Tensor Factorisation for Group Detection

Ange Kokotakis, Romain Ramez

directed by Rodrigo Cabral Farias

December 2023

Abstract

The aim of this project is to detect groups of people with a given dataset of interactions in a community. For this purpose we will represent the dataset as a tensor and use non-negative tensor factorization to approximate it as a product of different matrix from which we can get information about groups in this community.

1 Introduction

In this project, we use datasets similar to Figure 1 in which each line represents an interaction between two individuals: $id1$ and $id2$ at a certain $time$.

time	id1	id2
31220	58	63
31220	59	64
31220	63	66
31220	85	190
31220	85	214
31220	102	115
31220	191	199
31220	191	214
31240	58	63
31240	63	66
31240	85	190
31240	85	214
31240	102	115
31240	143	192
31240	188	194
31240	191	199

Figure 1: dataset

From this dataset we create matrices $\mathbf{X} \in \{0,1\}^{I \times I}$ which represent interactions between people during a certain interval of time where I is equal to the number of people in the community and each coefficient is equal to :

$$\mathbf{X}_{i,j} = \begin{cases} 1 & \text{if there is an interaction between person } i \text{ and } j \text{ during a given interval of time} \\ 0 & \text{otherwise} \end{cases}$$

And then we stack these matrices with different interval t_k of time to create our tensor $\mathbf{Y} \in \{0,1\}^{I \times I \times K}$ where K is the number of intervals of time. Finally the tensor containing the dataset has its coefficients equal to :

$$\mathbf{Y}_{i,j,k} = \begin{cases} 1 & \text{if there is an interaction between person } i \text{ and } j \text{ at interval of time } t_k \\ 0 & \text{otherwise} \end{cases}$$

2 Model

In order to approximate this tensor we create 3 matrices :

$$\mathbf{U} \in \mathbb{R}_+^{I \times R},\ \mathbf{V} \in \mathbb{R}_+^{I \times R},\ \mathbf{W} \in \mathbb{R}_+^{K \times R}$$

where :

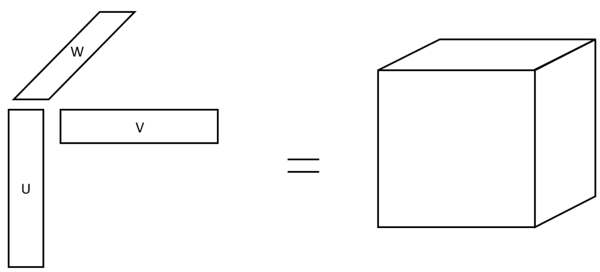
- I is the number of individuals in the community,
- K is the number of interval of time,
- R is the number of groups in the community (we choose one randomly at first we will see later how to choose it correctly).

The U and V matrices represent the membership level of a person to a certain group and W represents in which interval of time a group has been active.

We define $\mathbf{S} \in \mathbb{R}_+^{I \times I \times K}$ as :

$$\mathbf{S}_{i,j,k} = \sum_{r=1}^{R} \mathbf{U}_{i,r}\mathbf{V}_{j,r}\mathbf{W}_{k,r}$$

The result of the product $\mathbf{U}_{i,r}\mathbf{V}_{j,r}\mathbf{W}_{k,r}$ is called the outer product noted $\mathbf{U}_r \circ \mathbf{V}_r \circ \mathbf{W}_r$ which can be seen as :



Finally S is define as :

$$\mathbf{S} = \sum_{r=1}^{R} \mathbf{U}_r \circ \mathbf{V}_r \circ \mathbf{W}_r$$

In order to approximate Y with S we define a cost fonction $L(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \|\mathbf{Y} - \mathbf{S}\|_F^2$ where $\|.\|_F$ represents the Frobenius norm, moreover because U and V are representing the same thing we want these matrices to be similar so we add to L another term $\lambda\|\mathbf{U} - \mathbf{V}\|_F^2$. Finally L is defined by :

$$L(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \|\mathbf{Y} - \mathbf{S}\|_F^2 + \lambda\|\mathbf{U} - \mathbf{V}\|_F^2, \lambda \in \mathbb{R}_+$$

L can be rewritten by using unfolded forms of $\mathbf{Y}$ i.e. by transforming it in a matrix with all the layers of $\mathbf{Y}$ arranged in different ways and by using Khatri-Rao product $\odot$:

$$\begin{aligned} L(\mathbf{U}, \mathbf{V}, \mathbf{W}) &= \|\mathbf{Y}^{(1)} - \mathbf{U}(\mathbf{W} \odot \mathbf{V})^T\|_F^2 + \lambda\|\mathbf{U} - \mathbf{V}\|_F^2 \\ &= \|\mathbf{Y}^{(2)} - \mathbf{V}(\mathbf{W} \odot \mathbf{U})^T\|_F^2 + \lambda\|\mathbf{U} - \mathbf{V}\|_F^2 \\ &= \|\mathbf{Y}^{(3)} - \mathbf{W}(\mathbf{V} \odot \mathbf{U})^T\|_F^2 + \lambda\|\mathbf{U} - \mathbf{V}\|_F^2 \end{aligned}$$

To minimize this function we will use two different method : the multiplicative update algorithm (MU) and the hierarchical alternating least square (HALS) and these forms of L will be very useful.

3 Multiplicative Update

The principle of this algorithm is to alternatively minimize L with respect to each of its components by using a version of gradient algorithm.

At each iteration we calculate :

$$\nabla L = \begin{bmatrix} \nabla_{\mathbf{U}} L \\ \nabla_{\mathbf{V}} L \\ \nabla_{\mathbf{W}} L \end{bmatrix} = \begin{bmatrix} [\nabla_{\mathbf{U}} L]^+ - [\nabla_{\mathbf{U}} L]^- \\ [\nabla_{\mathbf{V}} L]^+ - [\nabla_{\mathbf{V}} L]^- \\ [\nabla_{\mathbf{W}} L]^+ - [\nabla_{\mathbf{W}} L]^- \end{bmatrix}$$

Where all the $[.]^+, [.]^-$ are positive

Then we update $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$:

$$\begin{aligned} \mathbf{U}_k &= \mathbf{U}_{k-1} \boxdot ([\nabla_{\mathbf{U}} L]^+ \boxslash [\nabla_{\mathbf{U}} L]^-) \\ \mathbf{V}_k &= \mathbf{V}_{k-1} \boxdot ([\nabla_{\mathbf{V}} L]^+ \boxslash [\nabla_{\mathbf{V}} L]^-) \\ \mathbf{W}_k &= \mathbf{W}_{k-1} \boxdot ([\nabla_{\mathbf{W}} L]^+ \boxslash [\nabla_{\mathbf{W}} L]^-) \end{aligned}$$

Where $\boxdot$ and $\boxslash$ are the Hadamard product and division respectively. To avoid division by zero, we replace all the zero coefficient by an epsilon strictly positive.

4 Hierarchical Alternating Least Square

This algorithm minimize L with respect to one column of $\mathbf{U}$, $\mathbf{V}$ or $\mathbf{W}$ at a time, all the other column are fixed to their previous approximation for instance with the column $\mathbf{U}_{r'}$:

$$\begin{aligned} &L = \|\mathbf{Y} - \mathbf{S}\|_F^2 + \lambda \|\mathbf{U} - \mathbf{V}\|_F^2 \\ &= \sum_{ijk} (\mathbf{Y}_{ijk} - \sum_{r \neq r'} (\mathbf{U}_{ir} \mathbf{V}_{jr} \mathbf{W}_{kr}) - \mathbf{U}_{ir'} \mathbf{V}_{jr'} \mathbf{W}_{kr'})^2 + \lambda \sum_i \sum_{r \neq r'} (\mathbf{U}_{ir} - \mathbf{V}_{ir})^2 + \lambda \sum_i (\mathbf{U}_{ir'} - \mathbf{V}_{ir'})^2 \\ &= \sum_i \mathbf{U}_{ir'}^2 (\lambda I + \sum_{jk} (\mathbf{W}_{kr'} \mathbf{V}_{kr'})^2) - 2\mathbf{U}_{ir'} (\lambda \mathbf{V}_{ir'} + \sum_{jk} \mathbf{W}_{kr'} \mathbf{V}_{kr'} (\mathbf{Y}_{ijk} - \sum_{r \neq r'} (\mathbf{U}_{ir} \mathbf{V}_{jr} \mathbf{W}_{kr}))) + C, \text{C independent of } \mathbf{U}_{ir'} \end{aligned}$$

So L is minimum with respect to $\mathbf{U}_{ir'}$ if $\mathbf{U}_{ir'} = \dfrac{\lambda \mathbf{V}_{ir'} + \sum_{jk} \mathbf{W}_{kr'} \mathbf{V}_{kr'} (\mathbf{Y}_{ijk} - \sum_{r \neq r'} (\mathbf{U}_{ir} \mathbf{V}_{jr} \mathbf{W}_{kr}))}{\lambda I + \sum_{jkr'} (\mathbf{W}_{kr'} \mathbf{V}_{kr'})^2}$

We can generalize this result to an entire column : $\mathbf{U}_{:,r'} = \frac{\lambda \mathbf{V}_{:,r'} + \sum_{jk} \mathbf{W}_{kr'} \mathbf{V}_{kr'} (\mathbf{Y}_{:,jk} - \sum_{r \neq r'} (\mathbf{U}_{:,r} \mathbf{V}_{jr} \mathbf{W}_{kr}))}{\lambda I + \sum_{jkr'} (\mathbf{W}_{kr'} \mathbf{V}_{kr'})^2}$, we can find the two other update by symetry for $\mathbf{V}$ and by doing the same calcul for $\mathbf{W}$.

At each iteration we update $\mathbf{U}_{r'}$, $\mathbf{V}_{r'}$, $\mathbf{W}_{r'}$, then we increment r' until we reach our convergence criterion or our maximum number of iteration.

5 Comparison of MU and HALS convergence

We note that HALS converge faster than MU.

6 Result

Now that we have code the MU factorization, we can compare our results with the exact solution. For that, we will use several R for several interval of time t_k and change the number of iteration and see what happened.

In this three first pictures, we can see the evolution of our model when we change the number of iteration.

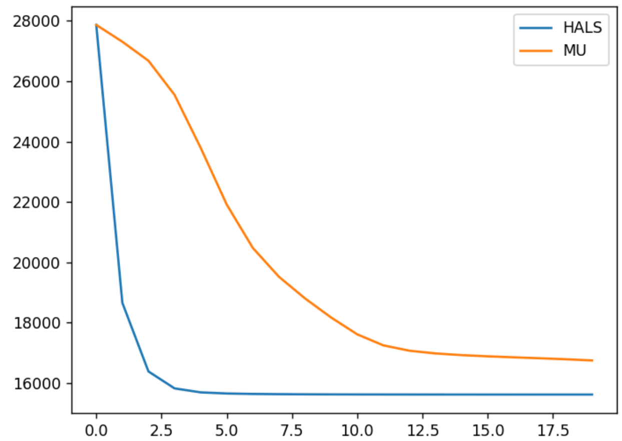


Figure 2: comparison between MU and HALS convergence

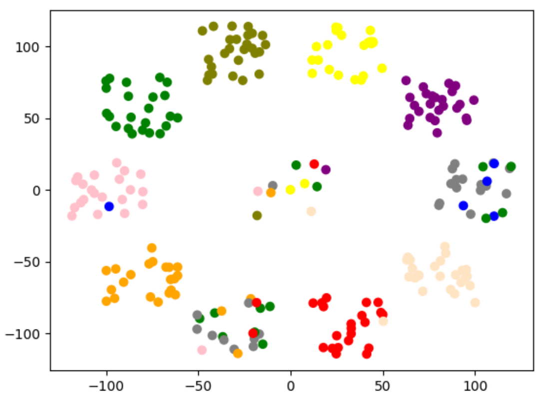


Figure 3: MU with R= 10, t_k= 3600 and Nb_it=50

In the first one, with Nb_it=50, we have a lot of person in the wrong group. We can count that ten persons are in another group and there's a full group made of people from others group. This model is clearly not that good.

It the second one, there's few people in the wrong group by still a group made by people from two other, and this time we can also see that there's two group in the same color, meaning that, in the model, the two groups are the same group. Finally, the result is better but we still have to improve this.

The third one is really better. Almost everyone is in his group, except the same group that is made of 2 groups, but this time we can see that there's more people from the "purple one", meaning that the group belong to them and so we can conclude that the "pinks" and the "purples" have a lot of contact with each other.

We can see that more the number of iteration is, more the result is close to the exact scheme. Showing the importance of this parameter.

In this three pictures, we will the how the parameter R can change the result of our model.

In the picture 5, we have R = 5, which means that we try to put a larger number of group in only 5

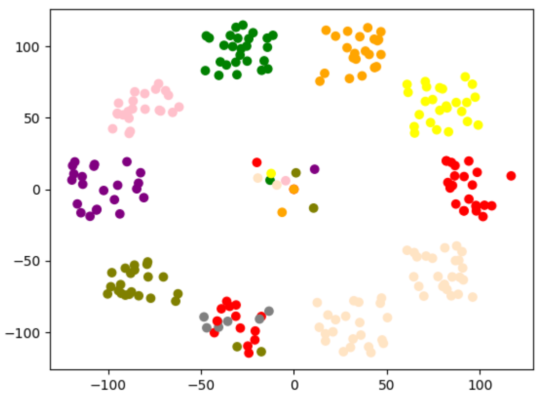


Figure 4: MU with R= 10, t_k= 3600 and Nb_it=150

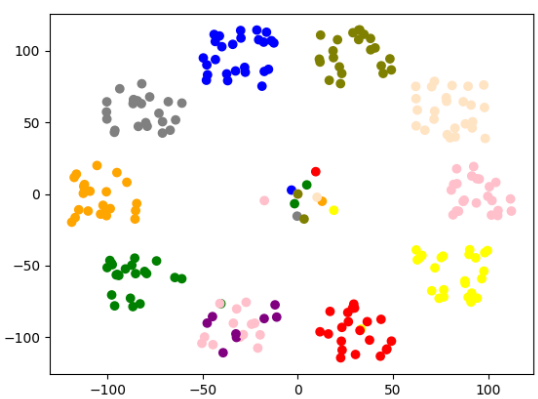


Figure 5: MU with R = 10, t_k = 3600 and Nb_it =500

groups. That's why we have lots of group in the same color. That's bad but the fact that we have five "well" separated show that the model if still efficient. We can see that all the people from each group are still together in one of the five "big" groups.

In the picture 6, the result is way better. We can see ten distinct groups and almost everyone in his group. In fact, ten is the real number of group in this case. Which means ten is the minimal number for R where we can see all the group well separated.

In the last picture, we still have ten separated groups but this time the separation is better. We have only three person in the wrong group over 242 persons. We can conclude on the efficiency of the model !

Finally, we can see that's we have a minimum for R, which is the real number of group but passed this number, we can grown it however we want and it will always increase the model.

Now, we have change the interval of time t_k.

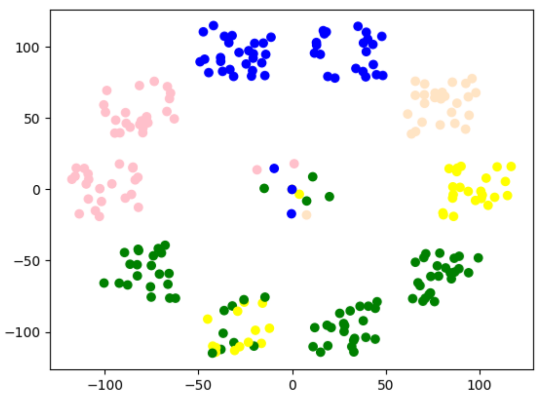


Figure 6: MU with R= 5, t_k= 3600 and Nb_it=500

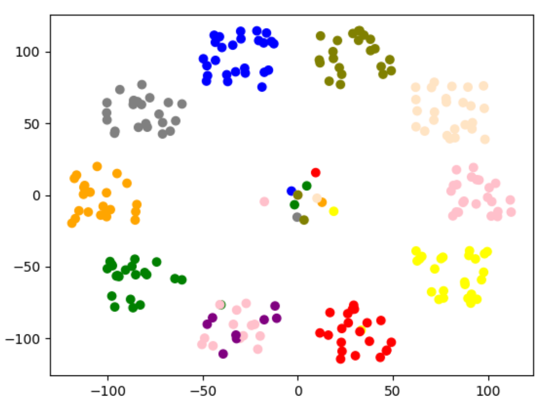


Figure 7: MU with R= 10, t_k= 3600 and Nb_it=500

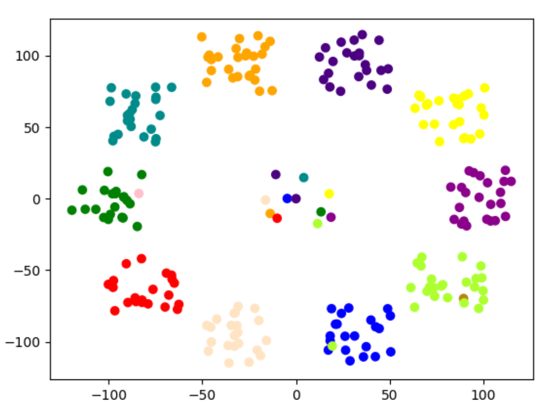


Figure 8: MU with R= 15, t_k= 3600 and Nb_it=500

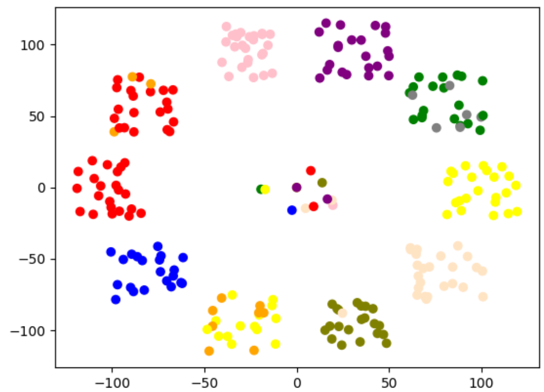


Figure 9: MU with R= 10, t_k= 900 and Nb_it=500

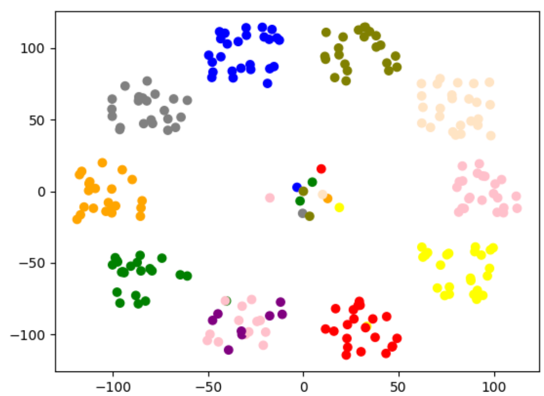


Figure 10: MU with R= 10, t_k= 3600 and Nb_it=500

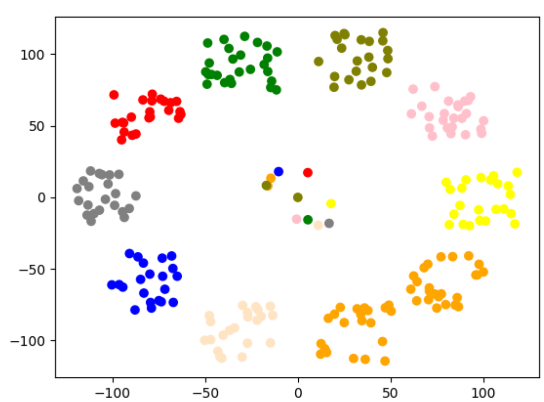


Figure 11: MU with R= 10, t_k= 18000 and Nb_it=500