# Atelier 1 - Systèmes de recommandation

Cet atelier couvre principalement la matière des filtres collaboratifs et comporte trois volets.

1. Nous avons vu au cours l'exemple de la factorisation SVD d'une matrice termes documents de Landauer et al. (1998). Refaites la même démonstration, mais cette fois utilisant la méthode ALS. Quels résultats obtenez-vous avec deux et trois dimensions latentes et comment ces résultats se comparent-ils avec SVD? (7.5 pts.)

## SVD

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('./data/landauer.csv')
U, s, V = np.linalg.svd(df, full_matrices=False)

k = 3

U = U[:, :k]
s = s[:k]
V = V[:k, :]

R_svd = pd.DataFrame((U * s) @ V, index=df.index, columns=df.columns)
```

## Alternate Least Squares (ALS)

```
import pandas as pd
import numpy as np

df = pd.read_csv('./data/landauer.csv')
n_terms, n_documents = df.shape



k = 3
np.random.seed(42)
rng = np.random.default_rng()
P_hat = rng.normal(size=(n_terms, k))
Q_hat = rng.normal(size=(k, n_documents))
R = df.values

while True:
    P_hati = P_hat
```

```
    Q_hati = Q_hat

    P_hat = R @ Q_hat.T @ np.linalg.inv(Q_hat @ Q_hat.T)
    Q_hat = np.linalg.inv(P_hat.T @ P_hat) @ P_hat.T @ R

    if np.allclose(P_hat, P_hati) and np.allclose(Q_hat, Q_hati):
        break

R_als = pd.DataFrame(P_hat @ Q_hat, index=df.index,
columns=df.columns)
```

## Comparaison des résultats

```
R_svd.corr().round(2)
```

| | c1 | c2 | c3 | c4 | c5 | m1 | m2 | m3 | m4 |
|---|---|---|---|---|---|---|---|---|---|
| c1 | 1.00 | 0.13 | 1.00 | 0.97 | -0.19 | -0.49 | -0.53 | -0.54 | -0.67 |
| c2 | 0.13 | 1.00 | 0.21 | -0.14 | 0.95 | -0.67 | -0.63 | -0.62 | -0.40 |
| c3 | 1.00 | 0.21 | 1.00 | 0.94 | -0.11 | -0.55 | -0.58 | -0.60 | -0.71 |
| c4 | 0.97 | -0.14 | 0.94 | 1.00 | -0.44 | -0.30 | -0.35 | -0.37 | -0.55 |
| c5 | -0.19 | 0.95 | -0.11 | -0.44 | 1.00 | -0.56 | -0.51 | -0.49 | -0.24 |
| m1 | -0.49 | -0.67 | -0.55 | -0.30 | -0.56 | 1.00 | 1.00 | 1.00 | 0.94 |
| m2 | -0.53 | -0.63 | -0.58 | -0.35 | -0.51 | 1.00 | 1.00 | 1.00 | 0.96 |
| m3 | -0.54 | -0.62 | -0.60 | -0.37 | -0.49 | 1.00 | 1.00 | 1.00 | 0.96 |
| m4 | -0.67 | -0.40 | -0.71 | -0.55 | -0.24 | 0.94 | 0.96 | 0.96 | 1.00 |

```
R_als.corr().round(2)
```

| | c1 | c2 | c3 | c4 | c5 | m1 | m2 | m3 | m4 |
|---|---|---|---|---|---|---|---|---|---|
| c1 | 1.00 | 0.13 | 1.00 | 0.97 | -0.19 | -0.49 | -0.53 | -0.54 | -0.67 |
| c2 | 0.13 | 1.00 | 0.21 | -0.14 | 0.95 | -0.67 | -0.63 | -0.62 | -0.40 |
| c3 | 1.00 | 0.21 | 1.00 | 0.94 | -0.11 | -0.55 | -0.58 | -0.60 | -0.71 |
| c4 | 0.97 | -0.14 | 0.94 | 1.00 | -0.44 | -0.30 | -0.35 | -0.37 | -0.55 |
| c5 | -0.19 | 0.95 | -0.11 | -0.44 | 1.00 | -0.56 | -0.51 | -0.49 | -0.24 |
| m1 | -0.49 | -0.67 | -0.55 | -0.30 | -0.56 | 1.00 | 1.00 | 1.00 | 0.94 |
| m2 | -0.53 | -0.63 | -0.58 | -0.35 | -0.51 | 1.00 | 1.00 | 1.00 | 0.96 |
| m3 | -0.54 | -0.62 | -0.60 | -0.37 | -0.49 | 1.00 | 1.00 | 1.00 | 0.96 |
| m4 | -0.67 | -0.40 | -0.71 | -0.55 | -0.24 | 0.94 | 0.96 | 0.96 | 1.00 |

```
is_same = not (R_svd-R_als).round(2).abs().all().all()
print(f'Is the ALS and SVD results the same? {is_same}')
```

```
Is the ALS and SVD results the same? True
```

Nous avons obtenu les même résultats avec ALS qu'avec SVD. Cependant, il est possible d'avoir plus de bruit avec ALS. En effet, ALS est un algorithme itératif qui converge vers une solution optimale. Il est donc possible que la solution ne soit pas optimale à chaque itération. Cependant, il est possible de réduire le bruit en augmentant le nombre d'itérations ou en ajoutant un critère de convergence avec un certain seuil.

2. Vous complétez le TP1 et obtenez des résultats meilleurs que ceux de l'article de Sarwar et al. (2001). Mis à part le fait que ce pourrait ne pas être les mêmes données, donnez au moins deux explications possibles. (7.5 pts,. 100 mots max.)

Puisque l'Étude fait une validation croisée, elle obtient un résultat plus représentatif (certains modèles performeront plus, mais simplement dû à la chance de la distribution aléatoire du split). Dans le labo, ne faisant pas de validation, il est possible qu'on ait une meilleure performance par chance car nous n'avons pas de validation croisée. Si on relance l'algorithme, il est possible que la performance du TP soit moins bonne, car on ne fait pas la moyenne sur plusieurs distributions d’entraînement et de tests