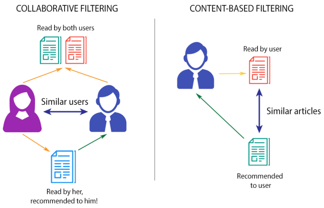
Algorithms and Data Structures

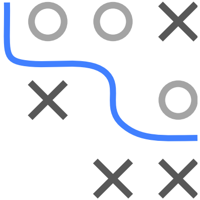
Matrix Approximation
Recommender Systems Application using SVD



Learning goals

- Recommender systems application

APPLICATION: RECOMMENDER SYSTEMS (1)



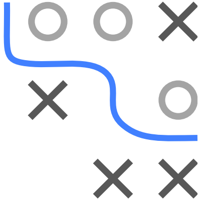
Initial situation:

- *m* users (e.g. Netflix users)
- *n* items (e.g. Movies)
- **X** User-Item Matrix: x_{ij} rating of user *i* for item *j*

Example: Suppose there are 4 movies and 6 users in our database.

	Die Hard	Top Gun	Titanic	Notting Hill
User 1	5	NA	3	NA
User 2	5	4	3	3
User 3	2	NA	5	NA
User 4	5	5	3	1
User 5	1	2	5	5
User 6	1	2	4	5

APPLICATION: RECOMMENDER SYSTEMS (1) / 2



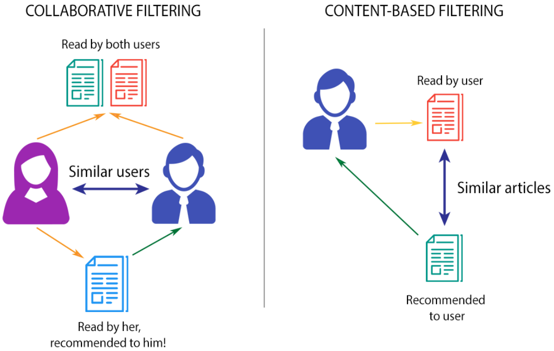
Of all available items only a few are evaluated by one user (e.g. Netflix, Amazon), thus the user-item matrix is **sparse** in many applications.

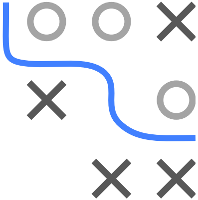
The target is to make a **prediction** for these missing values, which quantifies how high the interest of a user in the respective item is.

Then we recommend the items that users have not yet rated, but are likely to find interesting.

APPLICATION: RECOMMENDER SYSTEMS (1) / 3

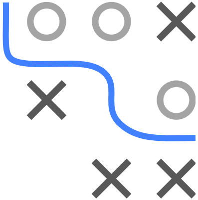
Basically one distinguishes between two approaches:



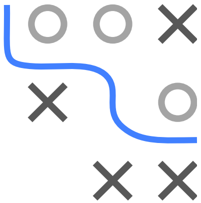


APPLICATION: RECOMMENDER SYSTEMS (1) / 4

- **Collaborative Filtering**: Identify "similar" users based on their behavior and recommend items in which similar users are most interested (e.g. by using singular value decomposition).
- **Content-based**: Identify - using a similarity measure - "similar" items and recommend items that are similar to the items that the user has rated high in the past.



APPLICATION: RECOMMENDER SYSTEMS (1) / 5



A collaborative filtering approach results from the singular value decomposition.

Procedure:

1. Fill up the data matrix **X** by imputation, e.g.
   - By item average rating, i.e. the column mean value
   - By user average rating, i.e. the row mean value
   - By overall average rating
2. Choice of rank k: Calculate singular values and select k so that $\sigma_k \gg \sigma_{k+1}$. Larger k yields a better approximation, smaller k a less complex model.
3. Calculate singular value decomposition of rank k and from it the matrices **W** and **H**.
4. Calculate **WH** and recommend to each user the movies with the best estimated rating from the ones he has not seen yet

APPLICATION: RECOMMENDER SYSTEMS (1) / 6

Back to the example:

```
X
```

```
##          Die Hard   Top Gun   Titanic   Notting Hill
## User 1       5         NA         3           NA
## User 2       5         4          3           3
## User 3       2         NA         5           NA
## User 4       5         5          3           1
## User 5       1         2          5           5
## User 6       1         2          4           5
```

1. We replace missing values with the mean value of each row:

```
X = ifelse(is.na(X), rowMeans(X, na.rm = TRUE), unlist(X))
```

APPLICATION: RECOMMENDER SYSTEMS (1) / 7

2. Choice of k:

```
svd(X)$d
## [1] 17.24 6.13 2.14 0.39
```

   We choose $k = 2$.

3. Calculate the matrices **W** and **H** using a singular value decomposition:

```
res = svd(X, nu = 2, nv = 2)
Uk = res$u
Vk = res$v
Sigmak = diag(res$d[1:2])
W = Uk %*% sqrt(Sigmak)
H = sqrt(Sigmak) %*% t(Vk)
```

APPLICATION: RECOMMENDER SYSTEMS (1) / 8

4. Calculate the prediction $\hat{\mathbf{X}} = \mathbf{WH}$

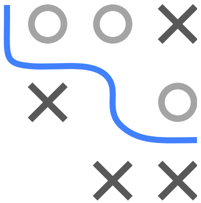
```
Xhat = W %*% H
```

	Die Hard	Top Gun	Titanic	Notting Hill
User 1	4.82	3.69	2.37	3.41
User 2	5.03	3.96	2.91	3.07
User 3	2.24	2.70	3.64	5.44
User 4	5.34	4.37	3.65	3.96
User 5	2.87	2.90	3.85	4.52
User 6	1.09	1.85	4.05	5.00

Table: User Ratings for Movies

Since user 1 is similar to user 2 and user 4 due to their past ratings, we would recommend "Top Gun". However, for user 3 we would recommend "Notting Hill", since this user is more similar to user 5 and user 6 and they rated the movie particularly well.

APPLICATION: RECOMMENDER SYSTEMS (1) / 9



Disadvantages of solution by singular value decomposition:

Often the resulting matrices **W** and **H** are not really interpretable because they contain negative values.

If the values are naturally non-negative, such as

- Pixel intensities
- Counts
- User scores / ratings
- ...

one often wants to find a non-negative matrix factorization to increase interpretability, i.e. $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$ $^{(*)}$.

$^{(*)}$ $\geq$ is to be understood component-wise