Unsupervised Learning: from big data to low-dimensional representations,
Master MVA, 2017/2018

Project 1: Face Completion and Movie Recommendation Challenge

Pirashanth RATNAMOGAN , Othmane SAYEM

ratnamogan@ensta.fr, sayem@ensta.fr

0. This project is to be done in MATLAB or Phyton and in groups of three students. Please report which student did which part. Each student should also email me a grade (0-100%=best) for the work of his/her mates.

We did the projet in both MATLAB (Othmane) and Python (Pirashanth), it allows us to both work hard on the projet and understand all what has been done in this homework. Moreover, we have both worked in all the parts of the project (that's why you can find a Python and a Matlab version for each part of the homework), that have allowed us to compare our outcomes and help each other. For the report, Othmane did the part 2 of the project while Pirashanth worked on the part 3.

1. Low Rank Matrix Completion (LRMC). Implement Algorithm 3.2 according to the following format.

We have implemented two versions of the code. One in Matlab and the other one in Python. The Matlab version is more efficient and take less time to converge. We have used this two implementations to take benefits of the pair work and check which language performs better.

2. *Face Completion :* Remove uniformly at random 0, 20, or 40 % of the entries of all images of individual 1. Apply the Low Rank Matrix Completion (LRMC) algorithm to fill in the missing entries. Plot the completed faces and comment on the quality of completion as a function of the percentage of missing entries by visually comparing the original images (before removing the missing entries) to the completed ones

First, we apply the LRMC Algorithm to the whole set of 10 pictures of the individual 1, in ten different illuminations. We have reduced the size of all images by two for computational reasons. We have flatten each image matrices to a vector and made up our matrix to complete by concatenating all the image vectors as a matrix of size 7728*10. Using a Bernoulli distribution, we remove uniformly and randomly 20%, 40%, 60% and 80% of the data matrix elements, then we complete them using the Low Rank Matrix Completion algorithm that we previously coded in question 1.

However, as we discussed in class, in order to apply the theorem, we must check first that our matrix X is ν-Incoherent. Which means that, given $X = U\Sigma^T$, the compact SVD of matrix X :

$$\max_i ||u_i||_2 \leq \nu \frac{\sqrt{d}}{\sqrt{D}}$$
$$\max_i ||v_i||_2 \leq \nu \frac{\sqrt{d}}{\sqrt{N}}$$
$$||UV^T||_\infty \leq \nu \frac{\sqrt{d}}{\sqrt{N * D}}$$

Moreover as discussed in class if the number of missing values is too large in comparison with the number of samples, the algorithm will have difficulties to converge. We have only ten images, hence as we will see in the following tests, the algorithm will have difficulties to converge if the percantage of missing entries is too high. We fix $\beta = \min(2, \ )$. and we vary τ.

We collect the resulting images :

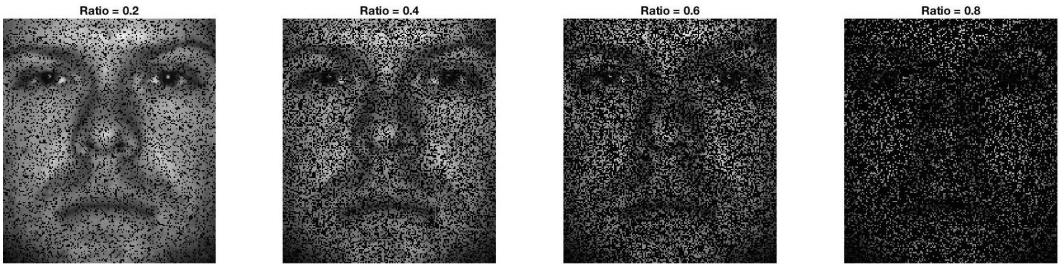

Figure 1: Face of person 1, in position 1, with percentage of missing entries

Figure 2: Face of person 1, in position 1, with percentage of missing entries : 20% and 40%

Figure 3: Reconstructed Face of person 1, in position 1(20%, 40%), with $\tau = 4 * 10^3$

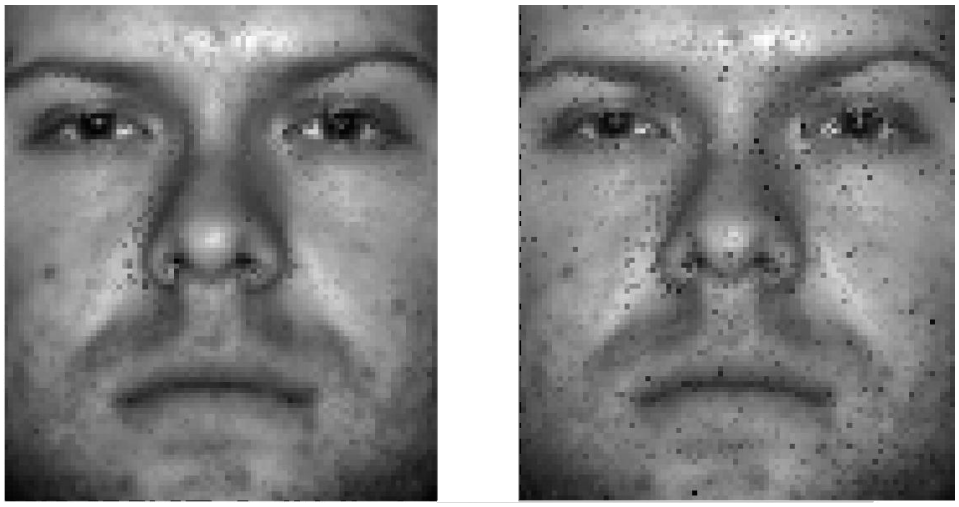

Figure 4: Reconstructed Face of person 1, in position 1(20%, 40%), with $\tau = 5 * 10^5$

In order to see clearly the effect of this parameter on the performance of the algorithm, we compute the Mean Square Error (MSE), for each percentage of missing entries and each choice of τ :

τ	Missing Entries	MSE
4*10^3	20%	987
4*10^3	40%	3133
4*10^5	20%	55.6
4*10^5	40%	225

Table 1: MSE vor various *tau* and percantage of missing values

As shown in the table above, the quality of the face reconstruction increases with the parameter τ. The best results are found for a value $\tau = 4 * 10^5$, we are able to reconstruct the image with 40% missing entries, within a reasonable time. However, we also tried taking even bigger values for τ, like 10^7 10^8, but it makes the convex optimization problem harder to solve, which results huge computation times (because of the SVD calculations..)

Same case for big percentages of missing entries, like 60% and 80%, Since we only have 10 illumination positions for the face, the algorithm is not really able to reconstruct the image. It runs for a very long time, without necessarily approaching the convergence..

3. Movie Recommandation CHallenge: In this exercise you will develop a new recommender system for movies. Given a matrix of incomplete user ratings for different movies, you will use low-rank matrix completion to predict the ratings for other movie. Apply Algorithm 3.2 to the Horror, Romance, and Horror+Romance training sets and report the mean-squared-error (MSE) on the held out test set ratings

1. Building the matrix of user ratings

To construct the matrix of user ratings using the `Movielens-Romance-Horror.csv` file we use the approach describe in the following figure.

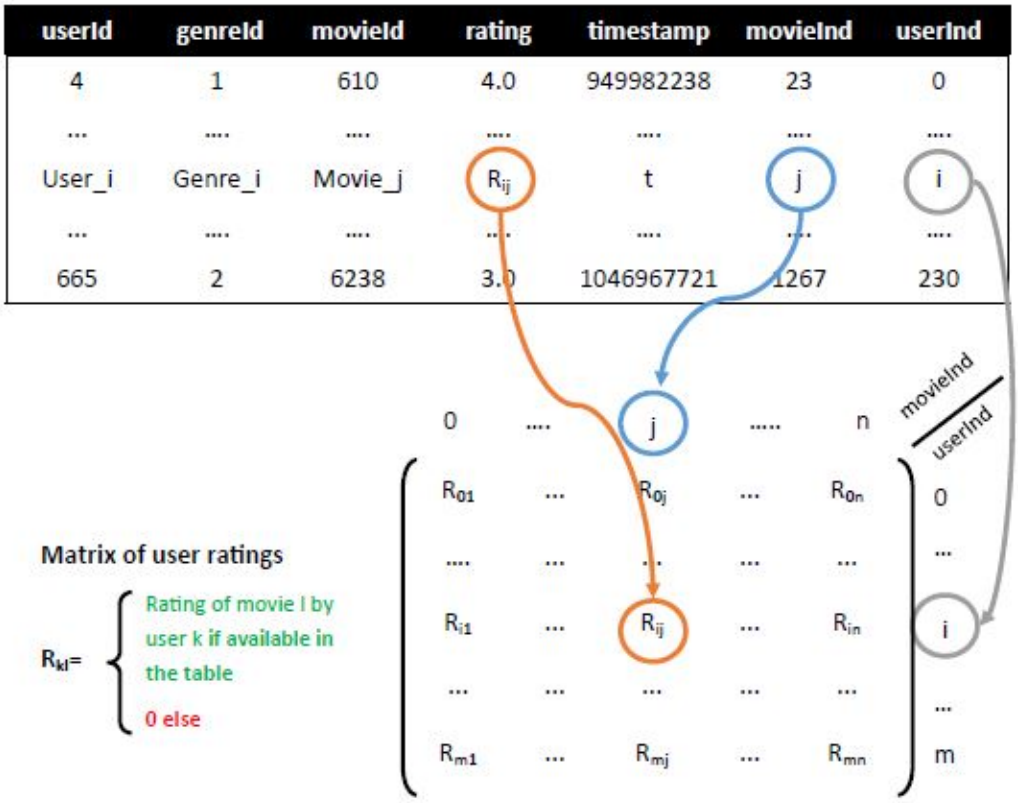

Figure 5: Approach to construct the matrix of user ratings

The matrix that we have build is of size (231,1615). If we actually want to construct part of the matrix corresponding to the genre `Horror` or the genre `Romance` we just have to use the operation described in 5 but with the database containing just the movies corresponding to the genre `Romance` or `Horror`.

2. Approach to evaluate the model

The matrix that we have built in the previous part is sparse and there is already a lot of missing values. However, we have no way to say that the reconstructed values correspond to the reality or not.

To evaluate our algorithm, we will proceed as follows:

- we will build a new incomplete matrix called M^{train} built removing a part of the ratings existing in the original matrix $M^{original}$ built thanks to the database.
- We will call Ω the space corresponding to the the indices that have been removed.
- We will compute the LRMC algorithm to M^{train}, the positions of the missing values would be define by all the positions of the 0 elements in M^{train}.

The LRMC algorithm will return a new completed matrix M^{test}. Using this matrix M^{test}. The final score of the various completions would be based on the mean squared error and on the mean absolute error and would be given using the previous notations by:

$$MSE_{score} = \frac{1}{|\Omega} \sum_{(i,j)\in\Omega} (M_{i,j}^{original} - M_{i,j}^{test})^2$$

$$MAE_{score} = \frac{1}{|\Omega} \sum_{(i,j)\in\Omega} |M_{i,j}^{original} - M_{i,j}^{test}|$$

3. Outcome

In the following outcomes we will use various number of parameters for the parameter τ and for the rate nb_mask (uniformly at randomly chosen) of combinations of users and movies used to compute the final evaluation. We will set the learning rate β to 2, and will denote by MAE and MSE the scores defined above. As discussed in the part about Face completion, we could use ν-incoherence, to test if the algorithm has proof of convergence or not.

	τ	β	nb_mask	MAE	MSE
0	10	2	0.2	3.167607	11.120511
1	10	2	0.4	3.227428	11.503177
2	10	2	0.6	3.292639	11.950174
3	100	2	0.2	2.194715	6.046141
4	100	2	0.4	2.397393	6.980971
5	100	2	0.6	2.686997	8.472760
6	1000	2	0.2	1.252231	2.464135
7	1000	2	0.4	1.377606	2.955410
8	1000	2	0.6	1.619353	3.859790
9	10000	2	0.2	1.006906	1.765337
10	10000	2	0.4	1.066507	1.953228
11	10000	2	0.6	1.221147	2.521395

Table 2: Outcome for Horror+Romance for various values of τ

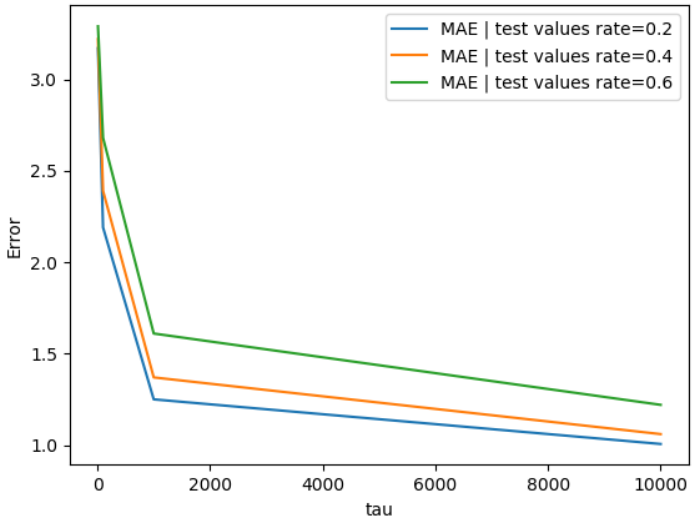

Figure 6: Evolution of the MAE score with the τ parameters and with the percantage of data used for the test for the full Data Base

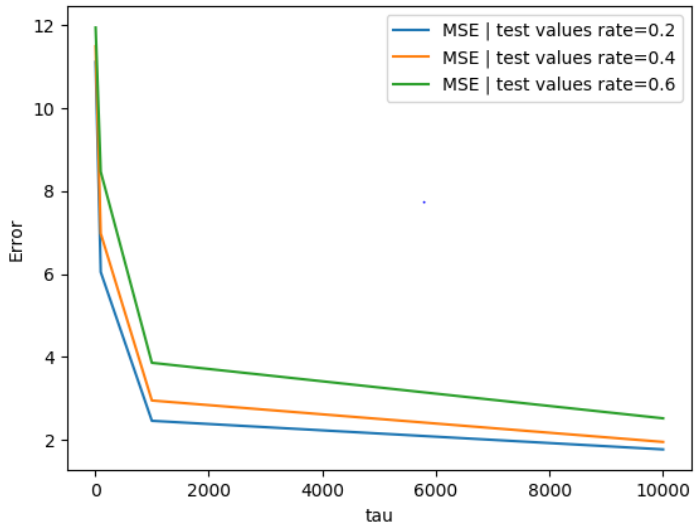

Figure 7: Evolution of the MSE score with the τ parameters and with the percantage of data used for the test for the full Data Base

	τ	β	nb_mask	MAE	MSE
0	10	2	0.2	2.999865	10.179820
1	10	2	0.4	3.096448	10.765478
2	10	2	0.6	3.169504	11.265868
3	100	2	0.2	2.080157	5.530890
4	100	2	0.4	2.351331	6.774850
5	100	2	0.6	2.645756	8.317101
6	1000	2	0.2	1.316931	2.759691
7	1000	2	0.4	1.448630	3.226871
8	1000	2	0.6	1.775455	4.462631
9	10000	2	0.2	1.147552	2.220284
10	10000	2	0.4	1.185681	2.382531
11	10000	2	0.6	1.398519	3.158672

Table 3: Outcome for genreId=1 for various values of τ

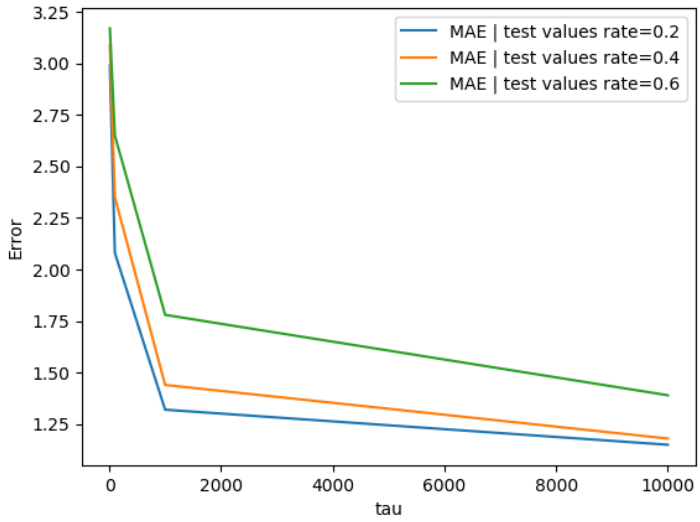

Figure 8: Evolution of the MAE score with the τ parameters and with the percantage of data used for the test for genreId=1

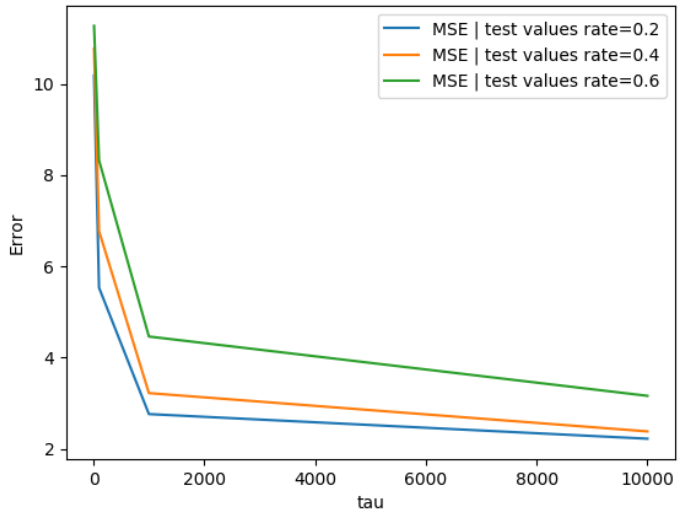

Figure 9: Evolution of the MSE score with the τ parameters and with the percantage of data used for genreId=1

	τ	β	taux_mask	MAE	MSE
0	10	2	0.2	3.254369	11.607362
1	10	2	0.4	3.314657	11.983846
2	10	2	0.6	3.378455	12.460092
3	100	2	0.2	2.140939	5.800344
4	100	2	0.4	2.362871	6.855983
5	100	2	0.6	2.702288	8.541588
6	1000	2	0.2	1.198992	2.329953
7	1000	2	0.4	1.348184	2.832371
8	1000	2	0.6	1.606741	3.819802
9	10000	2	0.2	1.015192	1.770456
10	10000	2	0.4	1.060555	1.962671
11	10000	2	0.6	1.199160	2.403928

Table 4: Outcome for genreId=2 for various values of τ

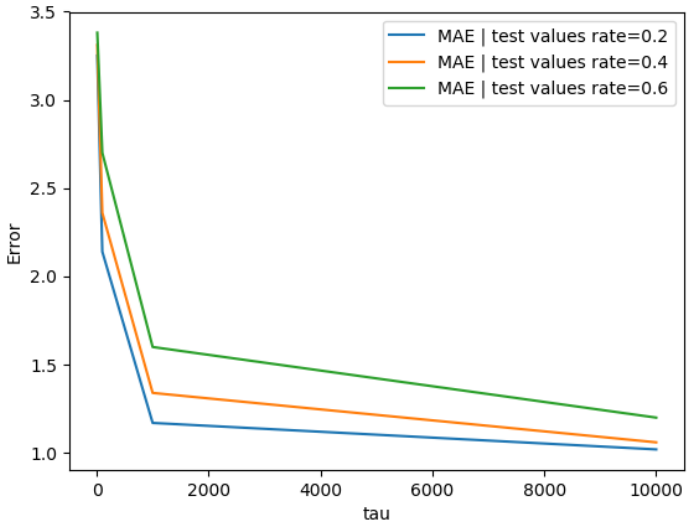

Figure 10: Evolution of the MAE score with the τ parameters and with the percantage of data used for the test for genreId=2

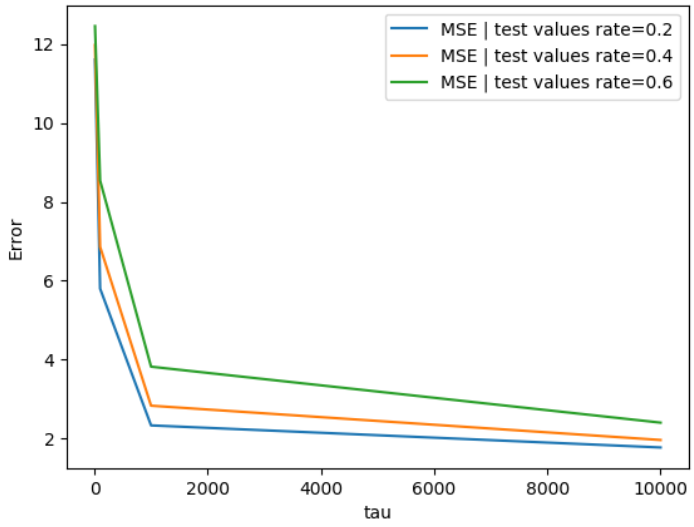

Figure 11: Evolution of the MSE score with the τ parameters and with the percantage of data used for genreId=2

As we can see by increasing the τ parameter we get relative good and fair outcomes. The main problem is when τ is increasing the computation time is increasing as well. We are stopping the algorithm if it hasn't converge after 10 000 iterations (the only case list is for genreId=2 and $\tau = 0.6$ Moreover we can remark than combining the two datasets of movies (Horror and Romances) allows to improve the final outcome in all the cases. Having additional data about the users always helps for the convergence but the computation of one iteration is more costly.