FACULTAD DE INGENIERÍA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Aprendizaje Automático para Datos en Grafos

Graph Neural Networks: Propiedades

Federico 'Larroca' La Rocca

Muy basado en transparencias de **Fernando Gama** y **Luana Ruiz**

flarroca@fing.edu.uy

http://iie.fing.edu.uy/personal/flarroca

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.
  $\Rightarrow$ Información local, implementación eficiente (distribuida)

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.
  - $\Rightarrow$ Información local, implementación eficiente (distribuida)
- Cuando la intuición falla tenemos que recurrir a propiedades matemáticas

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.
  $\Rightarrow$ Información local, implementación eficiente (distribuida)
- Cuando la intuición falla tenemos que recurrir a propiedades matemáticas $\Rightarrow$ ¿Qué sabemos sobre CNNs?

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.
  $\Rightarrow$ Información local, implementación eficiente (distribuida)
- Cuando la intuición falla tenemos que recurrir a propiedades matemáticas $\Rightarrow$ ¿Qué sabemos sobre CNNs?
- CNNs también aprovechan la información local y tienen una implementación eficiente

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.
  $\Rightarrow$ Información local, implementación eficiente (distribuida)
- Cuando la intuición falla tenemos que recurrir a propiedades matemáticas $\Rightarrow$ ¿Qué sabemos sobre CNNs?
- CNNs también aprovechan la información local y tienen una implementación eficiente
  $\Rightarrow$ Equivariancia (equivariance) a traslaciones y estabilidad [Mallat '12] $\Rightarrow$ explican su gran desempeño

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.
  $\Rightarrow$ Información local, implementación eficiente (distribuida)
- Cuando la intuición falla tenemos que recurrir a propiedades matemáticas $\Rightarrow$ ¿Qué sabemos sobre CNNs?
- CNNs también aprovechan la información local y tienen una implementación eficiente
  $\Rightarrow$ Equivariancia (equivariance) a traslaciones y estabilidad [Mallat '12] $\Rightarrow$ explican su gran desempeño
  Equivarianza $\neq$ Invarianza

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Graph Neural Networks: ¿Porqué?

- Tenemos intuición sobre convoluciones temporales. De las graph convolutions no tanto.

  $\Rightarrow$ Información local, implementación eficiente (distribuida)

- Cuando la intuición falla tenemos que recurrir a propiedades matemáticas $\Rightarrow$ ¿Qué sabemos sobre CNNs?
- CNNs también aprovechan la información local y tienen una implementación eficiente

  $\Rightarrow$ Equivariancia (equivariance) a traslaciones y estabilidad [Mallat '12] $\Rightarrow$ explican su gran desempeño

  Equivarianza $\neq$ Invarianza

- Permutation equivariance $\Rightarrow$ Aprovechar las simetrías internas del grafo

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una permutación $\mathbf{P}$ es una matriz binaria que cumple

$$\{\mathbf{P} \in \{0,1\}^{N \times N} : \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^{\mathsf{T}}\mathbf{1} = \mathbf{1}\}$$

Permutation Equivariance y Reordenamiento de Nodos

- Una permutación $\mathbf{P}$ es una matriz binaria que cumple

$$\{\mathbf{P} \in \{0,1\}^{N \times N} : \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^\mathsf{T}\mathbf{1} = \mathbf{1}\}$$

O sea, un único 1 por fila y columna. Permite definir reordenamientos de índices $\sigma(i) = j$. Ejemplo:

$$\mathbf{P}^\mathsf{T} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Permutation Equivariance y Reordenamiento de Nodos

- Una permutación $\mathbf{P}$ es una matriz binaria que cumple

$$\{\mathbf{P} \in \{0,1\}^{N\times N} : \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^\top \mathbf{1} = \mathbf{1}\}$$

O sea, un único 1 por fila y columna. Permite definir reordenamientos de índices $\sigma(i) = j$. Ejemplo:

$$\mathbf{P}^\top = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

  - El producto $\mathbf{P}^\top \mathbf{x}$ reordena las entradas del vector (tomar $\mathbf{P}^\top$ como un GSO)

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_2 \\ x_3 \\ x_1 \end{pmatrix} = \begin{pmatrix} x_{\sigma(1)} \\ x_{\sigma(2)} \\ x_{\sigma(3)} \end{pmatrix}$$

Permutation Equivariance y Reordenamiento de Nodos

- Una permutación $\mathbf{P}$ es una matriz binaria que cumple

$$\{\mathbf{P} \in \{0,1\}^{N\times N} : \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^\mathsf{T}\mathbf{1} = \mathbf{1}\}$$

O sea, un único 1 por fila y columna. Permite definir reordenamientos de índices $\sigma(i) = j$. Ejemplo:

$$\mathbf{P}^\mathsf{T} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

- El producto $\mathbf{P}^\mathsf{T}\mathbf{x}$ reordena las entradas del vector (tomar $\mathbf{P}^\mathsf{T}$ como un GSO)
- El producto $\mathbf{P}^\mathsf{T}\mathbf{S}$ reordena las filas de la matriz (pensar en $\mathbf{S}$ como N columnas concatenadas)

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T} \\ \mathbf{x}_1^\mathsf{T} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{\sigma(1)}^\mathsf{T} \\ \mathbf{x}_{\sigma(2)}^\mathsf{T} \\ \mathbf{x}_{\sigma(3)}^\mathsf{T} \end{pmatrix}$$

Permutation Equivariance y Reordenamiento de Nodos

- Una permutación $\mathbf{P}$ es una matriz binaria que cumple

$$\{\mathbf{P} \in \{0,1\}^{N\times N} : \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^\top\mathbf{1} = \mathbf{1}\}$$

O sea, un único 1 por fila y columna. Permite definir reordenamientos de índices $\sigma(i) = j$. Ejemplo:

$$\mathbf{P}^\top = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

  - El producto $\mathbf{P}^\top\mathbf{x}$ reordena las entradas del vector (tomar $\mathbf{P}^\top$ como un GSO)
  - El producto $\mathbf{P}^\top\mathbf{S}$ reordena las filas de la matriz (pensar en $\mathbf{S}$ como N columnas concatenadas)
  - El producto $\mathbf{SP}$ reordena las columnas de la matriz con el mismo orden que $\mathbf{P}^\top$

$$\begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{\sigma(1)} & \mathbf{x}_{\sigma(2)} & \mathbf{x}_{\sigma(3)} \end{pmatrix}$$

Permutation Equivariance y Reordenamiento de Nodos

- Una permutación $\mathbf{P}$ es una matriz binaria que cumple

$$\{\mathbf{P} \in \{0,1\}^{N \times N} : \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^\mathsf{T}\mathbf{1} = \mathbf{1}\}$$

O sea, un único 1 por fila y columna. Permite definir reordenamientos de índices $\sigma(i) = j$. Ejemplo:

$$\mathbf{P}^\mathsf{T} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

- El producto $\mathbf{P}^\mathsf{T}\mathbf{x}$ reordena las entradas del vector (tomar $\mathbf{P}^\mathsf{T}$ como un GSO)
- El producto $\mathbf{P}^\mathsf{T}\mathbf{S}$ reordena las filas de la matriz (pensar en $\mathbf{S}$ como N columnas concatenadas)
- El producto $\mathbf{S}\mathbf{P}$ reordena las columnas de la matriz con el mismo orden que $\mathbf{P}^\mathsf{T}$

$\Rightarrow$ El producto $\mathbf{P}^\mathsf{T}\mathbf{S}\mathbf{P}$ reordena las entradas de la matriz

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} x_{22} & x_{23} & x_{21} \\ x_{32} & x_{33} & x_{31} \\ x_{12} & x_{13} & x_{11} \end{pmatrix} = \begin{pmatrix} x_{\sigma(1)\sigma(1)} & x_{\sigma(1)\sigma(2)} & x_{\sigma(1)\sigma(3)} \\ x_{\sigma(2)\sigma(1)} & x_{\sigma(2)\sigma(2)} & x_{\sigma(2)\sigma(3)} \\ x_{\sigma(3)\sigma(1)} & x_{\sigma(3)\sigma(2)} & x_{\sigma(3)\sigma(3)} \end{pmatrix}$$

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal

$$\mathbf{P}^\mathsf{T}\mathbf{I}\mathbf{P} = \quad \mathbf{P}^\mathsf{T}\mathbf{P}$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^{\mathsf{T}}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^{\mathsf{T}}\mathbf{P}$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^{\mathsf{T}}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^{\mathsf{T}}\mathbf{P}$$

$\Rightarrow$ Podemos reordenar primero y operar después $\qquad (\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})(\mathbf{P}^{\mathsf{T}}\mathbf{x})$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^\mathsf{T}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^\mathsf{T}\mathbf{P}$$

$\Rightarrow$ Podemos reordenar primero y operar después u operar y luego reordenar $(\mathbf{P}^\mathsf{T}\mathbf{S}\mathbf{P})(\mathbf{P}^\mathsf{T}\mathbf{x}) = \mathbf{P}^\mathsf{T}(\mathbf{S}\mathbf{x})$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^{\mathsf{T}}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^{\mathsf{T}}\mathbf{P}$$

$\Rightarrow$ Podemos reordenar primero y operar después u operar y luego reordenar $(\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})(\mathbf{P}^{\mathsf{T}}\mathbf{x}) = \mathbf{P}^{\mathsf{T}}(\mathbf{S}\mathbf{x})$

- La permutación es equivalente a reordenar los nodos del grafo

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^{\mathsf{T}}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^{\mathsf{T}}\mathbf{P}$$

$\Rightarrow$ Podemos reordenar primero y operar después u operar y luego reordenar $(\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})(\mathbf{P}^{\mathsf{T}}\mathbf{x}) = \mathbf{P}^{\mathsf{T}}(\mathbf{S}\mathbf{x})$

- La permutación es equivalente a reordenar los nodos del grafo
- Elegir un GSO $\mathbf{S}$ para describir un grafo fuerza un ordenamiento de los nodos

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^{\mathsf{T}}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^{\mathsf{T}}\mathbf{P}$$

$\Rightarrow$ Podemos reordenar primero y operar después u operar y luego reordenar $(\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})(\mathbf{P}^{\mathsf{T}}\mathbf{x}) = \mathbf{P}^{\mathsf{T}}(\mathbf{S}\mathbf{x})$

- La permutación es equivalente a reordenar los nodos del grafo
- Elegir un GSO $\mathbf{S}$ para describir un grafo fuerza un ordenamiento de los nodos
  - $\Rightarrow$ Este ordenamiento es necesario para operar

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^{\mathsf{T}}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^{\mathsf{T}}\mathbf{P}$$

$\Rightarrow$ Podemos reordenar primero y operar después u operar y luego reordenar $(\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})(\mathbf{P}^{\mathsf{T}}\mathbf{x}) = \mathbf{P}^{\mathsf{T}}(\mathbf{S}\mathbf{x})$

- La permutación es equivalente a reordenar los nodos del grafo
- Elegir un GSO $\mathbf{S}$ para describir un grafo fuerza un ordenamiento de los nodos
  - $\Rightarrow$ Este ordenamiento es necesario para operar. Pero es arbitrario

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance y Reordenamiento de Nodos

- Una matriz de permutación es ortonormal ¿Cuánto es la identidad reordenada?

$$\mathbf{P}^{\mathsf{T}}\mathbf{I}\mathbf{P} = \mathbf{I} = \mathbf{P}^{\mathsf{T}}\mathbf{P}$$

$\Rightarrow$ Podemos reordenar primero y operar después u operar y luego reordenar $(\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})(\mathbf{P}^{\mathsf{T}}\mathbf{x}) = \mathbf{P}^{\mathsf{T}}(\mathbf{S}\mathbf{x})$

- La permutación es equivalente a reordenar los nodos del grafo
- Elegir un GSO $\mathbf{S}$ para describir un grafo fuerza un ordenamiento de los nodos
  - $\Rightarrow$ Este ordenamiento es necesario para operar. Pero es arbitrario
- Queremos algoritmos de procesamiento de señales que sean independientes de ordenamientos arbitrarios

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k\, \mathbf{S}^k\, \mathbf{x}$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones.

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ *y una señal en el grafo* $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ *permutados*

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ permutados se cumple que

$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ permutados se cumple que

$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^\mathsf{T}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^\mathsf{T}\mathbf{x}$ permutados se cumple que

$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^\mathsf{T}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ permutados se cumple que

$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ permutados se cumple que

$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Prueba $\Rightarrow$ $\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \sum_{k=0}^{\infty} h_k \, \hat{\mathbf{S}}^k \hat{\mathbf{x}}$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^\mathsf{T}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^\mathsf{T}\mathbf{x}$ permutados se cumple que

$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^\mathsf{T}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Prueba $\Rightarrow$ $\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \sum_{k=0}^{\infty} h_k \hat{\mathbf{S}}^k \hat{\mathbf{x}} = \sum_{k=0}^{\infty} h_k (\mathbf{P}^\mathsf{T}\mathbf{S}\mathbf{P})^k \mathbf{P}^\mathsf{T}\mathbf{x}$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ permutados se cumple que

$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Prueba $\Rightarrow$ $\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \sum_{k=0}^{\infty} h_k \, \hat{\mathbf{S}}^k \hat{\mathbf{x}} = \sum_{k=0}^{\infty} h_k \, (\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})^k \mathbf{P}^{\mathsf{T}}\mathbf{x} = \mathbf{P}^{\mathsf{T}} \left(\sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \mathbf{x} \right)$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance

- Consideremos la convolución en grafos $\mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \, \mathbf{x}$
- Depende de los parámetros del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ y del shift operator $\mathbf{S}$; aplicado a la señal de entrada $\mathbf{x}$

Teorema

Las convoluciones en grafos son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ permutados se cumple que

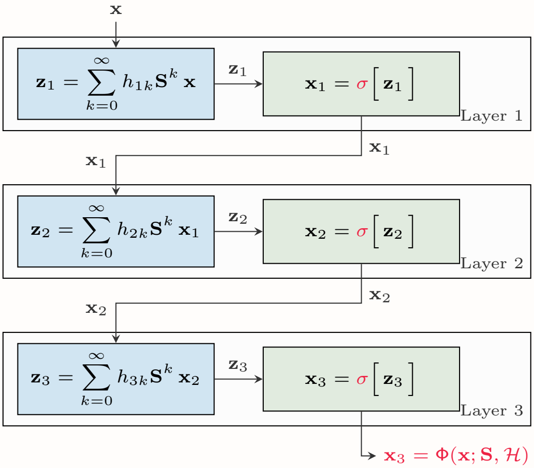
$$\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\mathbf{x}$$

Prueba $\Rightarrow$ $\mathbf{H}(\hat{\mathbf{S}})\hat{\mathbf{x}} = \sum_{k=0}^{\infty} h_k \, \hat{\mathbf{S}}^k \hat{\mathbf{x}} = \sum_{k=0}^{\infty} h_k \, (\mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P})^k \mathbf{P}^{\mathsf{T}}\mathbf{x} = \mathbf{P}^{\mathsf{T}} \left(\sum_{k=0}^{\infty} h_k \, \mathbf{S}^k \mathbf{x} \right) = \mathbf{P}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\mathbf{x}$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Las GNNs heredan la Permutation Equivariance de los grafos

- Una GNN es una composición de capas
  $\Rightarrow$ Filtros en grafos y no-linealidades punto-a-punto



Gama, Bruna, Ribeiro, “Stability Properties of Graph Neural Networks”, IEEE TSP 2020

Las GNNs heredan la Permutation Equivariance de los grafos

- Una GNN es una composición de capas
  $\Rightarrow$ Filtros en grafos y no-linealidades punto-a-punto
- Una operación punto-a-punto no mezcla los valores en los nodos

Layer 1

$$\mathbf{z}_1 = \sum_{k=0}^{\infty} h_{1k} \mathbf{S}^k \mathbf{x} \xrightarrow{\mathbf{z}_1} \mathbf{x}_1 = \sigma\left[\mathbf{z}_1\right]$$

Layer 2

$$\mathbf{z}_2 = \sum_{k=0}^{\infty} h_{2k} \mathbf{S}^k \mathbf{x}_1 \xrightarrow{\mathbf{z}_2} \mathbf{x}_2 = \sigma\left[\mathbf{z}_2\right]$$

Layer 3

$$\mathbf{z}_3 = \sum_{k=0}^{\infty} h_{3k} \mathbf{S}^k \mathbf{x}_2 \xrightarrow{\mathbf{z}_3} \mathbf{x}_3 = \sigma\left[\mathbf{z}_3\right]$$

$$\mathbf{x}_3 = \Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

Gama, Bruna, Ribeiro, “Stability Properties of Graph Neural Networks”, IEEE TSP 2020

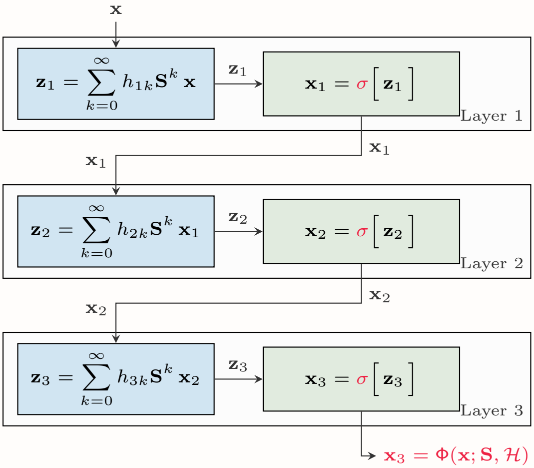
Las GNNs heredan la Permutation Equivariance de los grafos

- Una GNN es una composición de capas

  $\Rightarrow$ Filtros en grafos y no-linealidades punto-a-punto

- Una operación punto-a-punto no mezcla los valores en los nodos

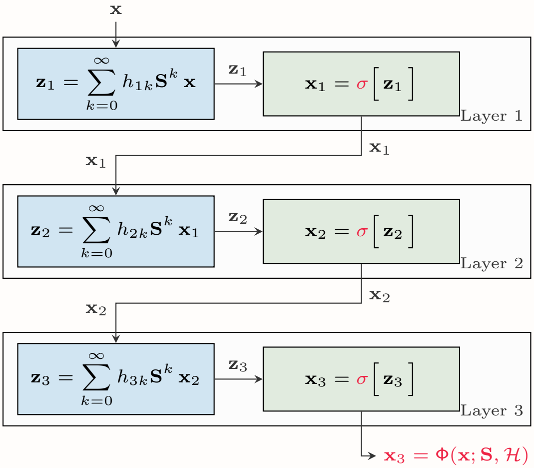
  $\Rightarrow$ Independiente del grafo



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Las GNNs heredan la Permutation Equivariance de los grafos

- Una GNN es una composición de capas
  - $\Rightarrow$ Filtros en grafos y no-linealidades punto-a-punto
- Una operación punto-a-punto no mezcla los valores en los nodos
  - $\Rightarrow$ Independiente del grafo
- La GNN conserva la permutation equivariance



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^\top \mathbf{S} \mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^\top \mathbf{x}$ permutados se cumple que

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^\top \Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

donde $\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$ es la salida de procesar $\hat{\mathbf{x}}$ en $\hat{\mathbf{S}}$ con la GNN $\mathcal{H}$ y $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ es la salida de procesar $\mathbf{x}$ con $\mathbf{S}$ usando la misma GNN $\mathcal{H}$.

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones.

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^\top \Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^{\mathsf{T}}\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ y una señal en el grafo $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ permutados

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^{\mathsf{T}}\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ *y una señal en el grafo* $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ *permutados se cumple que*

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^{\mathsf{T}}\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^\top \mathbf{S} \mathbf{P}$ *y una señal en el grafo* $\hat{\mathbf{x}} = \mathbf{P}^\top \mathbf{x}$ *permutados se cumple que*

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^\top \Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ *y una señal en el grafo* $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ *permutados se cumple que*

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^{\mathsf{T}}\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^{\mathsf{T}}\mathbf{S}\mathbf{P}$ *y una señal en el grafo* $\hat{\mathbf{x}} = \mathbf{P}^{\mathsf{T}}\mathbf{x}$ *permutados se cumple que*

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^{\mathsf{T}}\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

donde $\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$ *es la salida de procesar* $\hat{\mathbf{x}}$ *en* $\hat{\mathbf{S}}$ *con la GNN* $\mathcal{H}$ *y* $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ *es la salida de procesar* $\mathbf{x}$ *con* $\mathbf{S}$ *usando la misma GNN* $\mathcal{H}$.

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Permutation Equivariance en las GNNs

Teorema

Las GNNs son equivariantes a permutaciones. Para un grafo con shift operator $\hat{\mathbf{S}} = \mathbf{P}^\mathsf{T}\mathbf{S}\mathbf{P}$ *y una señal en el grafo* $\hat{\mathbf{x}} = \mathbf{P}^\mathsf{T}\mathbf{x}$ *permutados se cumple que*

$$\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H}) = \mathbf{P}^\mathsf{T}\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$$

donde $\Phi(\hat{\mathbf{x}}; \hat{\mathbf{S}}, \mathcal{H})$ *es la salida de procesar* $\hat{\mathbf{x}}$ *en* $\hat{\mathbf{S}}$ *con la GNN* $\mathcal{H}$ *y* $\Phi(\mathbf{x}; \mathbf{S}, \mathcal{H})$ *es la salida de procesar* $\mathbf{x}$ *con* $\mathbf{S}$ *usando la misma GNN* $\mathcal{H}$*.*

- El procesamiento de señales usando GNNs es independiente del orden y el etiquetado

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

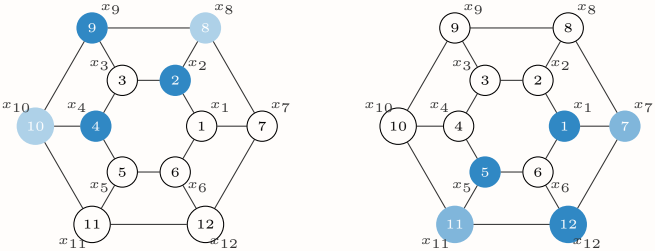
La Equivarianza a Permutaciones es muy Valiosa

- La invarianza al re-etiquetado de los nodos permite a las GNNs aprovechar la simetría de la señal

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

La Equivarianza a Permutaciones es muy Valiosa

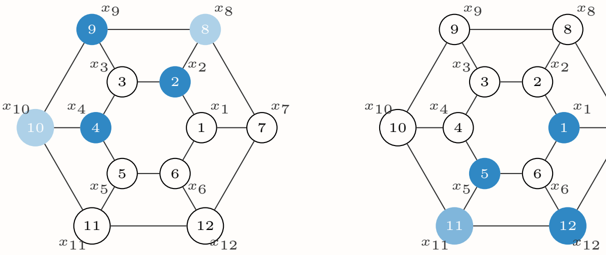
- La invarianza al re-etiquetado de los nodos permite a las GNNs aprovechar la simetría de la señal



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

La Equivarianza a Permutaciones es muy Valiosa

- La invarianza al re-etiquetado de los nodos permite a las GNNs aprovechar la simetría de la señal

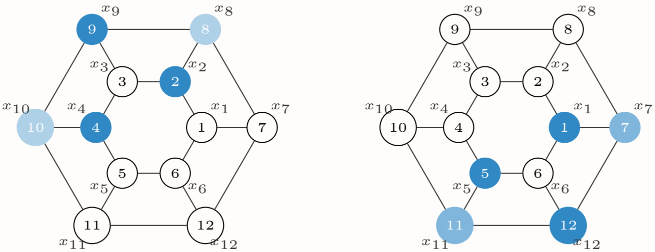


- Aunque diferentes, ambas señales con permutaciones una de la otra

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

La Equivarianza a Permutaciones es muy Valiosa

- La invarianza al re-etiquetado de los nodos permite a las GNNs aprovechar la simetría de la señal



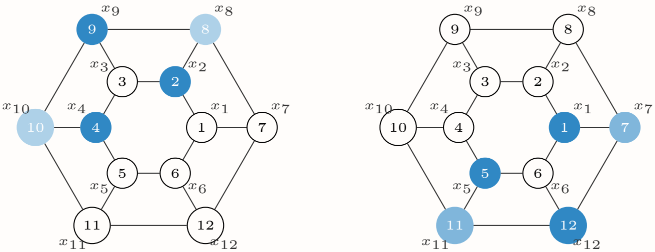
- Aunque diferentes, ambas señales con permutaciones una de la otra

  $\Rightarrow$ Permutation equivariance $\Rightarrow$ La GNN puede aprender a predecir la situación de la derecha

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

La Equivarianza a Permutaciones es muy Valiosa

- La invarianza al re-etiquetado de los nodos permite a las GNNs aprovechar la simetría de la señal



- Aunque diferentes, ambas señales con permutaciones una de la otra

  $\Rightarrow$ Permutation equivariance $\Rightarrow$ La GNN puede aprender a predecir la situación de la derecha

- Permutation Equivariance no es siempre una buena idea
  - Edge-Variant GNNs
  - Feature Augmentation
  - Attention

Isufi, Gama, Ribeiro, "EdgeNets: Edge Varying Graph Neural Networks", IEEE TPAMI 2021

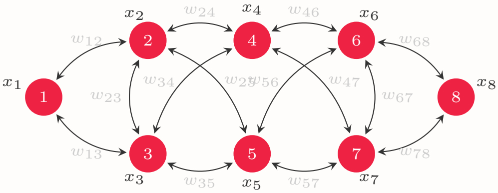
You, Gomes-Selman, Ying, Leskovec, "Identity-aware Graph Neural Networks", IEEE AAAI 2021

Veličković, Cucurull, Casanova, Romero, Liò, Bengio, "Graph Attention Networks", ICLR 2018

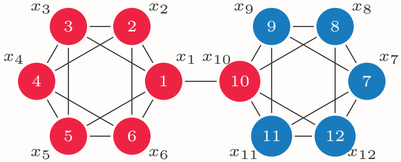
Convolución: Cambios en el grafo

- La convolución en grafos agrega información desde lo local a lo global
  - $\Rightarrow$ Igual que la clásica convolución en el tiempo o espacio

Graph Filter en un grafo



Mismo Graph Filter en otro grafo

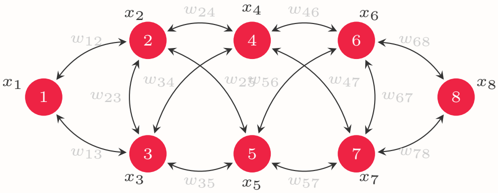


- Usando coeficientes $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ $\Rightarrow$ $\mathbf{z} = h_0\mathbf{S}^0\mathbf{x} + h_1\mathbf{S}^1\mathbf{x} + h_2\mathbf{S}^2\mathbf{x} + h_3\mathbf{S}^3\mathbf{x}$
- La salida de un grafo depende de los coeficientes $\mathbf{h}$ y el GSO $\mathbf{S}$

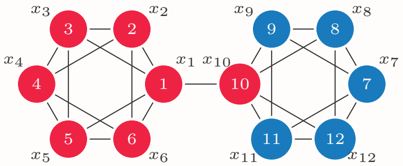
Convolución: Cambios en el grafo

- La convolución en grafos agrega información desde lo local a lo global
  $\Rightarrow$ Igual que la clásica convolución en el tiempo o espacio

Graph Filter en un grafo



Mismo Graph Filter en otro grafo



- Usando coeficientes $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ $\Rightarrow$ $\mathbf{z} = h_0\mathbf{S}^0\mathbf{x} + h_1\mathbf{S}^1\mathbf{x} + h_2\mathbf{S}^2\mathbf{x} + h_3\mathbf{S}^3\mathbf{x}$
- La salida de un grafo depende de los coeficientes **h** y el GSO **S**
- ¿Y si cambiamos el grafo? Lo vamos a estudiar via la transformada de Fourier en grafos

La Graph Fourier Transform

- Los filtros en grafos tienen una representación punto-a-punto en el dominio de la frecuencia

La Graph Fourier Transform

- Los filtros en grafos tienen una representación punto-a-punto en el dominio de la frecuencia

  $\Rightarrow$ La misma propiedad que la convolución temporal

La Graph Fourier Transform

- Los filtros en grafos tienen una representación punto-a-punto en el dominio de la frecuencia
  $\Rightarrow$ La misma propiedad que la convolución temporal
- Suponiendo que el GSO es normal; e.g., porque el grafo es no-dirigido $\Rightarrow$ $\mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathsf{H}}$

La Graph Fourier Transform

- Los filtros en grafos tienen una representación punto-a-punto en el dominio de la frecuencia
  $\Rightarrow$ La misma propiedad que la convolución temporal
- Suponiendo que el GSO es normal; e.g., porque el grafo es no-dirigido $\Rightarrow \mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{H}}$
- El Graph Fourier transform (GFT) de la señal $\mathbf{x}$ es otra señal $\tilde{\mathbf{x}} = \mathbf{V}^{\mathsf{H}}\mathbf{x}$

La Graph Fourier Transform

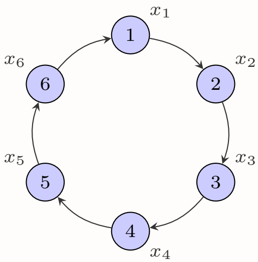
- Los filtros en grafos tienen una representación punto-a-punto en el dominio de la frecuencia
  $\Rightarrow$ La misma propiedad que la convolución temporal
- Suponiendo que el GSO es normal; e.g., porque el grafo es no-dirigido $\Rightarrow$ $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{H}}$
- El Graph Fourier transform (GFT) de la señal $\mathbf{x}$ es otra señal $\tilde{\mathbf{x}} = \mathbf{V}^{\mathsf{H}}\mathbf{x}$
- La inverse Graph Fourier transform (iGFT) del GFT $\tilde{\mathbf{x}}$ es la señal $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$

La Graph Fourier Transform

- Los filtros en grafos tienen una representación punto-a-punto en el dominio de la frecuencia
  $\Rightarrow$ La misma propiedad que la convolución temporal
- Suponiendo que el GSO es normal; e.g., porque el grafo es no-dirigido $\Rightarrow \mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathsf{H}}$
- El Graph Fourier transform (GFT) de la señal $\mathbf{x}$ es otra señal $\tilde{\mathbf{x}} = \mathbf{V}^{\mathsf{H}}\mathbf{x}$
- La inverse Graph Fourier transform (iGFT) del GFT $\tilde{\mathbf{x}}$ es la señal $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$
- Decimos que la GFT $\tilde{\mathbf{x}}$ es una representación de $\mathbf{x}$ en el dominio espectral (frecuencia) del grafo

La Graph Fourier Transform

- Los filtros en grafos tienen una representación punto-a-punto en el dominio de la frecuencia
  $\Rightarrow$ La misma propiedad que la convolución temporal
- Suponiendo que el GSO es normal; e.g., porque el grafo es no-dirigido $\Rightarrow \mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{H}}$
- El Graph Fourier transform (GFT) de la señal $\mathbf{x}$ es otra señal $\tilde{\mathbf{x}} = \mathbf{V}^{\mathsf{H}}\mathbf{x}$
- La inverse Graph Fourier transform (iGFT) del GFT $\tilde{\mathbf{x}}$ es la señal $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$
- Decimos que la GFT $\tilde{\mathbf{x}}$ es una representación de $\mathbf{x}$ en el dominio espectral (frecuencia) del grafo



- Estructura cíclica $\Rightarrow$ Transformada de Fourier $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}^H\mathbf{x}$ $\left(F_{kn} = \frac{e^{j2\pi kn/N}}{\sqrt{N}}\right)$
- **Transformada de Fourier** $\Rightarrow$ Proyección en el *eigenvector space* del ciclo

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$.
- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. $\mathbf{\Lambda}$ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$.
- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. Λ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$.
- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. $\mathbf{\Lambda}$ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$. GFT de la salida $\mathbf{V}^{\mathsf{H}}\mathbf{y} = \tilde{\mathbf{y}}$. GFT de la entrada $\mathbf{V}^{\mathsf{H}}\mathbf{x} = \tilde{\mathbf{x}}$. Se tacha $\mathbf{V}^{\mathsf{H}}\mathbf{V}$

$$\mathbf{V}^{\mathsf{H}}\mathbf{y} = \mathbf{V}^{\mathsf{H}} \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x} \quad \Rightarrow \quad \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \mathbf{\Lambda}^k\right)\tilde{\mathbf{x}}$$

- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. Λ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$. GFT de la salida $\mathbf{V}^{\mathsf{H}}\mathbf{y} = \tilde{\mathbf{y}}$. GFT de la entrada $\mathbf{V}^{\mathsf{H}}\mathbf{x} = \tilde{\mathbf{x}}$. Se tacha $\mathbf{V}^{\mathsf{H}}\mathbf{V}$

$$\mathbf{V}^{\mathsf{H}}\mathbf{y} = \mathbf{V}^{\mathsf{H}} \sum_{k=0}^{\infty} h_k \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x} \qquad \Rightarrow \qquad \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \boldsymbol{\Lambda}^k\right)\tilde{\mathbf{x}}$$

- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. Λ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$. GFT de la salida $\mathbf{V}^{\mathsf{H}}\mathbf{y} = \tilde{\mathbf{y}}$. GFT de la entrada $\mathbf{V}^{\mathsf{H}}\mathbf{x} = \tilde{\mathbf{x}}$. Se tacha $\mathbf{V}^{\mathsf{H}}\mathbf{V}$

$$\mathbf{V}^{\mathsf{H}}\mathbf{y} = \mathbf{V}^{\mathsf{H}} \sum_{k=0}^{\infty} h_k \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x} \quad \Rightarrow \quad \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \boldsymbol{\Lambda}^k\right)\tilde{\mathbf{x}}$$

- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. Λ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$. GFT de la salida $\mathbf{V}^{\mathsf{H}}\mathbf{y} = \tilde{\mathbf{y}}$. GFT de la entrada $\mathbf{V}^{\mathsf{H}}\mathbf{x} = \tilde{\mathbf{x}}$. Se tacha $\mathbf{V}^{\mathsf{H}}\mathbf{V}$

$$\mathbf{V}^{\mathsf{H}}\mathbf{y} = \mathbf{V}^{\mathsf{H}} \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x} \qquad \Rightarrow \qquad \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \mathbf{\Lambda}^k\right)\tilde{\mathbf{x}}$$

- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. Λ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$. GFT de la salida $\mathbf{V}^{\mathsf{H}}\mathbf{y} = \tilde{\mathbf{y}}$. GFT de la entrada $\mathbf{V}^{\mathsf{H}}\mathbf{x} = \tilde{\mathbf{x}}$. Se tacha $\mathbf{V}^{\mathsf{H}}\mathbf{V}$

$$\mathbf{V}^{\mathsf{H}}\mathbf{y} = \mathbf{V}^{\mathsf{H}} \sum_{k=0}^{\infty} h_k \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x} \quad \Rightarrow \quad \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \mathbf{\Lambda}^k\right)\tilde{\mathbf{x}}$$

- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. $\mathbf{\Lambda}$ es diagonal

Convolución en el Dominio de la Frecuencia

- Definimos la convolución en grafos como un polinomio en el GSO: $\mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$
- Descomposición espectral del GSO implica $\mathbf{S}^k = \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}$. $\Rightarrow \mathbf{y} = \sum_{k=0}^{\infty} h_k \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x}$
- Multiplico por $\mathbf{V}^{\mathsf{H}}$. GFT de la salida $\mathbf{V}^{\mathsf{H}}\mathbf{y} = \tilde{\mathbf{y}}$. GFT de la entrada $\mathbf{V}^{\mathsf{H}}\mathbf{x} = \tilde{\mathbf{x}}$. Se tacha $\mathbf{V}^{\mathsf{H}}\mathbf{V}$

$$\mathbf{V}^{\mathsf{H}}\mathbf{y} = \mathbf{V}^{\mathsf{H}}\sum_{k=0}^{\infty} h_k \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{\mathsf{H}}\mathbf{x} \qquad \Rightarrow \qquad \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \boldsymbol{\Lambda}^k\right)\tilde{\mathbf{x}}$$

- La GFT de las señales de entrada y salida se relacionan por una matriz diagonal. $\boldsymbol{\Lambda}$ es diagonal

Respuesta en Frecuencia del Grafo

- Convolución representada en el dominio de la frecuencia (espectro) $\Rightarrow \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \mathbf{\Lambda}^k \right) \tilde{\mathbf{x}}$

Respuesta en Frecuencia del Grafo

- Convolución representada en el dominio de la frecuencia (espectro) $\Rightarrow \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \mathbf{\Lambda}^k \right) \tilde{\mathbf{x}}$
- Las convoluciones son punto-a-punto en el espectro $\Rightarrow \tilde{y}_i = \left(\sum_{k=0}^{\infty} h_k \lambda_i^k \right) \tilde{x}_i$

Respuesta en Frecuencia del Grafo

- Convolución representada en el dominio de la frecuencia (espectro) $\Rightarrow \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \mathbf{\Lambda}^k\right) \tilde{\mathbf{x}}$
- Las convoluciones son punto-a-punto en el espectro $\Rightarrow \tilde{y}_i = \left(\sum_{k=0}^{\infty} h_k \lambda_i^k\right) \tilde{x}_i$

Definición

Dado un graph filter con coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$, la respuesta en frecuencia (función de transferencia) del grafo es el polinomio

$$\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$$

Respuesta en Frecuencia del Grafo

- Convolución representada en el dominio de la frecuencia (espectro) $\Rightarrow \tilde{\mathbf{y}} = \left(\sum_{k=0}^{\infty} h_k \mathbf{\Lambda}^k \right) \tilde{\mathbf{x}}$
- Las convoluciones son punto-a-punto en el espectro $\Rightarrow \tilde{y}_i = \left(\sum_{k=0}^{\infty} h_k \lambda_i^k \right) \tilde{x}_i$

Definición

Dado un graph filter con coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$, la respuesta en frecuencia (función de transferencia) del grafo es el polinomio

$$\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$$

- De esta forma podemos escribir $\Rightarrow \tilde{y}_i = \tilde{\mathsf{h}}(\lambda_i) \tilde{x}_i$

Graph Frequency Response

Definición

Dado un grafo con coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$, la respuesta en frecuencia (función de transferencia) del grafo es el polinomio

$$\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$$

- La respuesta en frecuencia es el mismo polinomio que define el filtro pero en la variable escalar λ

Graph Frequency Response

Definición

Dado un grafo con coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$, la respuesta en frecuencia (función de transferencia) del grafo es el polinomio

$$\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$$

- La respuesta en frecuencia es el mismo polinomio que define el filtro pero en la variable escalar λ
- La respuesta en frecuencia es independendiente del grafo.

Graph Frequency Response

Definición

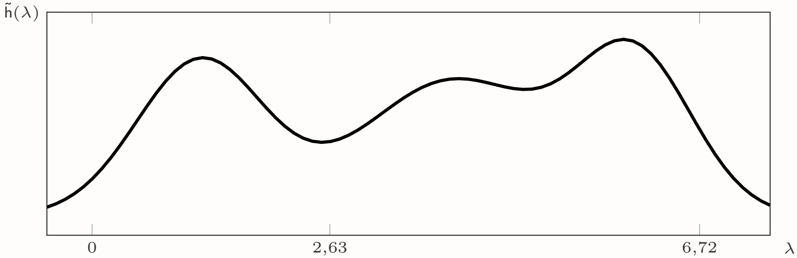
Dado un grafo con coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$, la respuesta en frecuencia (función de transferencia) del grafo es el polinomio

$$\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$$

- La respuesta en frecuencia es el mismo polinomio que define el filtro pero en la variable escalar λ
- La respuesta en frecuencia es independendiente del grafo. Está determinada únicamente por los coeficientes (taps) del filtro
- El rol del grafo es determinar los valores propios en los que se evaluará el polinomio

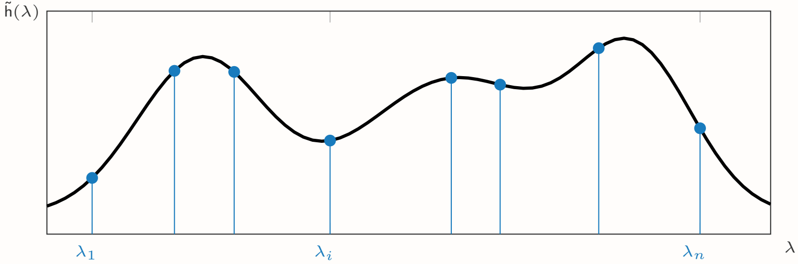
Usar el Mismo Filtro en Distintos Grafos

- Dados los coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$ la respuesta en frecuencia del filtro resulta $\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$



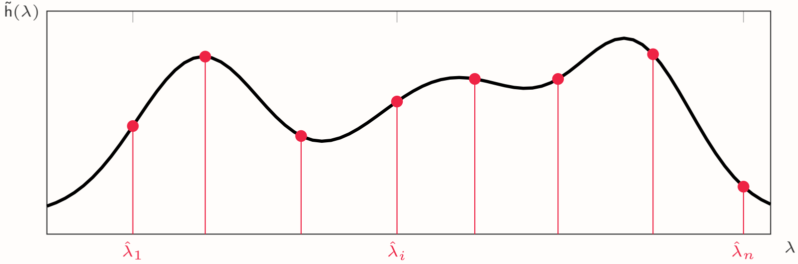
Usar el Mismo Filtro en Distintos Grafos

- Dados los coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$ la respuesta en frecuencia del filtro resulta $\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$
- Para un grafo específico, se evalúa en sus valor propios λ_i



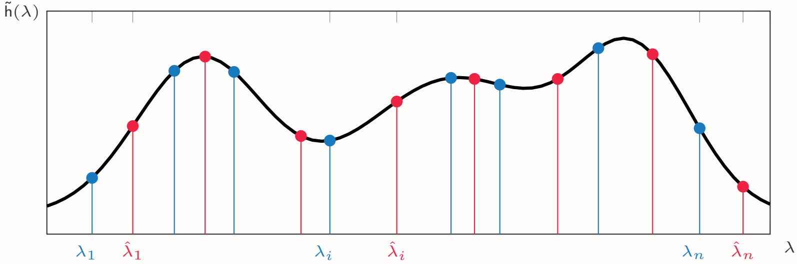
Usar el Mismo Filtro en Distintos Grafos

- Dados los coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$ la respuesta en frecuencia del filtro resulta $\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$
- Para un grafo específico, se evalúa en sus valor propios λ_i
- Para un grafo diferente la respuesta se evalúa en valores propios distintos $\hat{\lambda}_i$



Usar el Mismo Filtro en Distintos Grafos

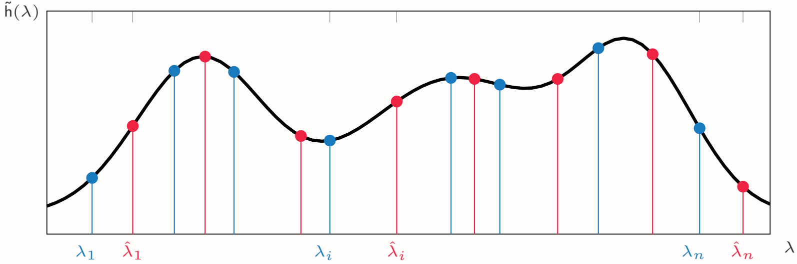
- Dados los coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$ la respuesta en frecuencia del filtro resulta $\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$
- Para un grafo específico, se evalúa en sus valor propios λ_i
- Para un grafo diferente la respuesta se evalúa en valores propios distintos $\hat{\lambda}_i$



- Clave para el análisis de perturbación del grafo $\Rightarrow$

Usar el Mismo Filtro en Distintos Grafos

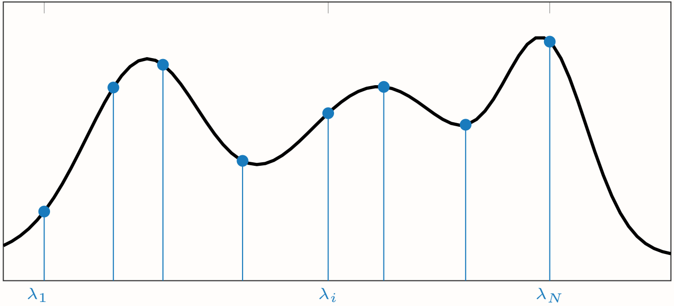
- Dados los coeficientes $\mathbf{h} = \{h_k\}_{k=1}^{\infty}$ la respuesta en frecuencia del filtro resulta $\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$
- Para un grafo específico, se evalúa en sus valor propios λ_i
- Para un grafo diferente la respuesta se evalúa en valores propios distintos $\hat{\lambda}_i$



- Clave para el análisis de perturbación del grafo $\Rightarrow$ explicará el buen desempeño de las GNNs

Estabilidad ante perturbaciones

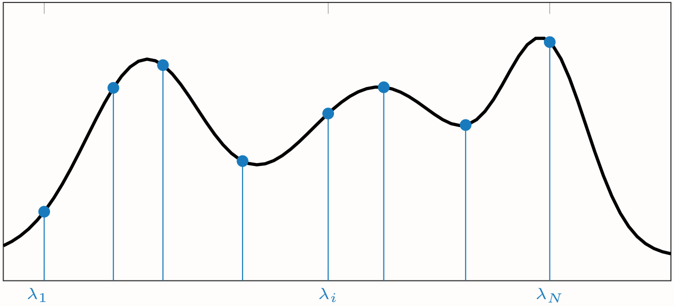
- Caso sencillo de perturbación: dilatación de aristas $\Rightarrow \hat{\mathbf{S}} = (1+\varepsilon)\mathbf{S}$



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Estabilidad ante perturbaciones

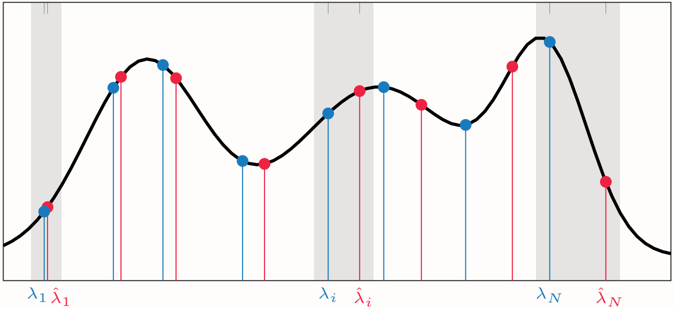
- Caso sencillo de perturbación: dilatación de aristas $\Rightarrow \hat{\mathbf{S}} = (1+\varepsilon)\mathbf{S}$
- En este caso, se produce una dilatación espectral $\Rightarrow \hat{\lambda}_i = (1+\varepsilon)\lambda_i$ (y los vectores propios no cambian)



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Estabilidad ante perturbaciones

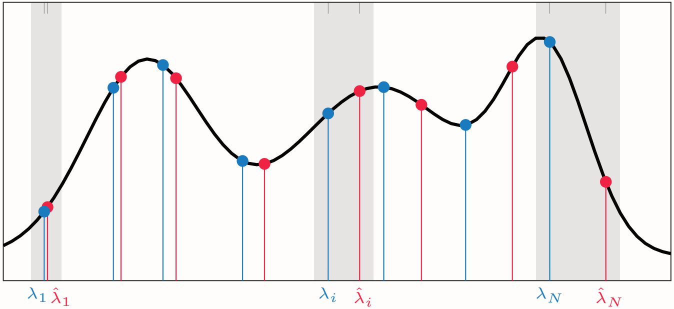
- Caso sencillo de perturbación: dilatación de aristas $\Rightarrow \hat{\mathbf{S}} = (1+\varepsilon)\mathbf{S}$
- En este caso, se produce una dilatación espectral $\Rightarrow \hat{\lambda}_i = (1+\varepsilon)\lambda_i$ (y los vectores propios no cambian)



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Estabilidad ante perturbaciones

- Caso sencillo de perturbación: dilatación de aristas $\Rightarrow \hat{\mathbf{S}} = (1+\varepsilon)\mathbf{S}$
- En este caso, se produce una dilatación espectral $\Rightarrow \hat{\lambda}_i = (1+\varepsilon)\lambda_i$ (y los vectores propios no cambian)

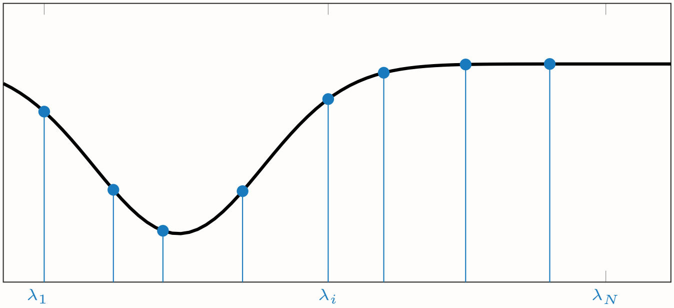


- Pequeñas deformaciones resultan en grandes variaciones a la salida del filtro para valores altos de λ

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Estabilidad ante perturbaciones

- Caso sencillo de perturbación: dilatación de aristas $\Rightarrow \hat{\mathbf{S}} = (1+\varepsilon)\mathbf{S}$
- En este caso, se produce una dilatación espectral $\Rightarrow \hat{\lambda}_i = (1+\varepsilon)\lambda_i$ (y los vectores propios no cambian)

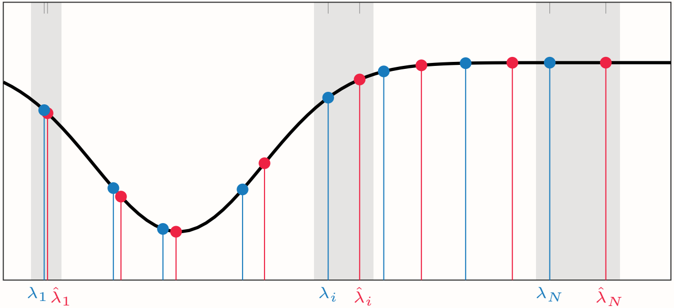


- Si $\tilde{\mathsf{h}}(\lambda)$ suave para valores altos de λ $\Rightarrow$ los valores propios no se mueven o el filtro no cambia

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Estabilidad ante perturbaciones

- Caso sencillo de perturbación: dilatación de aristas $\Rightarrow \hat{\mathbf{S}} = (1+\varepsilon)\mathbf{S}$
- En este caso, se produce una dilatación espectral $\Rightarrow \hat{\lambda}_i = (1+\varepsilon)\lambda_i$ (y los vectores propios no cambian)

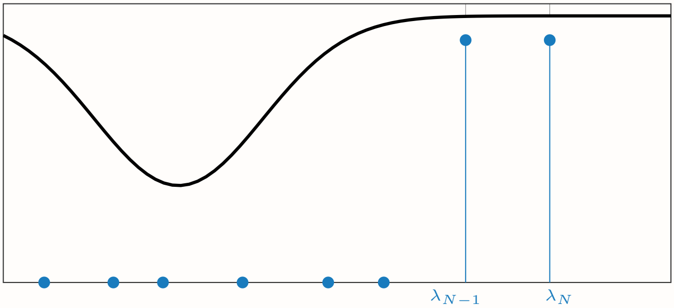


- Si $\tilde{\mathsf{h}}(\lambda)$ suave para valores altos de λ $\Rightarrow$ los valores propios no se mueven o el filtro no cambia

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Banco de Filtros Selectivos son Inestables

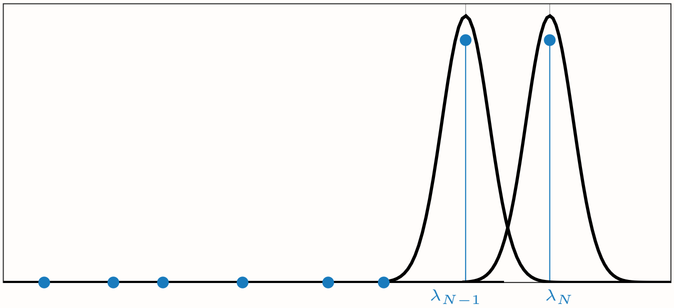
- ¿Qué tienen de malo las convoluciones lineales?
- No pueden ser a la vez estables a deformaciones y discriminar features a frecuencias altas



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Banco de Filtros Selectivos son Inestables

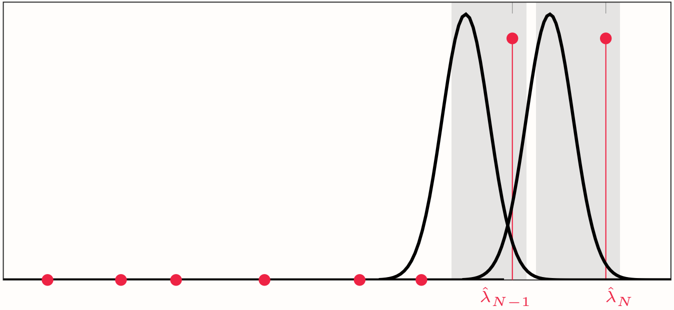
- ¿Qué tienen de malo las convoluciones lineales?
- No pueden ser a la vez estables a deformaciones y discriminar features a frecuencias altas



Gama, Bruna, Ribeiro, “Stability Properties of Graph Neural Networks”, IEEE TSP 2020

Banco de Filtros Selectivos son Inestables

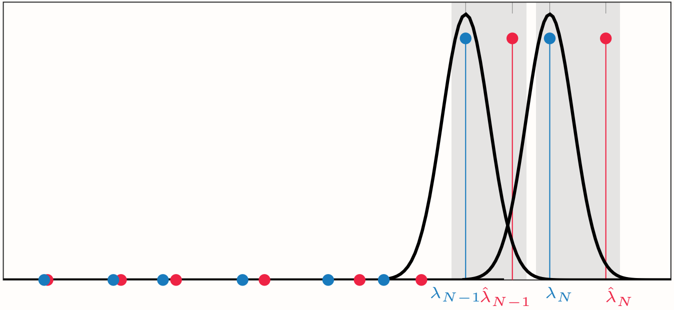
- ¿Qué tienen de malo las convoluciones lineales?
- No pueden ser a la vez estables a deformaciones y discriminar features a frecuencias altas



Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Banco de Filtros Selectivos son Inestables

- ¿Qué tienen de malo las convoluciones lineales?
- No pueden ser a la vez estables a deformaciones y discriminar features a frecuencias altas

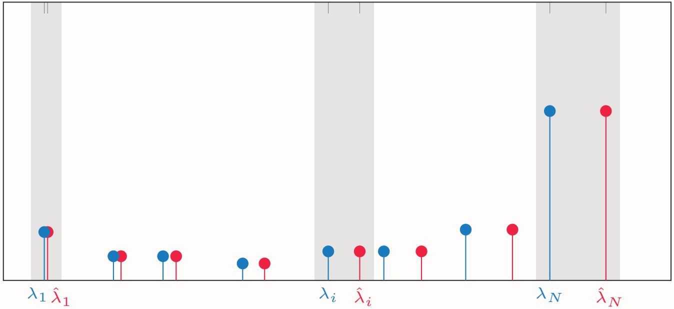


- Limita su utilidad en problemas de aprendizaje automático cuando los features importantes están en frecuencias (valores propios) altos

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Las No-Linealidades Cambian el Soporte del Espectro

- ¿Qué aportan las no-linealidades punto-a-punto?
- Siguen siendo permutation equivariant pero a la vez generan componentes en otras frecuencias (bajas)
  $\Rightarrow$ Que podemos discriminar con filtros estables



Spectrum of rectified graph signal

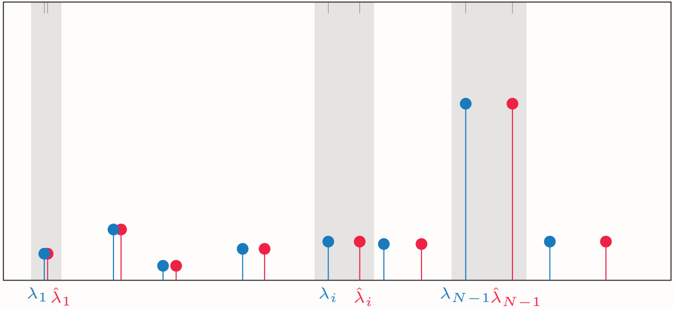
$$\mathbf{x}_{\text{relu}} = \max(\mathbf{x}, 0)$$

- La no-linealidad demodula. Crea nuevos componentes en frecuencias bajas

Gama, Bruna, Ribeiro, “Stability Properties of Graph Neural Networks”, IEEE TSP 2020

Las No-Linealidades Cambian el Soporte del Espectro

- ¿Qué aportan las no-linealidades punto-a-punto?
- Siguen siendo permutation equivariant pero a la vez generan componentes en otras frecuencias (bajas)
  $\Rightarrow$ Que podemos discriminar con filtros estables



Spectrum of rectified graph signal

$$\mathbf{x}_{\text{relu}} = \text{máx}(\mathbf{x}, 0)$$

- La no-linealidad demodula. Crea nuevos componentes en frecuencias bajas

Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

FACULTAD DE INGENIERÍA UDELAR

Las No-Linealidades Cambian el Soporte del Espectro

Las GNNs son arquitecturas de procesamiento de información **estables** y **selectivas**

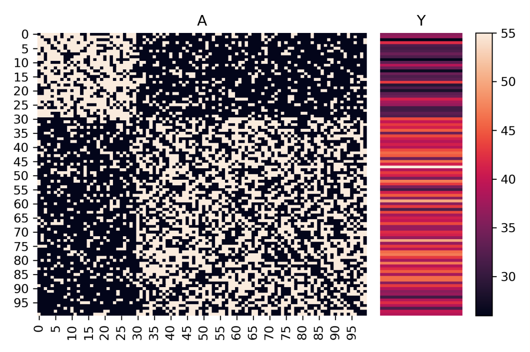
Gama, Bruna, Ribeiro, "Stability Properties of Graph Neural Networks", IEEE TSP 2020

Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)



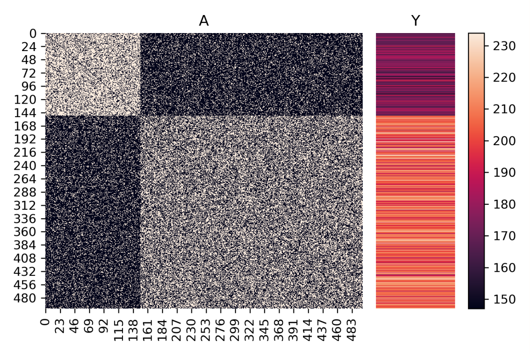
n=100

Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)



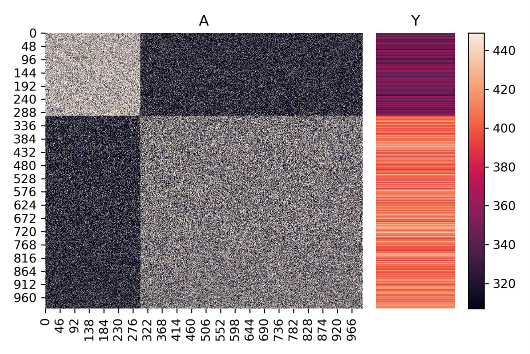
n=500

Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)



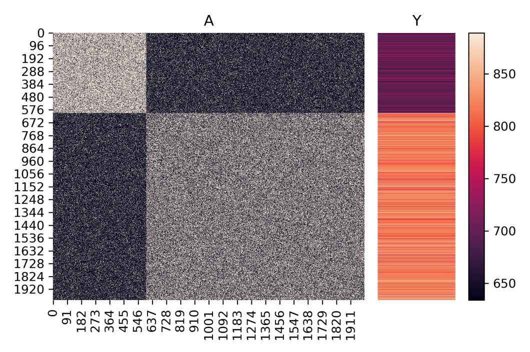
n=1000

Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)



n=2000

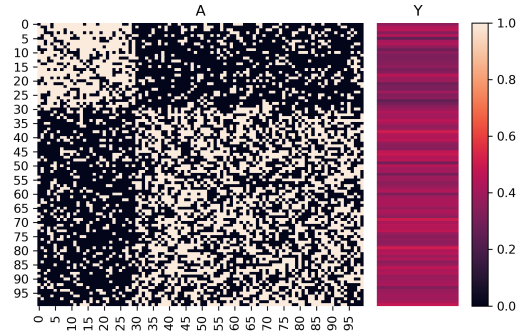
Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)

  - Ahora re-escalando $\mathbf{S}_n = \mathbf{A}/n$.



n=100

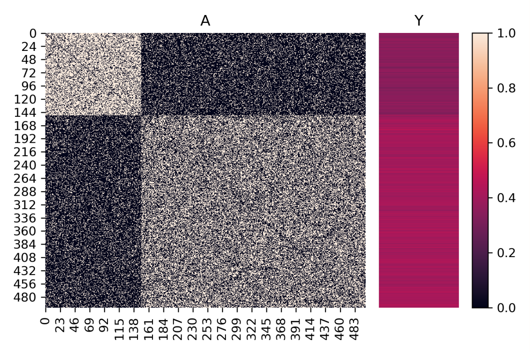
Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos “crecientes”. Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)

  - Ahora re-escalando $\mathbf{S}_n = \mathbf{A}/n$.



n=500

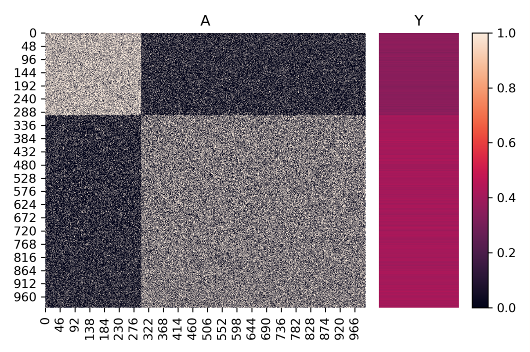
Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{Ax}$ (i.e. $\mathbf{S} = \mathbf{A}$)

  - Ahora re-escalando $\mathbf{S}_n = \mathbf{A}/n$.



n=1000

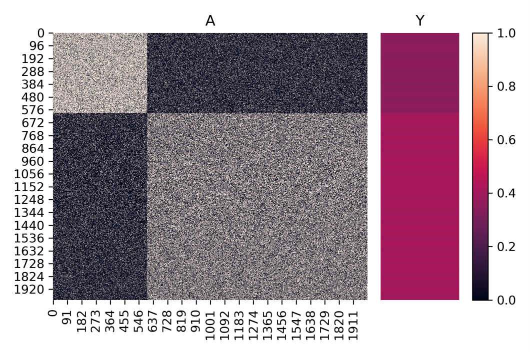
Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)

  - Ahora re-escalando $\mathbf{S}_n = \mathbf{A}/n$.



n=2000

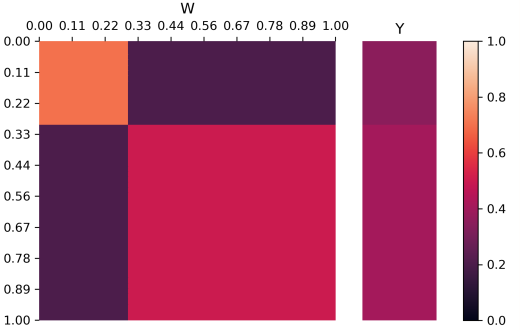
Escalando una GNN

- ¿Cómo se comporta una GNN a medida que el grafo crece?
  - Ya conocemos algunos modelos para grafos "crecientes". Consideremos el grafón
  - Ejemplo: Convolución de la señal $\mathbf{x} = \mathbf{1}$ con un SBM de parámetros:

$$\alpha = [0{,}3, 0{,}7] \quad \pi = \begin{pmatrix} 0{,}7 & 0{,}2 \\ 0{,}2 & 0{,}5 \end{pmatrix}$$

$\Rightarrow$ $\mathbf{y} = \mathbf{A}\mathbf{x}$ (i.e. $\mathbf{S} = \mathbf{A}$)

  - Ahora re-escalando $\mathbf{S}_n = \mathbf{A}/n$. En el infinito...



n=∞

Grafón Shift Operator

- En el infinito (y normalizando el GSO) la convolución en grafos se convierte en una integral en $\mathbf{W}(u, v)$

Grafón Shift Operator

- En el infinito (y normalizando el GSO) la convolución en grafos se convierte en una integral en $\mathbf{W}(u,v)$
- Se define entonces la convolución en grafones, parametrizada por el grafón shift operator

Definición

Sea $X(u)$ una función en $L_2([0,1])$ (señal en el grafón). El grafón shift operator de un grafón $\mathbf{W}$ se define como

$$Y(v) = (T_{\mathbf{W}}X)(v) = \int_0^1 \mathbf{W}(u,v)X(u)du.$$

- El grafón shift operator es un operador lineal e integral con un kernel dado por el grafón $\mathbf{W}$

L. Ruiz, Luiz F.O. Chamon, y A. Ribeiro. "Graphon signal processing" IEEE Transactions on Signal Processing 2021.

Grafón Shift Operator

- En el infinito (y normalizando el GSO) la convolución en grafos se convierte en una integral en $\mathbf{W}(u, v)$
- Se define entonces la convolución en grafones, parametrizada por el grafón shift operator

Definición

Sea $X(u)$ una función en $L_2([0,1])$ (señal en el grafón). El grafón shift operator de un grafón $\mathbf{W}$ se define como

$$Y(v) = (T_{\mathbf{W}}X)(v) = \int_0^1 \mathbf{W}(u, v)X(u)du.$$

- El grafón shift operator es un operador lineal e integral con un kernel dado por el grafón $\mathbf{W}$

- Convolución en grafones $\Rightarrow Z = \quad h_0 \, T_{\mathbf{W}}^0 X$

L. Ruiz, Luiz F.O. Chamon, y A. Ribeiro. "Graphon signal processing" IEEE Transactions on Signal Processing 2021.

Grafón Shift Operator

- En el infinito (y normalizando el GSO) la convolución en grafos se convierte en una integral en $\mathbf{W}(u,v)$
- Se define entonces la convolución en grafones, parametrizada por el grafón shift operator

Definición

Sea $X(u)$ una función en $L_2([0,1])$ (señal en el grafón). El grafón shift operator de un grafón $\mathbf{W}$ se define como

$$Y(v) = (T_{\mathbf{W}}X)(v) = \int_0^1 \mathbf{W}(u,v)X(u)du.$$

- El grafón shift operator es un operador lineal e integral con un kernel dado por el grafón $\mathbf{W}$

- Convolución en grafones $\Rightarrow Z = \; h_0 \, T_{\mathbf{W}}^0 X \; + h_1 \, T_{\mathbf{W}}^1 X$

L. Ruiz, Luiz F.O. Chamon, y A. Ribeiro. "Graphon signal processing" IEEE Transactions on Signal Processing 2021.

Grafón Shift Operator

- En el infinito (y normalizando el GSO) la convolución en grafos se convierte en una integral en $\mathbf{W}(u, v)$
- Se define entonces la convolución en grafones, parametrizada por el grafón shift operator

Definición

Sea $X(u)$ una función en $L_2([0,1])$ (señal en el grafón). El grafón shift operator de un grafón $\mathbf{W}$ se define como

$$Y(v) = (T_{\mathbf{W}} X)(v) = \int_0^1 \mathbf{W}(u, v) X(u) du.$$

- El grafón shift operator es un operador lineal e integral con un kernel dado por el grafón $\mathbf{W}$

- Convolución en grafones $\Rightarrow Z = h_0 T_{\mathbf{W}}^0 X + h_1 T_{\mathbf{W}}^1 X + h_2 T_{\mathbf{W}}^2 X$

L. Ruiz, Luiz F.O. Chamon, y A. Ribeiro. “Graphon signal processing” IEEE Transactions on Signal Processing 2021.

Grafón Shift Operator

- En el infinito (y normalizando el GSO) la convolución en grafos se convierte en una integral en $\mathbf{W}(u, v)$
- Se define entonces la convolución en grafones, parametrizada por el grafón shift operator

Definición

Sea $X(u)$ una función en $L_2([0,1])$ (señal en el grafón). El grafón shift operator de un grafón $\mathbf{W}$ se define como

$$Y(v) = (T_{\mathbf{W}} X)(v) = \int_0^1 \mathbf{W}(u, v) X(u) du.$$

- El grafón shift operator es un operador lineal e integral con un kernel dado por el grafón $\mathbf{W}$
- Convolución en grafones $\Rightarrow Z = \ h_0 \, T_{\mathbf{W}}^0 X \ + h_1 \, T_{\mathbf{W}}^1 X \ + h_2 \, T_{\mathbf{W}}^2 X \ + h_3 \, T_{\mathbf{W}}^3 X$

L. Ruiz, Luiz F.O. Chamon, y A. Ribeiro. “Graphon signal processing” IEEE Transactions on Signal Processing 2021.

Grafón Shift Operator

- En el infinito (y normalizando el GSO) la convolución en grafos se convierte en una integral en $\mathbf{W}(u, v)$
- Se define entonces la convolución en grafones, parametrizada por el grafón shift operator

Definición

Sea $X(u)$ una función en $L_2([0,1])$ (señal en el grafón). El grafón shift operator de un grafón $\mathbf{W}$ se define como

$$Y(v) = (T_{\mathbf{W}} X)(v) = \int_0^1 \mathbf{W}(u, v) X(u) du.$$

- El grafón shift operator es un operador lineal e integral con un kernel dado por el grafón $\mathbf{W}$
- Convolución en grafones $\Rightarrow Z = \; h_0 T_{\mathbf{W}}^0 X \; + h_1 T_{\mathbf{W}}^1 X \; + h_2 T_{\mathbf{W}}^2 X \; + h_3 T_{\mathbf{W}}^3 X \; \ldots = \sum_{k=0}^{K-1} h_k T_{\mathbf{W}}^k X$

L. Ruiz, Luiz F.O. Chamon, y A. Ribeiro. "Graphon signal processing" IEEE Transactions on Signal Processing 2021.

Aproximación de convolución en grafones por convolución en grafos

- ¿Cómo se comporta esta convergencia para n finito?

Teorema

Sea una señal en el grafo $(\mathbf{S}_n, \mathbf{x}_n)$ *muestreada de un grafón* $(\mathbf{W}, X)$ *y las convoluciones* $\mathbf{y}_n = \mathbf{H}(\mathbf{S}_n)\mathbf{x}_n$ *y* $Y = T_{\mathbf{H}} X$*. La norma de la diferencia de las señales está acotada por*

$$\|Y_n - Y\| \leq \left(A_h + \pi \frac{B^c_{\mathbf{W}_n}}{\delta^c_{\mathbf{W}\mathbf{W}_n}}\right)\left(\frac{2(A_w \alpha(n,\chi_1) + \beta(n,\chi_3))}{n}\right)\|X\| + \left(\frac{A_x \alpha(n,\chi_2)(a_h c + 2)}{n}\right) + 4 a_h c \|X\|$$

- La cota es decreciente en n $\Rightarrow$ cuanto más grande el grafo mejor la aproximación del grafón

L. Ruiz, L.F.O. Chamon, y A. Ribeiro. "Graphon signal processing" IEEE Transactions on Signal Processing, 2021.

L. Ruiz, L.F.O. Chamon, y A. Ribeiro. "Transferability properties of graph neural networks" IEEE Transactions on Signal Processing, 2023.

Aproximación de convolución en grafones por convolución en grafos

- ¿Cómo se comporta esta convergencia para n finito?

Teorema

Sea una señal en el grafo $(\mathbf{S}_n, \mathbf{x}_n)$ *muestreada de un grafón* $(\mathbf{W}, X)$ *y las convoluciones* $\mathbf{y}_n = \mathbf{H}(\mathbf{S}_n)\mathbf{x}_n$ *y* $Y = T_{\mathbf{H}} X$. *La norma de la diferencia de las señales está acotada por*

$$\|Y_n - Y\| \leq \left(A_h + \pi \frac{B^c_{\mathbf{W}_n}}{\delta^c_{\mathbf{W}\mathbf{W}_n}}\right)\left(\frac{2(A_w \alpha(n,\chi_1) + \beta(n,\chi_3))}{n}\right)\|X\| + \left(\frac{A_x \alpha(n,\chi_2)(a_h c + 2)}{n}\right) + 4 a_h c \|X\|$$

- La cota es decreciente en n $\Rightarrow$ cuanto más grande el grafo mejor la aproximación del grafón
- Las constantes A_h, a_h y c se refieren a $\tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$:
  - La función $\tilde{\mathsf{h}}(\lambda)$ es A_h-Lipschitz en $[-1, -c] \cup [c, 1]$ (y a_h-Lipschitz en $(-c, c)$, con $a_h < A_h$)
  - O sea, $|\tilde{\mathsf{h}}'(\lambda)| < A_h$ para $|\lambda| > c$ (y $|\tilde{\mathsf{h}}'(\lambda)| < a_h$ para $|\lambda| < c$, con $a_h < A_h$)

L. Ruiz, L.F.O. Chamon, y A. Ribeiro. "Graphon signal processing" IEEE Transactions on Signal Processing, 2021.

L. Ruiz, L.F.O. Chamon, y A. Ribeiro. "Transferability properties of graph neural networks" IEEE Transactions on Signal Processing, 2023.

Aproximación de convolución en grafones por convolución en grafos

- ¿Cómo se comporta esta convergencia para n finito?

Teorema

Sea una señal en el grafo $(\mathbf{S}_n, \mathbf{x}_n)$ *muestreada de un grafón* $(\mathbf{W}, X)$ *y las convoluciones* $\mathbf{y}_n = \mathbf{H}(\mathbf{S}_n)\mathbf{x}_n$ *y* $Y = T_{\mathbf{H}}X$. *La norma de la diferencia de las señales está acotada por*

$$\|Y_n - Y\| \leq \left(A_h + \pi \frac{B^c_{\mathbf{W}_n}}{\delta^c_{\mathbf{W}\mathbf{W}_n}}\right)\left(\frac{2(A_w \alpha(n,\chi_1) + \beta(n,\chi_3))}{n}\right)\|X\| + \left(\frac{A_x \alpha(n,\chi_2)(a_h c + 2)}{n}\right) + 4a_h c\|X\|$$

- La cota es decreciente en n $\Rightarrow$ cuanto más grande el grafo mejor la aproximación del grafón
- Las constantes A_h, a_h y c se refieren a $\tilde{h}(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$:
  - La función $\tilde{h}(\lambda)$ es A_h-Lipschitz en $[-1,-c] \cup [c,1]$ (y a_h-Lipschitz en $(-c,c)$, con $a_h < A_h$)
  - O sea, $|\tilde{h}'(\lambda)| < A_h$ para $|\lambda| > c$ (y $|\tilde{h}'(\lambda)| < a_h$ para $|\lambda| < c$, con $a_h < A_h$)
- ¿Porqué aparecen estas constantes? Lo vamos a ver en el espectro...

L. Ruiz, L.F.O. Chamon, y A. Ribeiro. "Graphon signal processing" IEEE Transactions on Signal Processing, 2021.

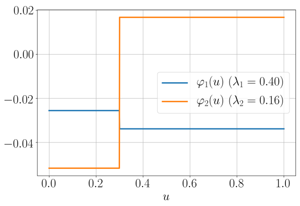
L. Ruiz, L.F.O. Chamon, y A. Ribeiro. "Transferability properties of graph neural networks" IEEE Transactions on Signal Processing, 2023.

Representación espectral de los filtros en grafones

- Los grafones también se pueden descomponer en valores y funciones propias

$$\Rightarrow W(u,v) = \sum_{j\in\mathcal{Z}\setminus\{0\}} \lambda_j \varphi_j(u)\varphi_j(v) \quad (\text{con } \langle \varphi_i, \varphi_j \rangle = \int_0^1 \varphi_i(u)\varphi_j(u)du = 0 \text{ y } \|\varphi_i\| = 1)$$

  - El SBM de dos comunidades de ejemplo tiene sólo dos funciones propias con $\lambda > 0$:

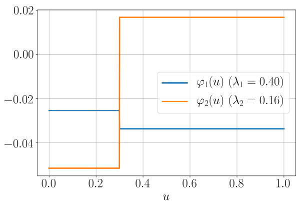


Representación espectral de los filtros en grafones

- Los grafones también se pueden descomponer en valores y funciones propias

$$\Rightarrow W(u,v) = \sum_{j\in\mathcal{Z}\setminus\{0\}} \lambda_j \varphi_j(u)\varphi_j(v) \quad \left(\text{con } \langle \varphi_i, \varphi_j\rangle = \int_0^1 \varphi_i(u)\varphi_j(u)du = 0 \text{ y } \|\varphi_i\| = 1\right)$$

  - El SBM de dos comunidades de ejemplo tiene sólo dos funciones propias con $\lambda > 0$:



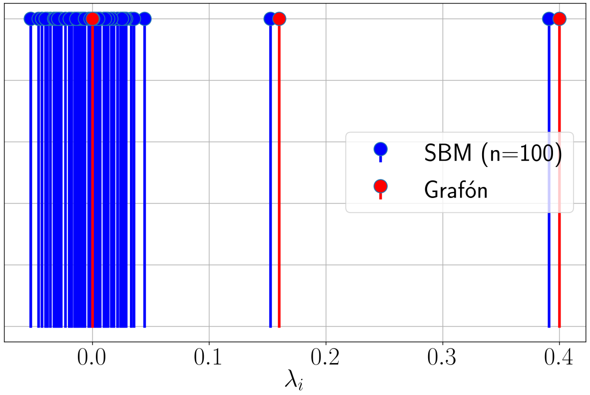
Teorema

Sean un filtro en el grafón con coeficientes h_k, una señal X y la señal filtrada Y. Las Transformadas de Fourier en el Grafón $\tilde{X}_j = \int_0^1 \varphi_j(u)X(u)du$ y $\tilde{Y}_j = \int_0^1 \varphi_j(u)Y(u)du$ son tales que

$$\tilde{Y}_j = \sum_{k=0}^{K-1} h_k \lambda_j^k \tilde{X}_j = \tilde{\mathsf{h}}(\lambda)\tilde{X}_j \qquad \Rightarrow \qquad \tilde{\mathsf{h}}(\lambda) = \sum_{k=0}^{K-1} h_k \lambda^k$$

Representación espectral de los filtros en grafones

- ¿Porqué la aproximación de la convolución en el grafón depende de A_h, a_h y c? Porqué hace falta imponer restricciones sobre la derivada de $\tilde{\mathsf{h}}(\lambda)$ cerca del origen ($|\lambda| < c$)?
  - Ejemplo de valores propios del SBM con $n = 100$ y el grafón. ¿El filtro puede ser muy selectivo en bajas frecuencias?



Graph Neural Networks

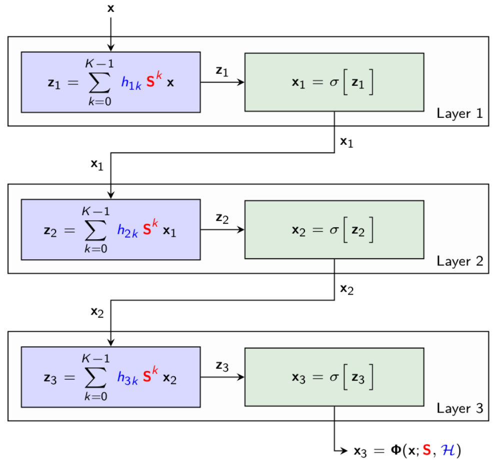
- Una GNN es una cascada de capas
- Cada cual una composición de

  ⇒ convolución en el grafo $\mathbf{H}(\mathbf{S})$

  ⇒ y no-linealidades punto-a-punto σ
- Los parámetros a aprender son el conjunto

  $\mathcal{H} = \{h_{kl}\}$
- Una GNN se puede representar como

  $\mathbf{y} = \mathbf{\Phi}(\mathcal{H}; \mathbf{S}; \mathbf{x})$



Graphon Neural Networks

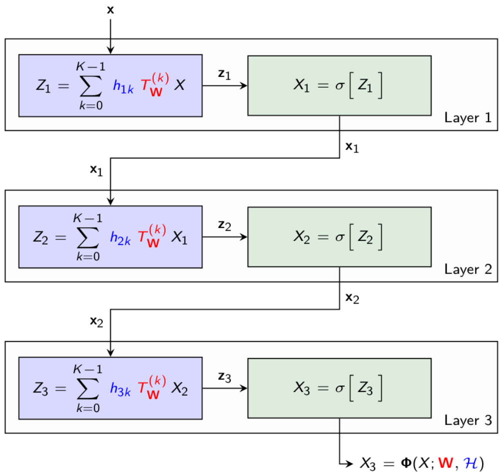
- Una NN en el grafón es una cascada de capas
- Cada cual una composición de

  $\Rightarrow$ convolución en el grafón $T_{\mathbf{H}}$

  $\Rightarrow$ y no-linealidades punto-a-punto σ
- Los parámetros a aprender son el conjunto

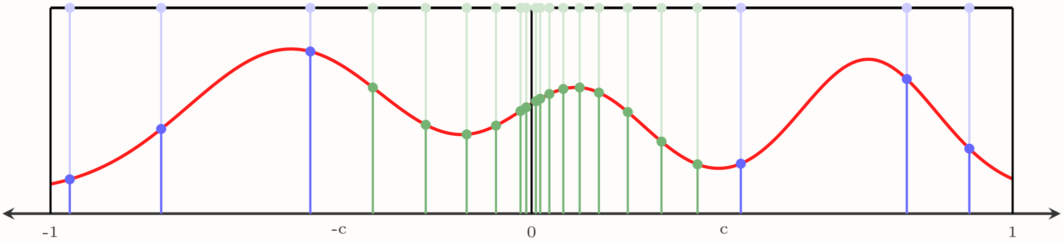
  $\mathcal{H} = \{h_{kl}\}$
- Una WNN se puede representar como

  $Y = \mathbf{\Phi}(\mathcal{H}; \mathbf{W}; X)$



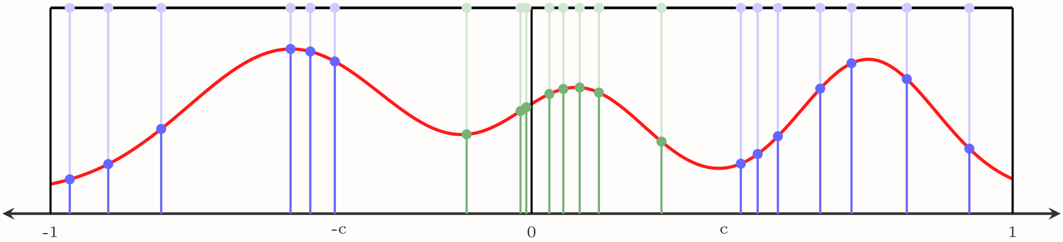
Graph Filters vs. Graph Neural Networks

- Las propiedades de transferabilidad de los filtros se heredan en la GNN
- La diferencia en la GNN es que las no-linealidades mueven las componentes espectrales por el espectro



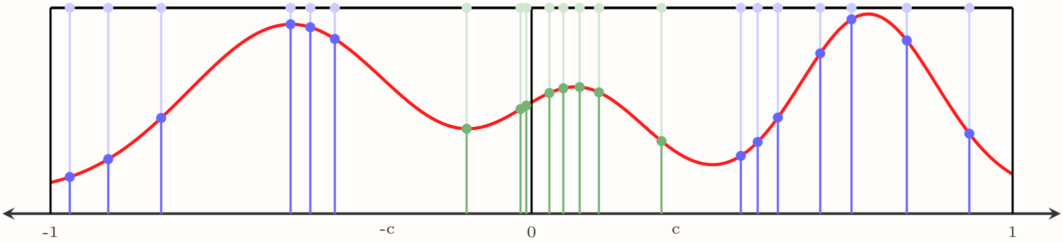
Graph Filters vs. Graph Neural Networks

- Las propiedades de transferabilidad de los filtros se heredan en la GNN
- La diferencia en la GNN es que las no-linealidades mueven las componentes espectrales por el espectro



Graph Filters vs. Graph Neural Networks

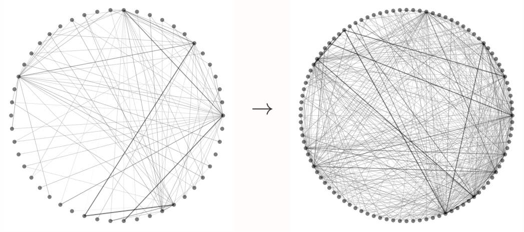
- Las propiedades de transferabilidad de los filtros se heredan en la GNN
- La diferencia en la GNN es que las no-linealidades mueven las componentes espectrales por el espectro

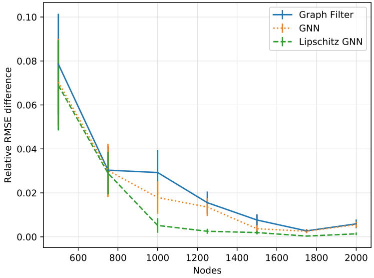


- Lo que permite aumentar la discriminabilidad sin sacrificar transferabilidad. O sea:
  - ⇒ Para el mismo nivel de transferabilidad ⇒ GNNs son más discriminativas que los filtros
  - ⇒ Para el mismo nivel de discriminabilidad ⇒ GNNs son más transferibles que los filtros

Transferabilidad en GNNs

- Evidencia empírica de la transferabilidad de las GNNs ⇒ sistema de recomendación





- La diferencia en desempeño entre entrenar usando todo el grafo y un sub-conjunto decrece con el tamaño de este último
- GNNs son más transferibles sobre todo si los filtros subyacentes son Lipschitz

Wi-Fi Indoor Positioning

- Problema: Desplegar información de la obra frente al usuario

Bracco, Grunwald, Navcevich, Capdehourat, Larroca, “Museum Accessibility Through Wi-Fi Indoor Positioning”, arXiv:2008.11340 [cs.CY], 2020

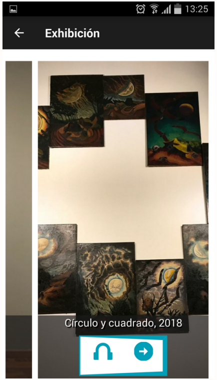
Wi-Fi Indoor Positioning

- Problema: Desplegar información de la obra frente al usuario
- ¿El usuario está frente a

Bracco, Grunwald, Navcevich, Capdehourat, Larroca, "Museum Accessibility Through Wi-Fi Indoor Positioning", arXiv:2008.11340 [cs.CY], 2020

Wi-Fi Indoor Positioning

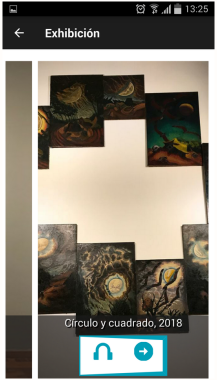
- Problema: Desplegar información de la obra frente al usuario
- ¿El usuario está frente a
  - Circulo y cuadrado o



Bracco, Grunwald, Navcevich, Capdehourat, Larroca, "Museum Accessibility Through Wi-Fi Indoor Positioning", arXiv:2008.11340 [cs.CY], 2020

Wi-Fi Indoor Positioning

- Problema: Desplegar información de la obra frente al usuario
- ¿El usuario está frente a
  - Circulo y cuadrado o
  - Juan cree que el sol es una estrella?



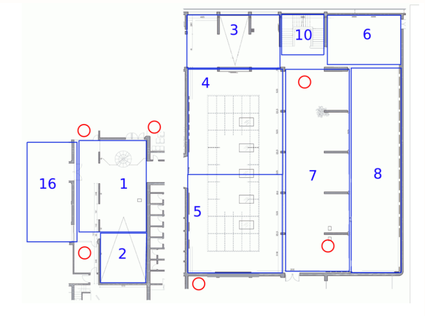
Bracco, Grunwald, Navcevich, Capdehourat, Larroca, “Museum Accessibility Through Wi-Fi Indoor Positioning”, arXiv:2008.11340 [cs.CY], 2020

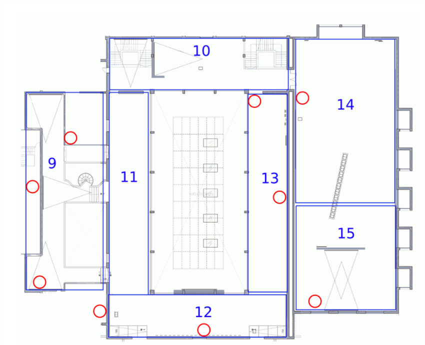
Wi-Fi Indoor Positioning

- Claramente ningún GNSS (e.g. GPS) va obtener la precisión necesaria
- Posibilidades
    - Desplegar un sistema de posicionamiento alternativo (e.g. BLE)

Wi-Fi Indoor Positioning

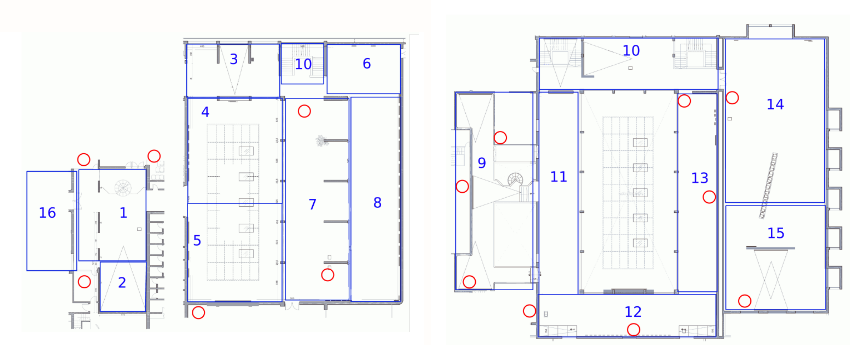
- Claramente ningún GNSS (e.g. GPS) va obtener la precisión necesaria
- Posibilidades
  - Desplegar un sistema de posicionamiento alternativo (e.g. BLE)
  - Usar señales ya disponibles (99 % de los casos: Wi-Fi). Ejemplo MNAV: 15 APs en dos pisos





Wi-Fi Indoor Positioning

- Claramente ningún GNSS (e.g. GPS) va obtener la precisión necesaria
- Posibilidades
  - Desplegar un sistema de posicionamiento alternativo (e.g. BLE)
  - Usar señales ya disponibles (99 % de los casos: Wi-Fi). Ejemplo MNAV: 15 APs en dos pisos



- ¿Son necesarias las coordenadas del usuario? No, típicamente con saber en qué zona está es suficiente

Wi-Fi Fingerprinting

- ¿Qué podemos usar del Wi-Fi disponible para inferir la posición?
  - La versión más sencilla es simplemente la potencia recibida de cada AP (RSSI)
  - Fuertemente relacionado con la distancia al AP

Wi-Fi Fingerprinting

- ¿Qué podemos usar del Wi-Fi disponible para inferir la posición?
  - La versión más sencilla es simplemente la potencia recibida de cada AP (RSSI)
  - Fuertemente relacionado con la distancia al AP
- Convertimos el problema de posicionamiento en uno de calificación:
  - Sean n_{AP} el número de APs y n_Z el número de zonas
  - Sea $\mathbf{X} \in \mathbb{R}^{n_{AP} \times F_{\text{in}}}$ la medida de RSSI del móvil en cierta zona ($F_{\text{in}} = 2$ si doble banda)
    - La fila i corresponde a la medida del AP i-ésimo
  - El objetivo es encontrar una función $\mathbf{\Phi}$ que mapee de las medidas a las zonas. O sea:

$$\mathbf{\Phi} : \mathbb{R}^{n_{AP} \times F_{in}} \to \mathbb{R}^{n_Z}$$

Wi-Fi Fingerprinting

- ¿Qué podemos usar del Wi-Fi disponible para inferir la posición?
  - La versión más sencilla es simplemente la potencia recibida de cada AP (RSSI)
  - Fuertemente relacionado con la distancia al AP
- Convertimos el problema de posicionamiento en uno de calificación:
  - Sean n_{AP} el número de APs y n_Z el número de zonas
  - Sea $\mathbf{X} \in \mathbb{R}^{n_{AP} \times F_{\text{in}}}$ la medida de RSSI del móvil en cierta zona ($F_{\text{in}} = 2$ si doble banda)
    - La fila i corresponde a la medida del AP i-ésimo
  - El objetivo es encontrar una función $\mathbf{\Phi}$ que mapee de las medidas a las zonas. O sea:

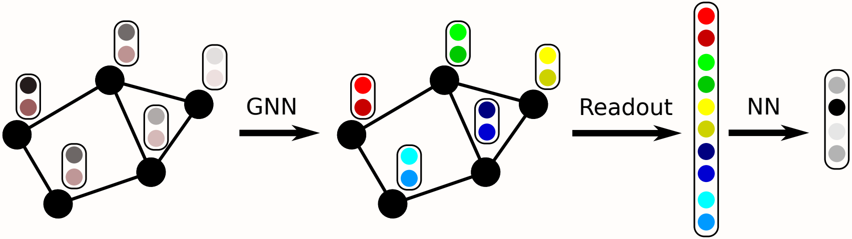
$$\mathbf{\Phi} : \mathbb{R}^{n_{AP} \times F_{in}} \to \mathbb{R}^{n_Z}$$

- Las arquitecturas usadas en este aprendizaje (e.g. KNN, FCNN) no consideran la geometría inherente al problema

Zafari, Gkelias, Leung, "A Survey of Indoor Localization Systems and Technologies", IEEE Communications Surveys & Tutorials, 2019

Wi-Fi Indoor Localization basado en GNNs

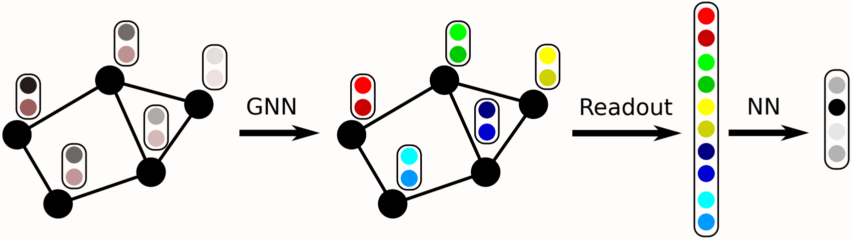
- Primera posibilidad: convertir el problema en uno de clasificación de grafos



Lezama, García-González, Larroca, Capdehourat, "Indoor Localization using Graph Neural Networks", URUCON 2021

Wi-Fi Indoor Localization basado en GNNs

- Primera posibilidad: convertir el problema en uno de clasificación de grafos



- Desventaja: pierdo la permutation equivariance
    - ¿Cómo recuperarla?

Lezama, García-González, Larroca, Capdehourat, "Indoor Localization using Graph Neural Networks", URUCON 2021

Wi-Fi Indoor Localization basado en GNNs

- Segunda posibilidad: agregar las zonas al grafo

Wi-Fi Indoor Localization basado en GNNs

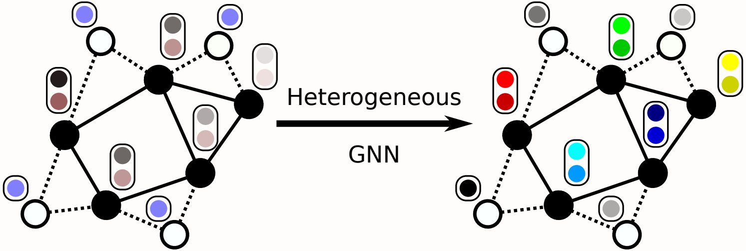
- Segunda posibilidad: agregar las zonas al grafo ¿Señal de entrada en las zonas? ¿Señal de salida en los APs?

Wi-Fi Indoor Localization basado en GNNs

- Segunda posibilidad: agregar las zonas al grafo ¿Señal de entrada en las zonas? ¿Señal de salida en los APs? ¿Aristas entre APs es lo mismo que entre zona y AP?

Wi-Fi Indoor Localization basado en GNNs

- Segunda posibilidad: agregar las zonas al grafo ¿Señal de entrada en las zonas? ¿Señal de salida en los APs? ¿Aristas entre APs es lo mismo que entre zona y AP?



- Heterogeneous GNN:

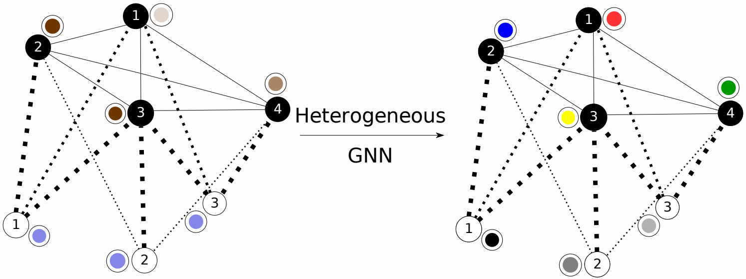
$$\mathbf{x}'_i = \sigma\left(\mathbf{H}_0\mathbf{x}_i + \sum_{r\in\mathcal{R}} \mathbf{H}_r \sum_{j\in\mathcal{N}_i^r} S_{j,i}^r \mathbf{x}_j\right),$$

Schlichtkrull, Kipf, Bloem, van den Berg, Titov, Welling, "Modeling relational data with graph convolutional networks", ESWC 2018

Lezama, Larroca, Capdehourat, "On the application of Graph Neural Networks for Indoor Positioning Systems", to appear

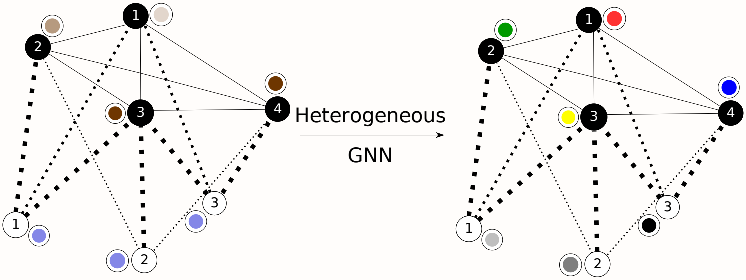
Permutation Equivariance para la HGNN

- Ejemplo:



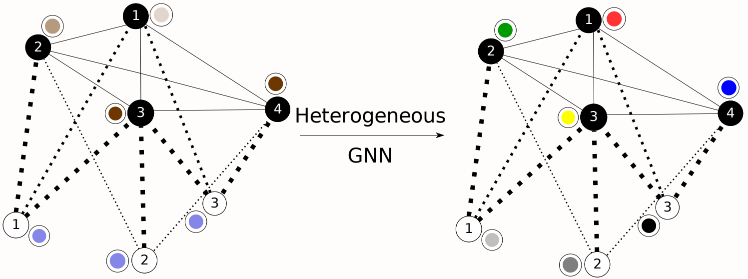
Permutation Equivariance para la HGNN

- Ejemplo:



Permutation Equivariance para la HGNN

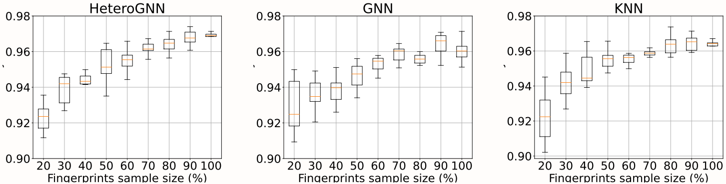
- Ejemplo:



- En la medida que haya (quasi-)simetrías internas, la GNN heterogénea debería funcionar mejor

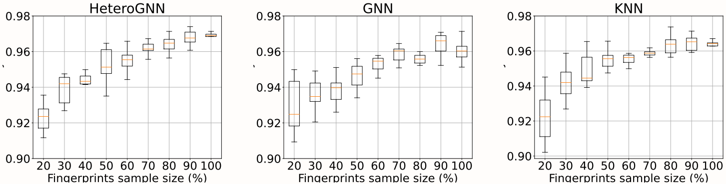
Algunos Resultados

- El mayor costo de estos sistemas es la recolección de datos
  - ¿Desempeño del sistema con pocas muestras? Resultados similares para FCNN, KNN y otros métodos shallow



Algunos Resultados

- El mayor costo de estos sistemas es la recolección de datos
  - ¿Desempeño del sistema con pocas muestras? Resultados similares para FCNN, KNN y otros métodos shallow



- ¿Y si se cae un AP?
  - Rule-of-thumb: al menos 90 % de accuracy para que el sistema "funcione"