Structured Graph Learning via Laplacian Spectral Constraints

Paul MARTIN - Samuel DIAI

April 2, 2021

1 Introduction

The aim of Graph Learning is to determine a graph structure which represents smartly the connections between the variables. In general, graph algorithms like spectral clustering rely on a hand-crafted similarity function between the data and then use a simple heuristic like k-Nearest Neighbors (kNN) or ϵ-thresholding to make the representation sparse. This pre-processing step based on heuristics plays a crucial role in the algorithms performance. The article [Kum+19] proposes a new algorithm that both learns the the graph connectivity (if an edge is present) and the parameters (the edges weights). The characterization of the problem relies on the spectral properties of the Laplacian matrix and can be seen as an eigenvalue problem.

In this project, we implement the algorithm proposed by [Kum+19], and apply it to synthetic dataset like the 'two moons' or 'circle' dataset then to real-world data characterizing whether animals or cancer type. We also try to extend this approach to time series. The code and notebooks reproducing the experiments can be found on GitHub : https://github.com/SamuelDiai/SGL.

2 Mathematical background

2.1 Graph notations and constraints

We consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, \dots, p\}$ is the set of nodes and $\mathcal{E}$ is the set of edges. We define the adjacency matrix of the graph $A \in \mathbb{R}^{p \times p}$ containing the weights of the edges between the nodes. The degree matrix is then defined as $D = \text{Diag}(\sum_i A_{i,j})$ and finally the Laplacian of the graph is $L = D - A$. The Laplacian is a positive definite matrix that belongs to the following set :

$$\mathcal{S}_\Theta = \{\Theta | \Theta_{ij} = \Theta_{ji} \leq 0 \; \forall i \neq j, \; \Theta_{ii} = -\sum_{j \neq i} \Theta_{ij}\}$$

The set $\mathcal{S}_\Theta$ has $p(p-1)/2$ degrees of freedom therefore we can introduce a linear operator $\mathcal{L} : w \in \mathbb{R}_+^{p(p-1)/2} \to \mathcal{L}w \in \mathcal{S}_\Theta$ which is bijective and defined as :

$$[\mathcal{L}w]_{ij} = \begin{cases} -w_{i+d_j} & i > j, \\ [\mathcal{L}w]_{ji} & i < j, \\ -\sum_{i \neq j} [\mathcal{L}w]_{ij} & i = j, \end{cases}$$

The goal of the algorithm is to learn a k-component graph, that is to say that the node can be partitioned in k disjoint subsets with no edges between these sets. In order to have such a graph, the Laplacian need to have exactly k null eigenvalues. We characterize this constraint on the eigenvalues of a matrix thanks to the following set :

$$\mathcal{S}_\lambda = \{\{\lambda_i = 0\}_{i=1}^k, \; c_1 \leq \lambda_{k+1} \leq \dots \leq \lambda_p \leq c_2\}$$

2.2 Optimization problem

We denote $S \in \mathbb{R}^{p\times p}$, the sample covariance matrix (SCM). We try to solve the following optimization problem :

$$\begin{array}{ll} \text{maximize} & \log\ \text{gdet}(\Theta) - \text{tr}(\Theta S) - \alpha h(\Theta) \\ \text{subject to} & \Theta \in \mathcal{S}_\Theta,\ \lambda(\Theta) \in \mathcal{S}_\lambda, \end{array} \tag{1}$$

With gdet corresponding to the general determinant i.e the product of the positive eigenvalues and h is a penalization function with α controlling its importance. This problem comes from probabilistic graphical model, if $x \sim \mathcal{N}(0, \Theta)$, then it can be seen as a maximum likelihood estimation of the inverse covariance matrix where the graph $\mathcal{G}$ is estimated thanks to Θ. It is a direct extend of the Graphical Lasso Algorithm which requires the Gaussian assumption [FHT08].

With an arbitrary distribution this problem still makes sense since it can be interpreted as a minimization of a Bregman divergence problem. We can use the linear mapping $\mathcal{L}$ to provide a better formulation. The regularization term $h(\Theta)$ can be rewritten as $\text{tr}(\mathcal{L}w)$ therefore the two regularization terms are then equivalent to $\text{tr}(\mathcal{L}wK)$ with $K = S + \alpha(I - 11^T)$. We also introduce $U \in \mathbb{R}^{p\times p}$ such that $UU^T = I$ and $U\text{Diag}(\lambda)U^T$ but instead of adding this equality constraint to the problem, we only add a penalty term $\frac{\beta}{2}\|\mathcal{L}w - U\text{Diag}(\lambda)U^T\|_F^2$ to the objective function leading to the following formulation :

$$\begin{array}{ll} \underset{w,\lambda,U}{\text{minimize}} & -\log \text{gdet}(U\text{Diag}(\lambda)U^T) + \text{tr}(K\mathcal{L}w) + \dfrac{\beta}{2}\|\mathcal{L}w - U\text{Diag}(\lambda)U^T\|_F^2, \\ \text{subject to} & w \geq 0,\ \lambda \in \mathcal{S}_\lambda,\ U^TU = I_p \end{array} \tag{2}$$

Since we have k null eigenvalues, by denoting $q = p - k$ the degree of freedom of λ, the optimization problem 2 deals only with the following variables : $(w \in \mathbb{R}^{p(p-1)/2}, \lambda \in \mathbb{R}^q, U \in \mathbb{R}^{p\times q})$.

2.3 Sequential update algorithm

In order to solve the problem 2, the authors of [Kum+19] proposed an algorithm based on block successive upper-bound minimization. The idea is to update sequentially the variable w, U and λ by solving sub-problems. Each sub-problem is deduced from 2 by only keeping the variable of interest (the other are fixed) and the corresponding part of the objective function. To update the variable w, we solve the following convex problem with $c = \mathcal{L}^\star(U\text{Diag}(\lambda)U^T - \beta^{-1}K)$:

$$\underset{w\geq 0}{\text{minimize}} \quad f(w) = \frac{1}{2}\|\mathcal{L}w\|_F^2 - c^Tw \tag{3}$$

By deriving the KKT conditions of this problem, we have the following update :

$$w^{t+1} = \max(w^t - \frac{1}{2p}\nabla f(w^t), 0) \tag{4}$$

Similarly, U is obtained by solving the following eigenvalue problem :

$$\underset{U}{\text{maximize}} \quad \text{tr}(U^T\mathcal{L}wU\text{Diag}(\lambda)) \quad \text{s.t} \quad U^TU = I_p \tag{5}$$

From it, we deduce the update of U:

$$U^{t+1} = \text{eigenvectors}(\mathcal{L}w)[k+1:p]$$

Finally, for λ, we have only q degrees of freedom since the first k eigenvalues are null. By denoting $d \in \mathbb{R}^q$ such that d_i is the i-th element of $\text{Diag}(U^T\mathcal{L}wU)$. We solve the following problem :

$$\underset{c_1 \leq \lambda_1 \leq \cdots \leq \lambda_q \leq c_2}{\text{minimize}} \quad -\sum_{i=1}^{q} \log \lambda_i + \frac{\beta}{2} \|\lambda - d\|_2^2 \tag{6}$$

It can be solved by quadratic programming but [Kum+19] proposed an iterative implementation is less than q steps that gives a solution based on KKT conditions. Finally the global algorithm consists in updating successively the three variables w, U and λ until convergence. We then output $\mathcal{L}w$, the estimated laplacian of the graph that corresponds to the representation learnt from the data.

3 Experiments

3.1 Synthetic dataset

We first apply the algorithm on the dataset 'two moons' with 100 samples. We choose $k = 2$ since there are two clusters in this dataset and use $\beta = 10^3$, $\alpha = 0$. We notice in figure 1 that the two clusters are distinct. Indeed there are few edges between them and here we only have one connected component while we precise $k = 2$, it's due to the relaxation of the equality constraint ($\mathcal{L}w = U\text{Diag}(\lambda)U^T$) in the minimization problem : we only add a penalty term to the objective. In figure 2, we remark that the objective is decreasing at each iteration of the algorithm and converges toward a minimum.

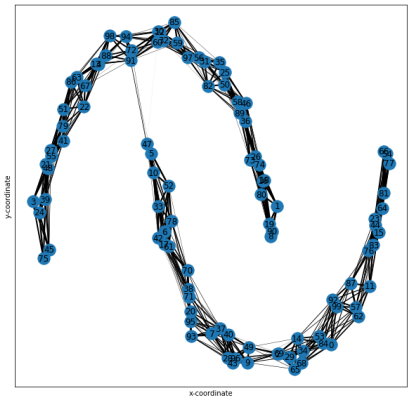

Figure 1: Learnt graph structure

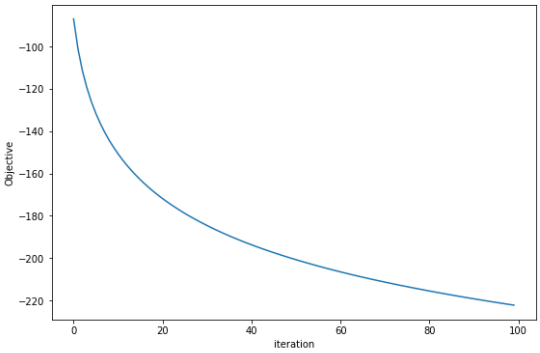

Figure 2: Objective evolution

We provide plenty of other experiments on the circle and blobs dataset that can be found on the GitHub.

3.2 Animal dataset

We now study a real world dataset containing 33 animals, each of them represented by 102 features which are answers to biological questions. We add the identity scaled by 1/3 to the covariance matrix. We choose $k = 5$ and observe the clustering results in figure 3. The five connected components seem coherent with a qualitative point of view. For example one cluster representing the insect (ant, bee,...) or another representing the birds (eagle, finch) therefore it suggests that the algorithm deduces a reasonable animals classification. Figure 4 depicts the laplacian of the graph, we remark that this matrix contains 'blocks' associated to each connected component and we can interpret the non diagonal element as the similarity measure of two animals ($L = D - A$).

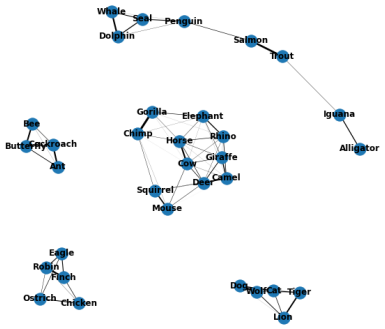

Figure 3: Animals graphs

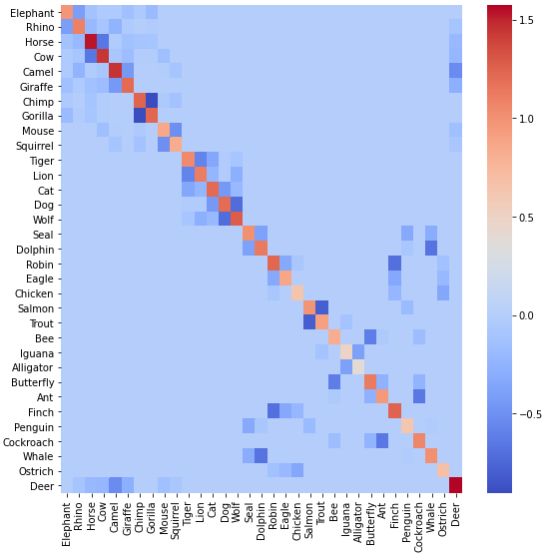

Figure 4: Laplacian of the graph

The algorithm is very useful since in this case, the features are binary (if a particular species have or not a feature). Thus, it is not possible to use the Graphical Lasso Algorithm since it relies on the Gaussian assumption.

3.3 Cancer genome dataset

We apply SGL algorithm on a more ambitious dataset : the gene expression cancer RNA-Seq Data Set[1]. The data consists in 801 samples associated to 20531 genomics features each. Every element is labelled with one of the five type of cancer considered.

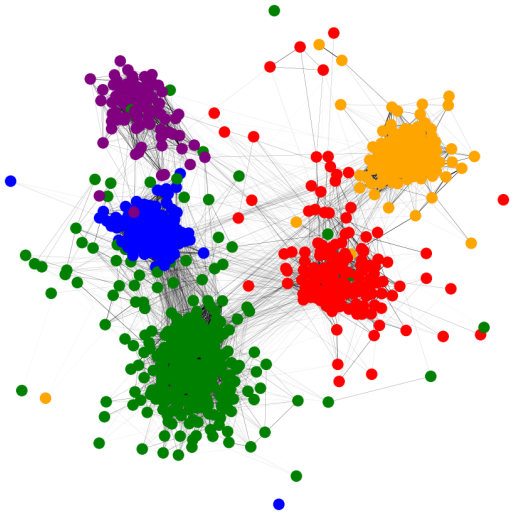

Figure 5: Cancer graph

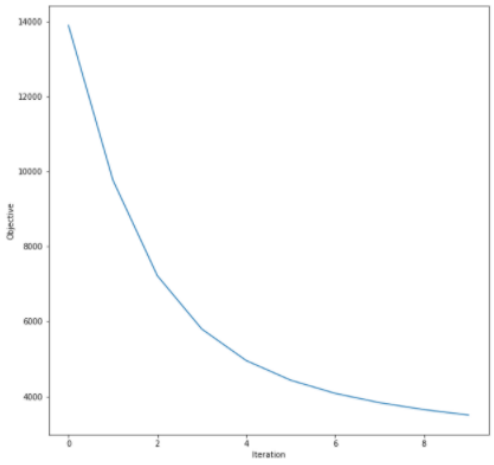

Figure 6: Objective evolution

[1]The dataset can be found UC-Irvine ML database https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq

In the figure 5 generated with $k = 5$, the color of the node represents the cancer type. Even though the graph does not have 5 distinct connected components, we observe 5 groups of dominant colors which means that the algorithm is able to construct a meaningful structure between the nodes since people with a given type of cancer are 'close'. We also remark that there are some outliers which have no neighbours or neighbours belonging to a wrong cluster but they are a minority.

3.4 Time series application

We have seen a method allowing us to simultaneously build a Graph and learn its parameters. Thus a very natural extend to the SGL algorithm is to apply it to various type of data. Indeed, it only requires a sample covariance matrix to work and a vast majority of data can be reformulated and synthesized into a covariance matrix. As we have seen in class, Graph learning is very helpful for Time-series data and a whole field is dedicated to this task : Signal Graph Processing. Thus, in this work, we tried to extend the work we have done on a previous assignment where we analysed a multivariate meteorological dataset [2].

The goal of the experiment was to design a graph which links meteorological sensors at different places in Brittany relatively to their similarity in temperature. In practice, we created the correlation matrix using all the timesteps available in the dataset (2 months) i.e. for a pair of sensors, we compute the correlation between the associated time vectors.

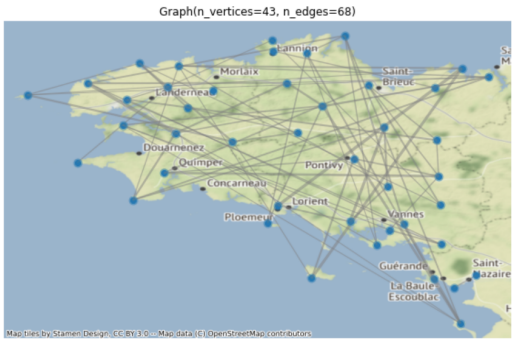

Figure 7: Sensors graph

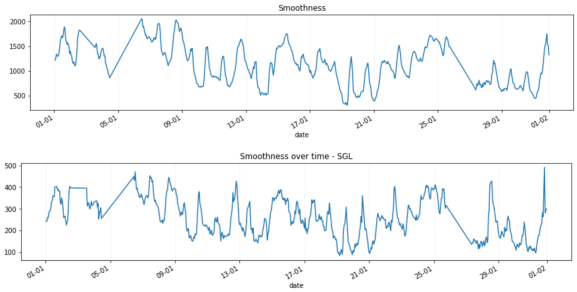

Figure 8: Smoothness evolution

We observe in figure 8 that the smoothness of the signal on the SGL graph is lower on average than the smoothness of the signal on the Correlation graph. Since both laplacian are normalized, the smoothness on both graphs are comparable, and SGL seems to yield better results since the signal fluctuates less from one node to its neighbors (definition of the smoothness) than the method with the Correlation. More figures and details can be found on the dedicated notebook on the GitHub.

4 Conclusion

In this project, we implemented an algorithm solving the graph learning task. This approach use spectral constraints on the Laplacian matrix to provide an efficient block successive minimization algorithm. We show on both synthetic and real world dataset the quality of the representation learnt particularly on the Cancer dataset where the cancer type are naturally grouped. We also try this method with time series using the correlation matrix by creating a single graph. However, this method does not use the temporal aspect of the time series data. An extent could be to try to learn a representation for each time step by adding a constraint such that laplacian matrices stay close between two time steps. Another idea, might be to create a single graph representation by merging some graphs that are created at each timestep.

[2]The dataset is provided by the French national meteorological service https://www.data.gouv.fr/fr/datasets/projections-climatiques-sur-la-zone-large-molene-sur-un-mois/

References

[FHT08] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3 (2008), pp. 432–441.

[Kum+19] Sandeep Kumar et al. *A Unified Framework for Structured Graph Learning via Spectral Constraints*. 2019. arXiv: 1904.09792 [stat.ML].