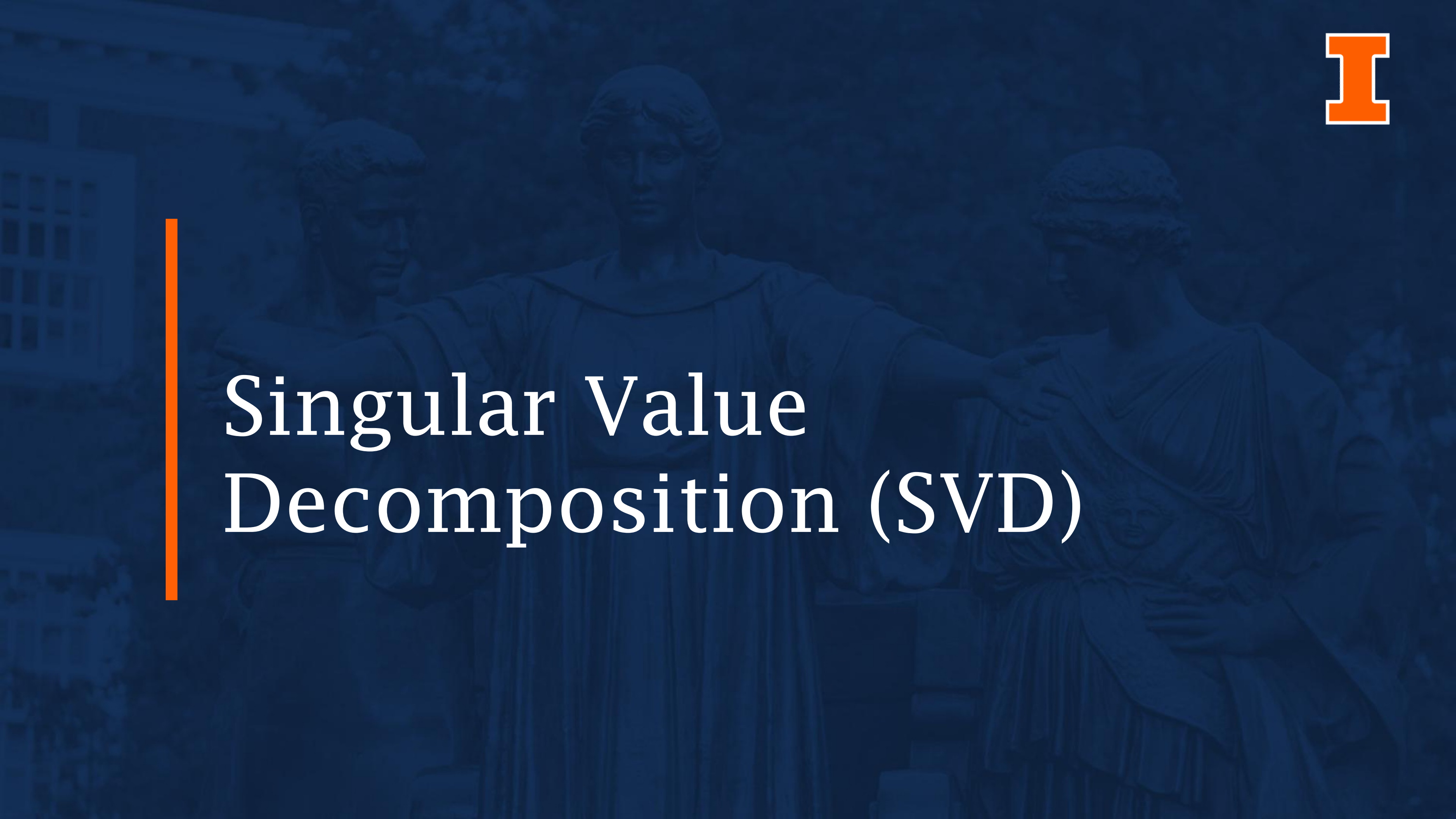
| S | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ | $i_7$ | $i_8$ | $\hat{r}_a$ | *k=3* |
|---|---|---|---|---|---|---|---|---|---|---|
| $i_1$ | - | 0.1 | 0 | **0.3** | **0.2** | **0.4** | 0 | 0.1 | - | |
| $i_2$ | 0.1 | - | **0.8** | **0.9** | 0 | **0.2** | 0.1 | 0 | 0.0 | |
| $i_3$ | 0 | **0.8** | - | 0 | **0.4** | 0.1 | 0.3 | **0.5** | 4.6 | |
| $i_4$ | **0.3** | **0.9** | 0 | - | 0 | 0.1 | 0 | **0.2** | 3.2 | |
| $i_5$ | **0.2** | 0 | **0.4** | 0 | - | 0.1 | **0.2** | 0.1 | - | |
| $i_6$ | **0.4** | **0.2** | 0.1 | **0.3** | 0.1 | - | 0 | 0.1 | 2.0 | |
| $i_7$ | 0 | **0.1** | **0.3** | 0 | **0.2** | 0 | - | 0 | 4.0 | |
| $i_8$ | **0.1** | 0 | **0.5** | **0.2** | 0.1 | 0.1 | 0 | - | - | |
| $u_a$ | 2 | ? | ? | ? | 4 | ? | ? | 5 | | |

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

# Item-Based CF using KNN

Results

| Similarity measure | Optimal K | RMSE |
|---|---|---|
| Pearson | 7 | 0.6829 |
| Cosine | 8 | 0.6338 |
| Euclidean | 11 | 0.6337 |



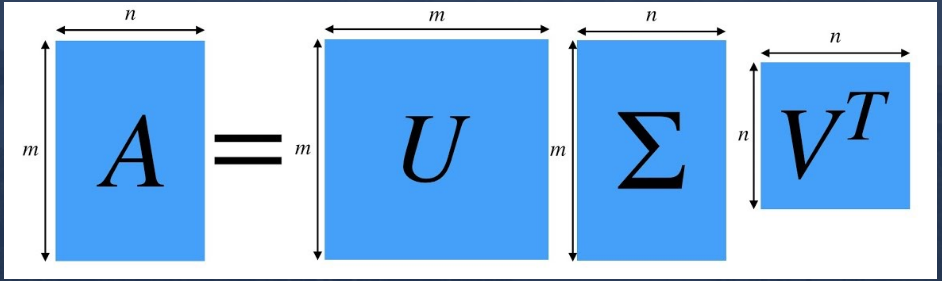
# Singular Value Decomposition (SVD)

# SVD

## Methodology

- Singular Value Decomposition (SVD) of a matrix $A$ is a factorization into three matrices $U$, $\Sigma$, and $V$, with $U$ and $V$ being orthogonal matrices and $\Sigma$ being a diagonal matrix with singular value entries.

$$A = U\Sigma V^T$$



# Iterative SVD

Steps:

1. Initial guess for NaN values in the matrix A
2. Apply SVD to A
3. Apply Low-Rank Matrix Approximation
4. Replace the known value in A to get matrix A’
5. Repeat the process until the difference between A and A’ is less than a pre-determined threshold

*Note that the results highly depend on the initial matrix.

# SVD

Results

| How NaN values were initialized | RMSE |
|---|---|
| 3 | 0.7236 |
| Row mean | 0.6706 |
| Column mean | 0.6417 |

What if we used other initial matrices to implement SVD?

# KNN+SVD

Steps:

1. Use the results from KNN as matrix A
2. Apply SVD to A
3. Apply Low-Rank Matrix Approximation
4. Replace the known value in A to get matrix A’
5. Repeat the process until the difference between A and A’ is less than a pre-determined threshold

# KNN+SVD

Results

| KNN | RMSE |
|---|---|
| User-Based | 0.6516 |
| Item Based | 0.6412 |

# Results Comparison

# Results Comparison

|  | IBCF Using KNN | UBCF Using KNN | SVD | KNN+SVD |
|---|---|---|---|---|
| RMSE | 0.6337 | 0.6415 | 0.6417 | 0.6412 |

# Difficulties

# Difficulties

## Problem 1- Dataset

- We observed that most of the values in the provided dataset contains 3, making google reviews or other ratings online unreliable

## Solution:

- We decided to collect our own dataset, in ways that maximizes its similarity with Feedback.csv

# Difficulties

Problem 2- User-Based and Item-Based CF

- User rating bias: some users tend to use higher ratings while some tend to use lower ratings

Solution:

- Center the rows of user-item rating by doing normalization

$$h(r_{jl}) = r_{jl} - \bar{r_j}$$

# Thank you