

Seminario de Aprendizaje Topológico 2025-2

Simplicial Neural Networks

Stefania Ebli & Michaël Defferrard

Diego Leipen Lara

Cocadenas

Sea K un complejo simplicial abstracto y sea K_p su p -esqueleto. Definimos $C^p(K)$ como el espacio vectorial (sobre \mathbb{R}) cuyos elementos son funciones de K_p a \mathbb{R} . Decimos que los elementos de $C^p(K)$ son p -cocadenas. Para cada $\sigma \in K_p$ sea $e_\sigma \in C^p(K)$ tal que

$$e_\sigma(\tau) = \begin{cases} 1 & \text{si } \tau = \sigma, \\ 0 & \text{e.o.c.} \end{cases}$$

El conjunto $\{e_\sigma \mid \sigma \in K_p\}$ forma una base de $C^p(K)$ tal que

$$c = \sum_{\sigma \in K_p} c(\sigma) e_\sigma.$$

En lo que sigue, consideramos a $C^p(K)$ con el producto punto $\langle \cdot, \cdot \rangle_p$ que hace que el conjunto $\{e_\sigma \mid \sigma \in K_p\}$ sea ortonormal.

El operador cofrontera

Definimos el *operador cofrontera* $\delta_p : C^p(K) \rightarrow C^{p+1}(K)$ por

$$\delta_p(c)([v_0, \dots, v_{p+1}]) = \sum_{i=0}^{p+1} (-1)^i c([v_0, \dots, \hat{v}_i, \dots, v_{p+1}]).$$

La matriz B_p asociada a δ_p es de dimensión $|K_{p+1}| \times |K_p|$ y

$$(B_p)_{\tau, \sigma} = \begin{cases} \pm 1 & \text{si } \sigma \subset \tau \\ 0 & \text{e.o.c.} \end{cases}$$

En efecto,

$$\delta_p(e_\sigma) = \sum_{\tau \in K_{p+1}} \delta_p(e_\sigma)(\tau) e_\tau \quad \text{y} \quad \delta_p(e_\sigma)(\tau) = \begin{cases} \pm 1 & \text{si } \sigma \subset \tau \\ 0 & \text{e.o.c.} \end{cases}$$

El operador cofrontera (continuación)

En particular, (i) se puede demostrar que B_0 es la matriz de incidencia orientada usual y (ii) en la mayoría de los casos prácticos, B_p se puede calcular explícitamente como una matriz dispersa.

Se puede demostrar que $\delta_p \circ \delta_{p-1} = 0$, es decir, $\text{im } \delta_{p-1} \subset \ker \delta_p$. Definimos el *grupo de cohomología reducida de grado p* ,

$$\tilde{H}^p(K) := \ker \delta_p / \text{im } \delta_{p-1}.$$

El operador Laplaciano

El adjunto del operador cofrontera $\delta_p : C^p(K) \rightarrow C^{p+1}(K)$ es el operador $\delta_p^* : C^{p+1}(K) \rightarrow C^p(K)$ que satisface

$$\langle \delta_p(f), g \rangle_{p+1} = \langle f, \delta_p^*(g) \rangle_p$$

para todo $f \in C^p(K)$ y $g \in C^{p+1}(K)$. Usando esto, definimos el *operador Laplaciano* $\mathcal{L}_p : C^p(K) \rightarrow C^p(K)$ por

$$\mathcal{L}_p = \underbrace{\delta_p^* \circ \delta_p}_{\mathcal{L}_p^{\text{up}}} + \underbrace{\delta_{p-1} \circ \delta_{p-1}^*}_{\mathcal{L}_p^{\text{down}}}.$$

Si L_p denota a la matriz asociada a \mathcal{L}_p , entonces (i) L_0 es la matriz Laplaciana usual y (ii) podemos calcular $L_p = B_p^T B_p + B_{p-1} B_{p-1}^T$ (en particular, L_p es diagonalizable).

Teorema de Eckmann

Teorema 1 (Eckmann, 1944)

Para cualquier complejo simplicial abstracto K , $\ker \mathcal{L}_p \cong \tilde{H}^p(K)$.

Demostración. Notemos que

$$\begin{aligned}\mathcal{L}_p^{\text{up}} \circ \mathcal{L}_p^{\text{down}} &= \delta_p^* \circ \underbrace{\delta_p \circ \delta_{p-1}}_0 \circ \delta_{p-1}^* = 0, \\ \mathcal{L}_p^{\text{down}} \circ \mathcal{L}_p^{\text{up}} &= \delta_{p-1} \circ \underbrace{\delta_{p-1}^* \circ \delta_p^*}_0 \circ \delta_p = 0.\end{aligned}$$

Entonces $\text{im } \mathcal{L}_p^{\text{down}} \subset \ker \mathcal{L}_p^{\text{up}}$ y $\text{im } \mathcal{L}_p^{\text{up}} \subset \ker \mathcal{L}_p^{\text{down}}$. Por lo tanto,

$$\begin{aligned}\ker \mathcal{L}_p &= \ker \delta_p^* \circ \delta_p \cap \ker \delta_{p-1} \circ \delta_{p-1}^* \\ &= \ker \delta_p \cap \ker \delta_{p-1}^* = \ker \delta_p \cap (\text{im } \delta_{p-1})^\perp \cong \tilde{H}^p(K).\end{aligned}$$

La transformada de Fourier

Sean $\lambda_1 \leq \dots \leq \lambda_{|K_p|}$ los eigenvalores de \mathcal{L}_p y sean $c_1, \dots, c_{|K_p|}$ sus respectivos eigenvectores. Definimos la *transformada de Fourier* $\mathcal{F}_p : C^p(K) \rightarrow \mathbb{R}^{|K_p|}$ por

$$\mathcal{F}_p(c) = \left(\langle c, c_1 \rangle_p, \dots, \langle c, c_{|K_p|} \rangle_p \right).$$

Como L_p es simétrica y real, sus eigenvectores forman una base ortonormal. En particular, la matriz asociada a \mathcal{F}_p es ortonormal. Más aun, si denotamos $\Lambda = (\lambda_1, \dots, \lambda_{|K_p|})$ y U es la matriz asociada a \mathcal{F}_p^{-1} , entonces

$$\begin{aligned} U \operatorname{diag}(\Lambda) U^T c_i &= U \operatorname{diag}(\Lambda) (0, \dots, 1, \dots, 0) \\ &= U(0, \dots, \lambda_i, \dots, 0) = \lambda_i c_i = L_p c_i. \end{aligned}$$

Por lo tanto, $L_p = U \operatorname{diag}(\Lambda) U^T$.

Capas convolucionales

Motivados por la propiedad $\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g)$ de la transformada de Fourier clásica, definimos

$$c *_p c' = \mathcal{F}_p^{-1}(\mathcal{F}_p(c) \odot \mathcal{F}_p(c')).$$

Esto nos lleva a definir una *capa convolucional* con input $c \in C^p(K)$ y pesos W como una función de la forma

$$\psi \circ (\mathcal{F}_p^{-1}(\varphi_W) *_p c)$$

donde (i) ψ es una función de activación y sesgo y (ii) φ_W es un polinomio de grado pequeño en $(\lambda_1, \dots, \lambda_{|K_p|})$, es decir,

$$\varphi_W = \sum_{i=0}^N W_i(\lambda_1^i, \dots, \lambda_{|K_p|}^i).$$

Capas convolucionales (continuación)

La razón para restringirnos a los φ_W es que de esta manera

$$\mathcal{F}_p^{-1}(\varphi_W) *_p c = \sum_{i=0}^N W_i L_p^i c$$

donde $c^T = (c(\sigma))_{\sigma \in K_p}$ es la matriz asociada a c . En efecto,

$$\begin{aligned} \mathcal{F}_p^{-1}(\varphi_W) *_p c &= \mathcal{F}_p^{-1}(\varphi_W \odot \mathcal{F}_p(c)) = U(\text{diag}(\varphi_W) U^T c) = \\ U \left(\sum_{i=0}^N W_i \text{diag}(\Lambda)^i U^T c \right) &= \sum_{i=0}^N W_i U \text{diag}(\Lambda)^i U^T c = \sum_{i=0}^N W_i L_p^i c. \end{aligned}$$

Esto es importante por 3 razones. Primero, la convolución puede ser implementada por N multiplicaciones de matrices dispersas. Esto reduce la complejidad de $\mathcal{O}(|K_p|^2)$ a $\mathcal{O}(\xi |K_p|)$ donde ξ es el factor de densidad. Segundo, el número de pesos por aprender se reduce de $\mathcal{O}(|K_p|)$ a $\mathcal{O}(1)$.

Capas convolucionales (continuación)

Tercero, la operación es N -localizante en el sentido de que si la distancia entre dos simplejos σ, τ es mayor a N , entonces la capa convolucional no causa interacción entre $c(\sigma)$ y $c(\tau)$ en su output. Estas interacciones pueden ser interpretadas como pase de mensajes entre simplejos y esta falta de interacción es análoga a que una CNN usual de tamaño d no distribuye información entre pixeles que distan a mas de d pixeles.

Precisemos esto. Supongamos que σ y τ son p -simplejos. Un camino de longitud k de σ a τ es una sucesión $(\nu_0, \nu_1, \dots, \nu_k)$ de p -simplejos tal que (i) $\nu_0 = \sigma$, (ii) $\nu_k = \tau$, y (iii) ν_i comparte una cara o cocara con ν_{i-1} y ν_i comparte una cara o cocara con ν_{i+1} . La distancia simplicial entre σ y τ es el mínimo de las longitudes de los caminos de σ a τ .

Capas convolucionales (continuación)

Se puede demostrar que si d es la distancia simplicial entre σ y τ , entonces para todo $k < d$, la entrada de L_p^k correspondiente a σ y τ es 0. Específicamente, si e_σ denota la matriz asociada a la cocadena e_σ , entonces $(L_p^k)_{\sigma\tau} := L_p^k e_\sigma e_\tau = 0$.

Por otro lado, como la matriz asociada a $c \in C^p(K)$ es el vector columna $c = (c(\tau'))_{\tau' \in K_p}$, entonces

$$L_p^k c = \left(\sum_{\tau' \in K_p} (L_p^k)_{\sigma'\tau'} c(\tau') \right)_{\sigma' \in K_p}.$$

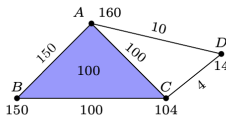
Por lo tanto, el coeficiente correspondiente a τ en $(L_p^k c)(\sigma)$ es 0.

La implementación

El *Semantic Scholar Open Research Corpus* es un dataset de artículos publicados con sus respectivos autores y numero de citas. Un *complejo de coautores* es un complejo simplicial donde un artículo con k autores esta representado por un $(k - 1)$ -simplejo. Nos interesan las cocadenas dadas por $c(\{v_0, \dots, v_p\}) =$ el número total de citas de artículos que contienen a v_0, \dots, v_p como autores.

Papers	Authors	Citations
Paper I	A, B, C	100
Paper II	A, B	50
Paper III	A, D	10
Paper IV	C, D	4

(a)



(b)

$$\begin{pmatrix} AB & AC & AD & BC & CD \\ 3 & 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 & -1 \\ 1 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 3 & -1 \\ 0 & -1 & 1 & -1 & 2 \end{pmatrix} \begin{matrix} AB \\ AC \\ AD \\ BC \\ CD \end{matrix}$$

(c)

La implementación (continuación)

Method. We evaluated the performance of the SNNs on the task of imputing missing citations on the k -cochains (for $k = 0, 1, 2$) of the extracted coauthorship complexes. As in a typical pipeline for this task [28], missing values are artificially introduced by replacing a portion of the values with a constant. Specifically, given a fixed coauthorship complex, missing data is introduced at random on the k -cochains at 5 rates: 10%, 20%, 30%, 40%, and 50%. The SNN is given as input the k -cochains on which missing citations are substituted by the median of known citations (as a reasonable first guess) and is trained to minimize the L_1 norm over known citations. We trained SNNs made of 3 layers with 30 convolutional filters of degree $N = 5$ with Leaky ReLu for 1000 iterations with the Adam optimizer and a learning rate of 10^{-3} .

Sampling papers. From the Semantic Scholar Open Research Corpus [30], we excluded papers with fewer than 5 citations or more than 10 authors. To sample a CC, we sampled 80 papers (corresponding to maximal simplices in the CC) by performing a random walk (of length 80, from a randomly chosen starting paper) on the graph whose vertices represent papers and edges connect papers sharing at least one author.

Mean accuracy and absolute error. A missing value is considered to be correctly imputed if the imputed value differs by at most 10% from the true value. The *accuracy* is the percentage of missing values that has been correctly imputed and the *absolute error* is the magnitude of the difference between the imputed and true value. For each rate of missing values, we compute the *mean accuracy* \pm standard deviation over 5 samples with randomly damaged portions.

La implementación (continuación)

Results. Figure 2 shows the mean accuracy and absolute error distribution (see the supplementary material for definitions) of the SNN in inputting missing citations on CC1. Observe that the distribution of the prediction error accumulates close to zero.

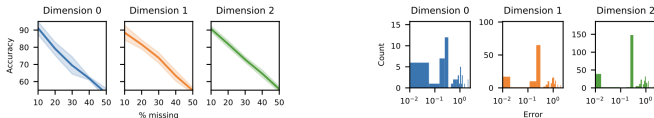


Figure 2: Performance of SNNs. Left: Mean accuracy \pm standard deviation over 5 samples in imputing missing citations on CC1. Right: Absolute error distribution over 1 sample for 40% missing citations on CC1.

Table 1 shows the performance of two baselines: missing values inferred as (i) the mean or median of all known values, and (ii) the mean of the $(k - 1)$ and $(k + 1)$ neighboring simplices. SNNs well outperform these baselines. Comparison with stronger imputation algorithms is left for future work.

To demonstrate that our filters transfer across complexes, we evaluated how accurately an SNN trained on one coauthorship complex can impute missing citations on a different complex. We found that when imputing citations on CC1, a SNN trained on CC2 is almost as good as one trained on CC1 (compare Figures 2 and 3). We expect this result as coauthorship complexes share a similar structure, and the same process underlies the generation of citations across coauthorship complexes.

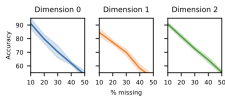


Figure 3: Performance on CC1 with an SNN trained on CC2.

Method	Dimension 0	Dimension 1	Dimension 2
Global Mean	3.30 ± 0.82	5.75 ± 1.28	2.96 ± 0.49
Global Median	7.78 ± 2.70	10.44 ± 1.00	12.50 ± 0.63
Neighbors Mean	11.88 ± 5.29	24.15 ± 1.85	27.38 ± 1.18

Table 1: Performance of baselines: mean accuracy \pm standard deviation over 5 samples for 30% missing citations on CC1.

Un par de detalles

1) En realidad, vamos a trabajar con cocadenas cuyo contradominio es cualquier espacio euclidiano \mathbb{R}^d . La única diferencia generada sera en el calculo de la convolución

$$\mathcal{F}_p^{-1}(\varphi_W) *_p c = \sum_{i=0}^N W_i L_p^i c.$$

En cambio, ahora tendremos

$$\mathcal{F}_p^{-1}(\varphi_W) *_p c = \sum_{j=0}^d \sum_{i=0}^N W_i L_p^i c_j$$

donde c_j es la matriz asociada a $c_j = \pi_j \circ c$.

Un par de detalles (continuación)

2) En realidad, no vamos a trabajar con los

$$\varphi_W = \sum_{i=0}^N W_i(\lambda_1^i, \dots, \lambda_{|K_p|}^i).$$

En realidad, vamos a trabajar con polinomios de la forma

$$\varphi_\theta = \sum_{k=0}^K \theta_k T_k(\tilde{\Lambda})$$

donde (i) los θ_k son parámetros aprendibles, (ii) T_k es el polinomio de Chebyshev de grado k , y (iii) $\tilde{\Lambda} = 2\Lambda/\lambda_{\max} - (1, \dots, 1)$ es Λ escalado a $[-1, 1]$. Cabe recalcar que $\tilde{\Lambda}$ es el vector de eigenvalores de el Laplaciano escalado $\tilde{L} = 2L/\lambda_{\max} - I$.

Un par de detalles (continuación)

Los *polinomios de Chebyshev* son una sucesión de polinomios $T_k : [-1, 1] \rightarrow \mathbb{R}$ que están definidos recursivamente por

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x).$$

La razón de escalar el Laplaciano es justo que el dominio de T_k es $[-1, 1]$. De esta manera, se puede demostrar que

$$\mathcal{F}_p^{-1}(\varphi_\theta) *_p c = \sum_{j=0}^d \sum_{k=0}^K \theta_k T_k(\tilde{L}) c_j.$$

En el artículo *Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering* de Michaël Defferrard et al., se menciona que la razón para considerar los φ_θ sobre los φ_W es una cuestión de costo computacional.