Factorization-Based Data Modeling
Practical Work 2

Umut Şimşekli
Télécom ParisTech, Université Paris-Saclay, Paris, France
umut.simsekli@telecom-paristech.fr

Instructions: (please read carefully)

1. This homework can be done in groups of **maximum 2** people.
2. Prepare your report as a pdf file in English by using LaTeX or a similar software (Word etc). Do not submit scanned papers.
3. Put all your files (code and/or report) in a zip file: *surname_name_tp2.zip* and submit it to `https://www.dropbox.com/request/Qz5XsTwv5SDcgd6XgXyj`. The deadline is **January 20th, 2020, 23.00**. Late submissions will not be accepted.
4. One submission per group is sufficient.

1 Stochastic Gradient Descent

In this section, you will implement the stochastic gradient descent algorithm for large-scale matrix factorization. The problem that we aim to solve is given as follows:

$$(W^\star, H^\star) = \arg\min_{W,H} \frac{1}{2}\|M \odot (X - WH)\|_F^2, \tag{1}$$

where $X \in \mathbb{R}^{I\times J}$ is the data matrix, and $W \in \mathbb{R}^{I\times K}$ and $H \in \mathbb{R}^{K\times J}$ are the unknown factor matrices. Here $\|A\|_F$ denotes the Frobenius norm of a matrix A and $\odot$ denotes element-wise multiplication. Finally, $M \in \{0,1\}^{I\times J}$ is the 'mask' matrix, denoting if a particular entry of X is observed or not: $m_{ij} = 1$ if x_{ij} is observed and $m_{ij} = 0$ otherwise.

1.1 Movie Recommendation

We will work on the MovieLens 1 Million dataset. This dataset contains ∼1 million ratings applied to $I = 3883$ movies by $J = 6040$ users, resulting in a sparse data matrix X with 4.3% non-zero entries. Our aim will be to decompose this matrix into W and H by only using its observed entries. Once we obtain estimates for W and H, we can then use them for predicting the unobserved entries of X, which will enable us to make recommendations.

1.2 Exercises

Now go to the file `matrix_factorization_template.m`

1. Complete the stochastic gradient algorithm.
2. At the end of each iteration, compute the roor-mean-squared-error, that is given as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\|M \odot (X - WH)\|_F^2}{N}} \tag{2}$$

   where N is the number of observed entries in X.

3. Play with the algorithm parameters, i.e. the step-size, the batch-size, initialization, and the rank of the factorization. What do you observe? How do the step-size and the batch-size interact?

4. After estimating W and H, use them to recommend a movie for a given user.

2 Distributed Stochastic Gradient Descent

In this part, the aim is to implement the Distributed Stochastic Gradient Descent (DSGD)[1] algorithm, which we covered earlier. You will implement the algorithm in C/C++ by using the OpenMPI and GSL libraries.

Throughout this practical work, we will only consider the usual matrix factorization problem, given as follows:

$$(Z_1^\star, Z_2^\star) = \underset{Z_1, Z_2}{\arg\min} \frac{1}{2} \| M \odot (X - Z_1 Z_2) \|_F^2 \tag{3}$$

where we have changed the notation from the earlier notes.

2.1 Exercises

In the following questions, we will work on the MovieLens 1 Million dataset. We will assume we have 4 processors, therefore the observed matrix will be partitioned into a 4×4 blocks.

1. Complete the file `dsgd_mf_template.cpp`.

2. Set the rank to 10 and the step size to 0.00001. Run the code for MovieLens 1 Million Dataset.

3. Compute the RMSE by using the code `compute_rmse.cpp` and plot the RMSE in Matlab by using `plot_rmse.m`.

4. Play with the rank and the step-size. What do you observe?

[1] Gemulla, Rainer, et al. "Large-scale matrix factorization with distributed stochastic gradient descent.", Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.