Mastère – ML 4 – Systèmes de recommandation
TP Netflix

Stéphane Canu
scanu@insa-rouen.fr, asi.insa-rouen.fr\~scanu

May 24, 2020

Practical session description

This practical session aims at showing how to build a simple recommender system on the Netflix data set[1], based on https://www.netflixprize.com/assets/ProgressPrize2008_BellKor.pdf and https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf.

Figure 1: Result of the netflix competition.

Ex. 1 — SVD and weighted SVD on Netflix

1. The Netflix dataset
   a) What was the Netflix competition?
   b) What is the score used to compare the models?
   c) download the Netflix training data and the probe set from moodle. Note that netflix_data_app.mat size is 232.1 Mo.
   d) load them into you python environment

```
import numpy as np
from time import time
from scipy.sparse import csr_matrix
from scipy.linalg import svd
from scipy.sparse.linalg import svds

import scipy.io

D = scipy.io.loadmat('netflix_data_app.mat')
P = scipy.io.loadmat('netflix_data_probe.mat')
```

   e) What is the size and the type of these data

```
print(D)
D.values()
```

   f) Build the associated data matrices, and explain the following piece of code.

[1] https://en.wikipedia.org/wiki/Netflix_Prize

```
M = D[’netflix_data_app’]
Mt = P[’netflix_data_probe’]
#help(M)
#print(M.__class__)
```

2. Pre process the data

a) Compute RMSE the root mean squared error (RMSE) on the probe set by predicting missing values by the global mean μ of the training data, that is:

$$\text{RMSE} = \sqrt{\frac{1}{n_t} \sum_{u,m; M_t(u,m) \neq 0} (M_t(u,m) - \mu)^2}.$$

How long does it takes to compute this error?

```
n = ???
moy = ???
nt = ???
pred = ???
Mt = Mt - pred
err0 = np.sqrt(np.sum(Mt.power(2))  / nt )
print(’ erreur : {0:.3f}’.format(err0))
```

b) Let M_c be the $n \times p$ centered sparse rating matrix (of general term $M_{u,m} - \mu$ when $M_{u,m} \neq 0$), n being the total number of user considered and p the number of movies. Let μ_m the mean score of for the movie m that is

$$\mu_m = \frac{1}{n_m} \sum_{u=1}^{n} M_{u,m},$$

where $n_m = \sum_{u=1}^{n} \mathbb{1}_{\{M_{u,m} \neq 0\}}$ denotes the number of users who have rated the movie m. What is the error made on the probe set by predicting the missing value by the global mean plus the mean of the movie?

c) What is the error made on the probe set by predicting missing value of the rating $P(m,u)$ given to the movie m by the user u, by summing the global mean μ plus the mean of the movie μ_m and the mean of the user μ_u, that is

$$P(m,u) = \mu + \mu_m + \mu_u$$

3. Recommend using SVD

a) Assuming the optimal number of factor to take into account is smaller than 100, as a first step of a SVD based recommender system, factorize the residual score matrix using the SVD. How long does it takes?

b) Explain what does this piece of code do.

```
from numpy.random import default_rng
ni = 3
k = 8
rng = default_rng()
U = rng.standard_normal((ni,k))
Vt = rng.standard_normal((k,1))
P = U*Vt.T
prediction = np.cumsum(P,1)
target = np.array([[0],[0],[0]])
err=np.sum((prediction - target)**2,0)
```

c) In one loop, predict the missing test values using the SVD with an increasing number of component and compute the associated RMSE, that is when the prediction is given by

$$P(m,u) = \sum_{k=1}^{n_k} U(u,k)V(m,k) + \mu + \mu_m + \mu_u$$

d) What is, in this case, the optimal number of factors? Have you got a better result than Cinematch scores, that is 0.9514 on the probe set?

4. Recommend using weighted SVD

a) Write a function to evaluate the performance of this approach on the test matrix.? How long does it takes to run?

```
def Netflix_probe_error(U,Vt,Maskt,Mt, Mmin,Mmax,nt):
'''
Compute the root mean squared error (RMSE) between Mt and U Vt
when Mt is a sparse matrix with sparsity mask given by Maskt.
'''
  p,n =  Mt.shape

  err = 0.
  for j in range(0,n):
       ...
           ???
           ...
       err+=np.sum(( ??? )**2)

return np.sqrt(err  / nt)
```

b) Improve the previous results by using the weighted SVD approach. Try with 50 factors and alternated least square

c) Improve again by using the SGD approach to factorization minimizing on the training set

$$\sum_{u,m;M_a(u,m)\neq 0} (M_a(u,m) - \sum_{k=1}^{n_k} U(u,k)V(m,k) + \mu + \mu_m + \mu_u)^2$$