SCCGNN: any ideas?

Anton Savostianov[1] • **Francesco Tudisco**[1]• **Nicola Guglielmi**[1]
[1]Gran Sasso Science Institute, viale F.Crispi 7, L'Aquila, Italy, 67100, email: `anton.savostianov@gssi.it`

Abstract: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords: NNs

I. Things

Let $\mathcal{K}$ be a simplicial complex where $\mathcal{V}_k(\mathcal{K})$ is the set of k-order simplices in $\mathcal{K}$, $\mathcal{K} = \mathcal{V}_0(\mathcal{K}) \cup \mathcal{V}_1(\mathcal{K}) \cup \mathcal{V}_2(\mathcal{K}) \cup \ldots$; let $m_k = |\mathcal{V}_k(\mathcal{K})|$. Let $B_k \in \mathrm{Mat}_{m_{k-1}\times m_k}$ be a boundary operator mapping simplex of order k to its border of the order $k-1$; by the fundamental lemma of topology $B_k B_{k+1} = 0$.

Given the lemma above, the homology group $\mathcal{H}_k = {}^{\operatorname{im} B_k}\!/_{\ker B_{k+1}}$ is correctly defined; the elements of $\mathcal{H}_k$ correspond to k-dimensional holes in the simplicial complex $\mathcal{K}$. Through the harmonic representative it is convenient to exploit the isomorphism:

$$\mathcal{H}_k \cong \ker\left(B_k^\top B_k + B_{k+1}B_{k+1}^\top\right)$$

Operators $L_k^\downarrow = B_k^\top B_k$, $L_k^\uparrow = B_{k+1}B_{k+1}^\top$ and $L_k = L_k^\downarrow + L_k^\uparrow$ are referred as $k-$th order up-, down- and complete graph Laplacians.

Algebra of boundary operators admits an important full space decomposition:

Lemma 1 **(Hodge Decomposition, [Lim19, Thm. 5.2])** For the k-th order Hodge Laplacian $L_k \in \mathbb{R}^{m_k\times m_k}$, the following decomposition holds:

$$\mathbb{R}^{m_k} = \overbrace{\operatorname{im} B_k^\top \oplus \ker L_k}^{\ker B_{k+1}^\top} \oplus \operatorname{im} B_{k+1}$$

$$\mathbb{R}^{m_k} = \operatorname{im} B_k^\top \oplus \underbrace{\ker L_k \oplus \operatorname{im} B_{k+1}}_{\ker B_k}$$

We also assume the case of weighted simplicial complex; in such situations the family of k weight functions are introduced: $w_k(\cdot) : \mathcal{V}_k(\mathcal{K}) \mapsto (0; +\infty)$ with corresponding diagonal weight matrices $W_k \in \mathrm{Mat}_{m_k\times m_k}$ where $[W_k]_{ii} = w_k(\sigma_i)$, $\sigma_i \in \mathcal{V}_k(\mathcal{K})$. Then the weighting scheme preserving the homology definition can be designed as follows:

$$B_k \to W_{k-1}^{-1} B_k W_k$$

Weighting does not change the dimensionality of the homology group and the remaining terms in the decomposition; isomorhism and the decomposition above hold in the weighted case as well.

II. SCCGNN

Let C_k be a formal linear space on $\mathcal{V}_k(\mathcal{K})$.

We consider a semi-supervised setting: we have the flow $\boldsymbol{x} \in C_1$ with a share of missing values; the goal is to reconstruct the missing values (let ν be the share of the missing values; missing entries in $\boldsymbol{x}$ are filled with the median value of the present values).

Convolutional layer induced by the simplicial geometry is given by:

$$\boldsymbol{x}_{n+1} = \sigma\left(\sum_{i=0}^{K} \alpha_i^{(n)} L_1^i \boldsymbol{x}_n\right), \qquad \text{where } \sigma = \text{reLU and } \alpha_i^{(n)} \in \mathbb{R}$$

where $\sigma(x) = x \cdot \chi(x)$. Note that $L_1^i = \left(B_1^\top B_1\right)^i + \left(B_2 B_2^\top\right)^i = L_1^{\downarrow i} + L_1^{\uparrow i}$; let $P_K(\boldsymbol{\alpha}, A) = \sum_{i=0}^{K} \alpha_i A^i$ be a matrix polynomial. Then, the convolution layer can be rewritten as

$$\boldsymbol{x}_{n+1} = \sigma\left(\sum_{i=0}^{K} \alpha_i^{(n)} L_1^i \boldsymbol{x}_n\right) = \sigma\left(P_K(\boldsymbol{\alpha}^{(n)}, L_1)\boldsymbol{x}_n\right) = \sigma\left(P_K(\boldsymbol{\alpha}^{(n)}, L_1^{\downarrow})\boldsymbol{x}_n + P_K(\boldsymbol{\alpha}^{(n)}, L_1^{\uparrow})\boldsymbol{x}_n\right) \tag{Eqn. 1}$$

Remark Given the classical definition, the layer produces the output vector $\boldsymbol{x}_{n+1}$ as an element of the Krylov subspaces spanned by the input $\boldsymbol{x}_n$ baring the activation function:

why do we need it? who knows

$$\boldsymbol{x}_n = \sigma\left(\boldsymbol{y}_n\right), \qquad \boldsymbol{y}_n \in \mathcal{K}_K(L_1, \boldsymbol{x}_n)$$

In a more general case, one could consider unmatching layer with two independent polynomials:

$$\boldsymbol{x}_{n+1} = \sigma\left(P_K(\boldsymbol{\alpha}^{(n)}, L_1^{\downarrow})\boldsymbol{x}_n + Q_K(\boldsymbol{\beta}^{(n)}, L_1^{\uparrow})\boldsymbol{x}_n\right) \tag{Eqn. 2}$$

Let us omit $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the polynomial arguments for clarity (we will refer to the coefficients of P_K as α_i and Q_k as β_i respectively).

III. Flow leakage

For the Hodge decomposition, Lemma 1, we define projectors on $\operatorname{im} B_1^\top$ and $\operatorname{im} B_2$ as $\Pi_1 = B_1^\top \left(B_1 B_1^\top\right)^+ B_1$ and $\Pi_2 = B_2 \left(B_2^\top B_2\right)^+ B_2^\top$ respectively.

note that one can use stochastic Cholesky and `HeCS` as fast projectors here

Then

$$\begin{aligned} \boldsymbol{x}_{n+1} &= \sigma\left(P_K(\boldsymbol{\alpha}^{(n)}, L_1^{\downarrow})\boldsymbol{x}_n + Q_K(\boldsymbol{\beta}^{(n)}, L_1^{\uparrow})\boldsymbol{x}_n\right) = \\ &= \sigma\left(\gamma^{(n)}\boldsymbol{x}_n + P_{K-1}(L_1^{\downarrow})L_1^{\downarrow}\boldsymbol{x}_n + Q_{K-1}(L_1^{\uparrow})L_1^{\uparrow}\boldsymbol{x}_n\right) \end{aligned} \tag{Eqn. 3}$$

Let $\boldsymbol{x}_n = \boldsymbol{y}_n + \boldsymbol{h}_n + \boldsymbol{z}_n$ such that $\boldsymbol{y}_n \in \operatorname{im} B_1^\top$, $\boldsymbol{z}_n \in \operatorname{im} B_2$ and $\boldsymbol{h}_n \in \ker L_1$. Then:

$$\boldsymbol{x}_n = \sigma\left(\gamma^{(n)}\boldsymbol{h}_n + \left[\gamma^{(n)} I + P_{K-1}(L_1^{\downarrow})L_1^{\downarrow}\right]\boldsymbol{y}_n + \left[\gamma^{(n)} I + Q_{K-1}(L_1^{\uparrow})L_1^{\uparrow}\right]\boldsymbol{z}_n\right) \tag{Eqn. 4}$$

Remark (reLU action of the subspaces) Let $\boldsymbol{y} \in \operatorname{im} B_1^\top$; then $\boldsymbol{y} = B_1^\top \boldsymbol{x}$. What can we say about $\sigma(B_1^\top \boldsymbol{x})$? By the definition of the reLU:

$$B_1^\top \boldsymbol{x} = \sigma(B_1^\top \boldsymbol{x}) - \sigma(-B_1^\top \boldsymbol{x}) \tag{Eqn. 5}$$

Projecting both sides of this equation by mutiplying by Π_2 we get

$$\begin{aligned} \Pi_2 \sigma(B_1^\top \boldsymbol{x}) &= \Pi_2 \sigma(-B_1^\top \boldsymbol{x}) \\ \Pi_H \sigma(B_1^\top \boldsymbol{x}) &= \Pi_H \sigma(-B_1^\top \boldsymbol{x}) \end{aligned} \tag{Eqn. 6}$$

III. References

[EDS20] Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial Neural Networks, December 2020.

[Lim19] Lek-Heng Lim. Hodge Laplacians on graphs, August 2019.

[YI23] Maosheng Yang and Elvin Isufi. Convolutional Learning on Simplicial Complexes, January 2023.