Algorithms and Data Structures

Matrix Approximation
Non-Negative Matrix Factorization &
Recommender Systems Application

Learning goals

- Non-negative matrix factorization
- Recommender systems application

NON-NEGATIVE MATRIX FACTORIZATION

This leads to a constrained optimization problem

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{m\times k},\mathbf{H}\in\mathbb{R}^{k\times n}} \quad & \|\mathbf{X}-\mathbf{WH}\|^2, \\ \text{with} \quad & \mathbf{W}\geq 0, \mathbf{H}\geq 0 \end{aligned}$$

The following problems must be addressed

1. NMF is NP-hard
2. NMF is ill-posed
3. Choice of rank k

NON-NEGATIVE MATRIX FACTORIZATION / 2

1. NMF is NP-hard
The problem is only convex in either **W** or **H**, but not in both simultaneously. Probably there is no efficient, exact solution for NMF. There are efficient heuristics such as **multiplicative update rules**, but convergence to a global optimum cannot be guaranteed.

Algorithm Multiplicative Update Rules

Initialize $\mathbf{W}, \mathbf{H} \geq \mathbf{0}$
repeat
$\quad h_{ij} \leftarrow h_{ij} \frac{(\mathbf{W}^\mathbf{T}\mathbf{X})_{ij}}{(\mathbf{W}^\mathbf{T}\mathbf{W}\mathbf{H})_{ij}}$
$\quad w_{ij} \leftarrow w_{ij} \frac{(\mathbf{X}\mathbf{H}^\mathbf{T})_{ij}}{(\mathbf{W}\mathbf{H}\mathbf{H}^\mathbf{T})_{ij}}$
until Stop criterion fulfilled

NON-NEGATIVE MATRIX FACTORIZATION / 3

❷ NMF is ill-posed

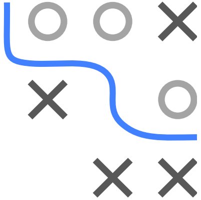
The problem can usually not be solved uniquely.

$$
\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}
= \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0.5 \end{pmatrix}
$$

$$
= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}
$$

Different factorizations mean different interpretations. Therefore in practice a regularization term is often added to the target function.

NON-NEGATIVE MATRIX FACTORIZATION / 4



❸ Choice of rank k

In contrast to singular value decomposition, it is much more difficult to determine the rank k in advance.

Possibilities:

- k is fixed in advance (based on prior knowledge / intuition) or results for different k are compared
- k is automatically estimated during NMF (not discussed here)

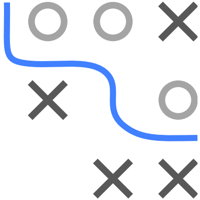
APPLICATION: RECOMMENDER SYSTEMS (2)

Back to our previous **example**:

	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

Non-negative matrix factorization offers an alternative to singular value decomposition. The advantage of a NMF solution is the increased interpretability of the matrices **W** and **H**.

APPLICATION: RECOMMENDER SYSTEMS (2) / 2



1. We replace missing values with the row mean value:
2. Choice of k:

   We suspect that our movie database contains movies from two different categories and set $k = 2$.

3. Non-negative matrix factorization:

```
set.seed(1)
res = nmf(X, rank = 2)
W = res@fit@W
H = res@fit@H
```

APPLICATION: RECOMMENDER SYSTEMS (2) / 3

```
W
##           Action  Romance
## User 1      3.86    1.861
## User 2      4.06    1.360
## User 3      1.83    2.978
## User 4      5.14    0.098
## User 5      0.38    3.918
## User 6      0.44    3.540
```

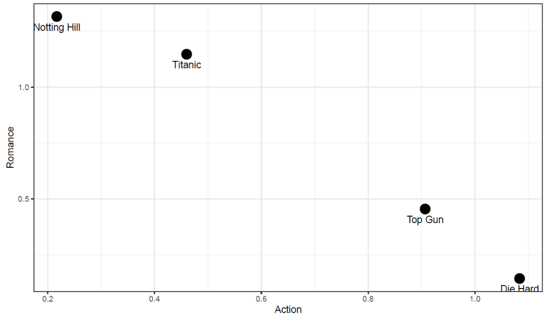
```
H
##          Die Hard  Top Gun  Titanic  Notting Hill
## Action       1.08     0.91     0.46          0.22
## Romance      0.14     0.46     1.15          1.32
```

APPLICATION: RECOMMENDER SYSTEMS (2) / 4

The columns of the 6×2 matrix **W** could be interpreted as movie categories (here: "Action" and "Romance"). The figure shows which users prefer which categories.

APPLICATION: RECOMMENDER SYSTEMS (2) / 5

The entries of the 2×4 matrix **H** describe which movies are to be assigned to which category.



APPLICATION: RECOMMENDER SYSTEMS (2) / 6

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

```
W %*% H
```

```
##            Die Hard  Top Gun  Titanic  Notting Hill
## User 1       4.45     4.3      3.9         3.3
## User 2       4.60     4.3      3.4         2.7
## User 3       2.41     3.0      4.3         4.3
## User 4       5.58     4.7      2.5         1.2
## User 5       0.97     2.1      4.7         5.2
## User 6       0.98     2.0      4.3         4.8
```

Here we would also recommend "Top Gun" to user 1, an action movie. For user 3 we recommend "Notting Hill", because he tends to prefer romantic movies.

MORE MATERIAL ON RECOMMENDER SYSTEMS

More on Recommender Systems:

- Matrix Factorization Techniques for Recommender Systems
- Recommender Systems Comparison (including implementation in `R`)
- Movielens Dataset