ENSAE ParisTech

Compressed Sensing

Teachers : Guillaume LECUE

MATRIX COMPLETION ON GRAPHS

Victor JOURNE, Neil AMHIS

Contents

List of Figures

I Introduction

The aim of matrix completion is to find the values given a few of its entries. This is a topic of interest since Candes' article a few years ago. The aim of our project is to perform some matrix completion using a innovative graph-based approach. Graph theory is the current biggest topic in data science and machine learning as many social media and web communities provide large-scale graphs and one wants to be able to build recommender systems. We start describing the main principles of matrix completion on graphs, then we present a toy example simulating a synthetic netflix dataset in order to build a simplified recommender system.

II Matrix completion

II.1 Matrix completion problem

Given a sparset set Ω of observations $M_ij : (i,j) \in \Omega \subseteq \{1, \cdots, n\} * \{1, \cdots, n\}$. Our aim is going to be able to solve the following minimization problem :

$$\min_{X \in \mathbb{R}^{m*n}} ||X||_* \text{ s.t } .\mathcal{A}_\Omega(X) = \mathcal{A}_\Omega(M)$$

where $||X||_*$ is defined as the classical nuclear norm. The previous optimization programm suggests there are no noise in our observations. This hypothesis is obviously wrong so we add an other term in order to reformulate the problem :

$$\min_{X \in \mathbb{R}^{m*n}} \gamma_n ||X||_* + l(\mathcal{A}_\Omega(X), \mathcal{A}_\Omega(M)) \text{ s.t } .\mathcal{A}_\Omega(X) = \mathcal{A}_\Omega(M)$$

where l is chosen depending on the type of noise considered. In our case, we'll consider the squared Frobenius norm :

$$l(\mathcal{A}_\Omega(X), \mathcal{A}_\Omega(M)) = ||\mathcal{A}_\Omega \circ (X - M)||_F^2$$

Let's note that the way we formulate our problem suggests we did the following assumption : elements of Ω are well distributed . In a recommender system, this implies each client rated the same number of items, and each item is rated by an equal number of client. Thus it's going to be the main flaw of our toy example in the next section

II.2 Graph-based matrix completion

Given a matrix X of dimension $m * n$ we have acess to $k \ll m.n$ entries and the goal is the prediction of the rest unobserved ones. The low rank hypothesis implies the linear dependences of row/columns of X. The specificity of the graph-based approach is to assume the row of X are given on vertices of graph. Formally, let us be given an undirected graph $\mathcal{G} = (V, E, W)$ on the rows with vertices $V = \{1, \cdots, m\}$, edges $E \subseteq V * V$ and non-negative weights on the edges represented by the symmetric matrix

$m*m$ called W. The weights can be interpretated as follows : they capture the strength of association between row elements. We then define positive semidefinite matrix L defined as $D - W$ where is defined such that :$(D)_{ij} = diag(\sum_{j=1}^{m} W_{ij})$.

The problem of matrix completion can be formulated as follows :

$$\min_X \gamma_n ||X||_* + l(\mathcal{A}_\Omega(X), \mathcal{A}_\Omega(M)) + \frac{\gamma_r}{2}||X||^2_{\mathcal{D},r} + \frac{\gamma_c}{2}||X||^2_{\mathcal{D},c}$$

where $||X||^2_{\mathcal{D},c} = tr(XL_cX^t)$[1].

II.3 Optimization

Let recall or optimization problem :

$$\min_X \gamma_n ||X||_* + l(\mathcal{A}_\Omega(X), \mathcal{A}_\Omega(M)) + \frac{\gamma_r}{2}||X||^2_{\mathcal{D},r} + \frac{\gamma_c}{2}||X||^2_{\mathcal{D},c}$$

This can't be tackled efficiently with classical algorithms. Though we did not implement it in the notebook [2] we present the *Alternating Direction Method of Multipliers* algorithm. This algorithm solves the problem in the form

$$\begin{gathered}\text{minimize } F(X) + G(Y) \\ \text{subject to } AX + BY = C\end{gathered}$$

In our case, we set : $F(X) = \gamma_n ||X||_*$ and $G(Y) = \frac{1}{2}||\mathcal{A}_\Omega \circ (Y - M)||^2_{\mathcal{F}} + \frac{\gamma_r}{2}||Y||^2_{\mathcal{D},r} + \frac{\gamma_c}{2}||Y||^2_{\mathcal{D},c}$. We can reformulate :

$$\underset{X,Y \in \mathbb{R}^{m*n}}{min} \; F(X) + G(Y) \text{ s.t } X = Y$$

We consider the augmented Lagrangian : $\mathcal{L}_\rho(X, Y, Z) = F(X) + G(Y) + \langle Z, X - Y \rangle + \frac{\rho}{2}||X - Y||^2_{\mathcal{F}}$ and the update rule is given by :

- $X^{k+1} = \underset{X}{argmin} \mathcal{L}(X, Y^k, Z^k)$
- $Y^{k+1} = \underset{Y}{argmin} \mathcal{L}(X^{k+1}, Y, Z^k)$
- $Z^{k+1} = Z^k + \rho(X^{k+1} - YK + 1)$

ADMM is more general than other methods in the sense that the loss function doesn't need to be differentiable. For example, traditional methods do not work for optimization with l_1 regularization while ADMM can handle it easily.

[1]This term a a graph smoothing regularization, we want the neighbors are well-represented by the rows of the matrix

[2]Actually we'll try and present it for the oral examination

III Implementation : Synthetic netflix dataset

III.1 Matrix construction

We try and implement a matrix recovery model with a simulated Netflix-like dataset. The aim is to model a simple recommender system on netflix. Given a fixed percentage of observations the matrix users and movies, we want to recover all the values i.e the rating on each movie for each user. As we use the *CVXPY* package, we need to take care of the *Discipline convex programming rules*. Thus we implemented a little change of basis to calculate the terme $tr(XL_cX^t)$. Let L_c the positive semi-definite lagrangian matrix. We can write : $L_c = P\Lambda P^t$ where $\Lambda = diag(\lambda_i)_{i\in(1,\cdots,n)}$.Thus $XL_cX^t = XP\Lambda(XP)^t$:

Let $\lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}$ then we have :

$$\Lambda = \begin{pmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_n} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_n} \end{pmatrix} = \Sigma\Sigma$$

We can reformulate the trace : $tr(XP\Lambda(XP)^t)$ and we obtain $tr(XP\Lambda(XP)^t) = tr(X_0X_O^t)$ where $X_0 = XP\Sigma$. By reformulating the trace, we can implement the algorithm as it follows the *DCP rules*. We are now able to implement a matrix-completion algorithm. All the results are in the notebook.

III.2 Algorithm

Our matrix completion algorithm is based on the optimization program above. We compare different methods. We test our method and compare it to the standard nuclear norm-based matrix completion and to a method that uses only the graph. This corresponds to put the values of γ_c and γ_r to 0 or to put $\gamma_n = 0$.

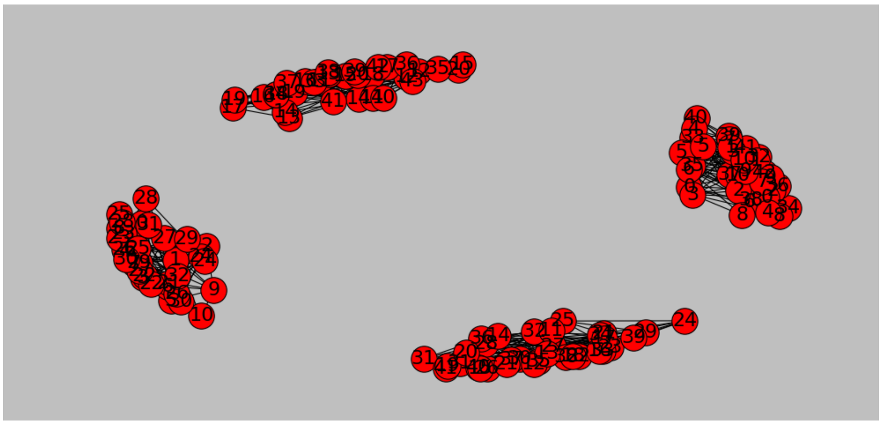

Figure 1: Graph of user communities

We try and reconstruct the matrix using different percentage of observed values. At this stage, the aim is not to describe the same result as the article or to be redundant with the notebook. Note that we did not connect the node within a community using a 3-nearest-neighbors graph. Moreover, in the modelling of the noise, we did not specify if an erroneous edge is both a wrong link or the absence of a real edge. Below, a zoom on one community.

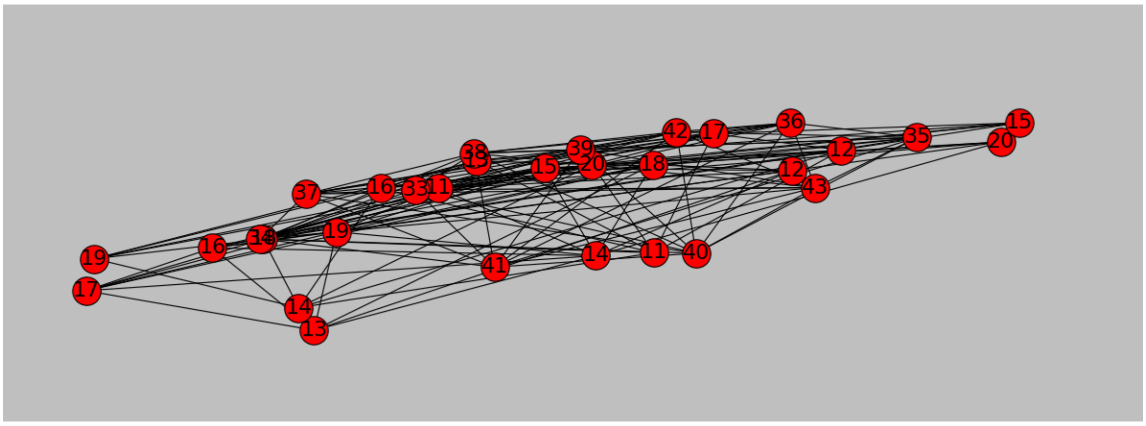

Figure 2: Graph of user communities

Like in the article, we reported the reconstruction root mean squared error with 10%, 20% and 30% of erroneous edges. Obviously, noisy graph[3] will not perfom as well as nuclear norm reconstructions.

IV Conclusion

In a word, we managed to implement a quite efficient matrix completion algorithm using graphs. This is evidence indeed our model is not applicable directly on a real dataset for obvious reasons. However, it's interesting to work on matrix completion on graphs as many web-dataset provide graph-based data, particularly social media and web-communities such as Amazon and Netflix. The main possible improvement of this model would be to consider -[2]- a robust on-line matrix completion algorithm using graphs. In our particular case, the dataset evolves over time and Netflix data scientist have to be able to check out the evolution of each community of users in order to improve the recommender system.

[3]Note that we use an operator OR to model noise, we replace arbitrary values by wrong values

V Bibliography

[1] Vassilis Kalofolias , *Matrix Completion on graphs* NIPS 2014 workshop "Out of the Box: Robustness in High Dimension"

[2] Symeon Chouvardas, *Robust on-line matrix completion on graphs* Mathematical and Algorithmic Sciences Lab