Cellular Sheaves on Graphs

Learning the Sheaf Laplacian through minimum total variation

April 16, 2024

SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

Learning the Sheaf Laplacian

1 From the sheaf Laplacian to the restriction maps

The main idea in tackling the problem of learning the sheaf laplacian from a set of smooth signals X has already been addressed by Jacob Hansen [1] by properly defining the convex cone of sheaf laplacians $\mathcal{L}_F$ and defining the following optimization problem over it:

$$\min_{L \in \mathcal{L}_F} \mathrm{tr}(X^T L X) + \alpha f_c(L) + \beta f_s(L) \tag{1}$$

In particular, it is a regularized minimum total variation problem where the regularization terms f_c and f_s encourage the connectivity and the sparsity respectively.

[1] Hansen J., Ghrist R., *Learning Sheaf Laplacians from Smooth Signals*, 2019

This approach properly defines the sheaf laplacian as the superimposition of the contributions of all the edges, each of them contributing for the coboundary map and the sparsity structure.
Taking into account that the sheaf laplacian can be rewritten as $L = B^T B$, where B is the coboundary map, we have:

$$X^T L X = X^T B^T B X = ||BX||^2 \tag{2}$$

In particular this means that minimizing the minimum total variation with respect to the sheaf laplacian is equivalent to minimize the norm of BX, that has the following block structure with respect to the restriction maps $\mathcal{F}_{u \triangleleft e} : \mathcal{F}_u \to \mathcal{F}_e$:

$$(BX)_e = \mathcal{F}_{u \triangleleft e} X_u - \mathcal{F}_{v \triangleleft e} X_v \tag{3}$$

So finally we have that

$$||BX||^2 = \sum_{e \in \mathcal{E}} ||\mathcal{F}_{u \triangleleft e} X_u - \mathcal{F}_{v \triangleleft e} X_v||^2 \tag{4}$$

Recasting the laplacian learning problem in this way allows us to control edge-wise the total variation, imposing a thighter control on the sparsity pattern.

Table of Contents

2 A proposal for an algorithm

Given the additive structure of the problem, we can design a procedure trying to generalize the topology learning algorithm in [2]: in particular

- We assume the number of edges E_0 is a priori known (or at least can be validated through a nested procedure);
- We compute maps minimizing $||\mathcal{F}_{u \triangleleft (u,v)} X_u - \mathcal{F}_{v \triangleleft (u,v)} X_v||^2$ for each possible (u, v): this means we have to solve $\binom{V}{2}$ subproblems;
- We sort our solutions with respect to the energy on the edge, and retrieve the first E_0: we then use this map to populate the coboundary map and build the sheaf laplacian.

[2] Barbarossa S., Sardellitti S., *Topological Signal Processing over Simplicial Complexes*, 2019

The core of the procedure is clearly the subproblem

$$\min_{\mathcal{F}_{u \triangleleft e}, \mathcal{F}_{v \triangleleft e}} \frac{1}{2} ||\mathcal{F}_{u \triangleleft (u,v)} X_u - \mathcal{F}_{v \triangleleft (u,v)} X_v||^2 - \lambda_u \log(\det(\mathcal{F}_{u \triangleleft e})) - \lambda_v \log(\det(\mathcal{F}_{v \triangleleft e})) \tag{5}$$

where the regularization terms avoid the trivial solution where the two maps are just matrices of all zeroes [3].

[3] Kalofolias V., *How to learn a graph from smooth signals*, 2016

This problem is non-convex but can be solved in an iterative way through successive convex approximations [4], being block-wise convex. The update equations are based on alternated proximal mapping, based on the results of [5] in deriving the proximal mapping of the logdet of a matrix. In particular, given $X \in \mathbb{R}^{n \times n}$ and its spectral decomposition $X = U^T \mathrm{Diag}(s) U$, the proximal mapping of $-\log\det(X)$ is

$$\mathrm{prox}_{-\gamma \log\det(X)} = U\,\mathrm{Diag}(z)\,U^T,\; z = \frac{1}{2}(s + \sqrt{s^2 + 4\gamma}) \tag{6}$$

[4] Scutari G., Facchinei F., Song P., Palomar D. P., and Pang J.-S., *"Decomposition by partial linearization: Parallel optimization of multi-agent systems"*, 2014

[5] Bauschke H.H., Combettes P.L., *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2017

Learning the restriction maps

Finally the blockwise convexity gives us the following update equations for $t = 0, \ldots, T$:

$$\begin{aligned}
\hat{\mathcal{F}}_{u \trianglelefteq e}(\mathcal{F}^t_{u \trianglelefteq e}, \mathcal{F}^t_{v \trianglelefteq e}) &= \operatorname{prox}_{-\lambda_u \log\det()}[\mathcal{F}^t_{v \trianglelefteq e}(X_v X_u^T)(X_u X_u^T)^{-1}] \\
\hat{\mathcal{F}}_{v \trianglelefteq e}(\mathcal{F}^t_{u \trianglelefteq e}, \mathcal{F}^t_{v \trianglelefteq e}) &= \operatorname{prox}_{-\lambda_v \log\det()}[\mathcal{F}^t_{u \trianglelefteq e}(X_u X_v^T)(X_v X_v^T)^{-1}] \\
\mathcal{F}^{t+1}_{u \trianglelefteq e} &= \mathcal{F}^t_{u \trianglelefteq e} - \eta^t[\hat{\mathcal{F}}_{u \trianglelefteq e}(\mathcal{F}^t_{u \trianglelefteq e}, \mathcal{F}^t_{v \trianglelefteq e}) - \mathcal{F}^t_{u \trianglelefteq e}] \\
\mathcal{F}^{t+1}_{v \trianglelefteq e} &= \mathcal{F}^t_{v \trianglelefteq e} - \eta^t[\hat{\mathcal{F}}_{v \trianglelefteq e}(\mathcal{F}^t_{u \trianglelefteq e}, \mathcal{F}^t_{v \trianglelefteq e}) - \mathcal{F}^t_{v \trianglelefteq e}]
\end{aligned} \tag{7}$$

However, this procedure soffers of many drawbacks, like the sensitivity to the initialization or the high number of stationary points in which it may get stuck.

We considered the same setting of [6], generating random Erdős–Rényi graphs ($p = 1.1\log(N_v)/N_V$) and over them random sheaves whose maps are entry-wise gaussians of

- $V = 50$ nodes, $d = 2$ stalks dimension;
- $V = 100$ nodes, $d = 1$ stalks dimension

Then in this case a random dataset X of $N = 100$ signals is sampled over the nodes and smoothed accordingly to the Fourier domain embedded in the sheaf laplacian in a Tikhonov fashion: given $L = U^T\mathrm{Diag}(\Lambda)U$, we have that the smoothed signals dataset is $Y = U\mathrm{Diag}(H(\Lambda))U^TX$, where $H(\Lambda) = \frac{1}{1+10\Lambda}$. In the end, gaussian noise is added with standard deviation $\sigma = 0.01$.

[6] Hansen J., Ghrist R., *Learning Sheaf Laplacians from Smooth Signals*, 2019

A simulation

Metric	V = 100, d = 1	V = 50, d = 2
Average entry-wise L_2 error	0.3568 **0.0.0496**	0.5178 **0.0653**
Average entry-wise L_1 error	0.5053 **0.0112**	0.6414 **0.0132**
F1-score [7]	0.5053 **0.9492**	0.6414 **0.9151**

Table: Comparing Hansen's results with ours (in bold)

[7] In Hansen's approach the number of edges is not a priori knowledge, so F1 score makes sense, while in our case its just coincides with precision.

A final remark

The proposed algorithm might be tackled also from another perspective. Let's consider the problem with a different notation to streamline the visualization:

$$\min_{A,B} \frac{1}{2}||AX - BY||^2 - \lambda_A \log(\det(A) - \lambda_B \log(\det(B)) \tag{8}$$

We have the following equations coming from considering again the block-wise convexity:

$$\begin{aligned} \frac{\partial}{\partial A}F(A,B) = AXX^T - BYX^T - \lambda_A(A^{-1})^T = AXX^T - BYX^T - \lambda_A(A^T)^{-1} \\ \rightarrow AXX^TA^T - BYX^TA^T - \lambda_A\mathbf{I} = 0 \\ \frac{\partial}{\partial B}F(A,B) = -AXY^T + BYY^T - \lambda_B(B^{-1})^T = BYY^T - AXY^T - \lambda_B(B^T)^{-1} \\ \rightarrow BYY^TB^T - AXY^TB^T - \lambda_B\mathbf{I} = 0 \end{aligned} \tag{9}$$

A final remark

These quadratic matrix equations

$$\begin{aligned} AXX^TA^T - BYX^TA^T - \lambda_A \mathbf{I} = 0 \\ BYY^TB^T - AXY^TB^T - \lambda_B \mathbf{I} = 0 \end{aligned} \tag{10}$$

have been proven to be solvable in [8], even if it requires a certain computational cost.

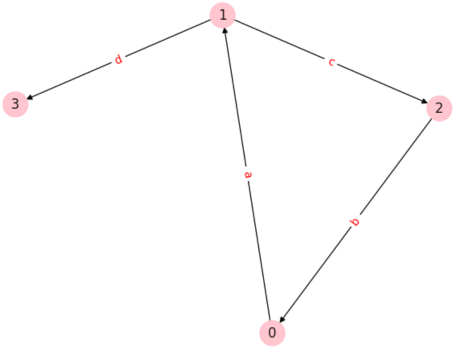
[8] Yongxin Y., Lina L., Huiting Z., Hao L., *Solutions to the quadratic matrix equation XAX + BX + D = 0*, 2021

Table of Contents

A motivating example

Let's consider the following cellular sheaf:

$$B = \begin{bmatrix} -F_{a1} & F_{b1} & 0 & 0 \\ F_{a2} & 0 & -F_{c2} & -F_{d2} \\ 0 & -F_{b3} & F_{c3} & 0 \\ 0 & 0 & 0 & F_{d4} \end{bmatrix}$$



We have that:

$$B^T x = \begin{bmatrix} -F_{a1}^T x_1 + F_{a2}^T x_2 \\ F_{b1}^T x_1 + F_{b3}^T x_3 \\ -F_{c2}^T x_2 + F_{c3}^T x_3 \\ -F_{d2}^T x_2 + F_{d4}^T x_4 \end{bmatrix}$$

A motivating example

3 Another approach

Given the previous it is clear that through a proper choice of matrices, under certain smoothness assumptions on the signals over the nodes, we can force the sheaf to have zero total variation.
Indee, after sampling x_1, we build the remaining dataset as:

$$\begin{aligned} x_2 &= F_{a1}^T x_1 \\ x_3 &= F_{b3}^T x_1 \\ x_4 &= F_{d2}^T F_{a1}^T x_1 \end{aligned} \tag{11}$$

Setting $F_{a2} = F_{b1} = F_{c3} = F_{d4} = \mathbb{I}$, $F_{c2} = F_{a1}^{-1} F_{b3}^{-1}$, for any choice of the other restriction maps we have total variation equal to zero.

A different approach to get a close form solution

If we focus to the contribution of a single edge, under the hypotesis where all the restriction maps are square invertible matrices, we can rewrite it as follow:

$$F_{u \triangleleft e} X_u - F_{v \triangleleft e} X_v = \frac{1}{2} [F_{u \triangleleft e} (X_u - F_{u \triangleleft e}^{-1} F_{v \triangleleft e} X_v) + F_{v \triangleleft e} (F_{v \triangleleft e}^{-1} F_{u \triangleleft e} X_u - X_v)] \tag{12}$$

This means that if we try to set to zero one component (or at least to reduce at minimum its mismatch with the model) we can focus just on the quantities within brackets, where we can set to zero and get closed form solutions.
Then we can adapt the previous of idea of sorting and retrieving the E_0 edges.