IIC-3641 Aprendizaje Automático basado en Grafos

https://github.com/marcelomendoza/IIC3641

- RECAPITULACIÓN -

Graph Isomorphic Networks (GIN)

En la AF9, trabajamos con GIN. Este modelo propone aprender *f* y φ en base al teorema de aproximación universal de MLPs[1].

En la práctica, usamos una MLP a la salida de la función de agregación y update según:

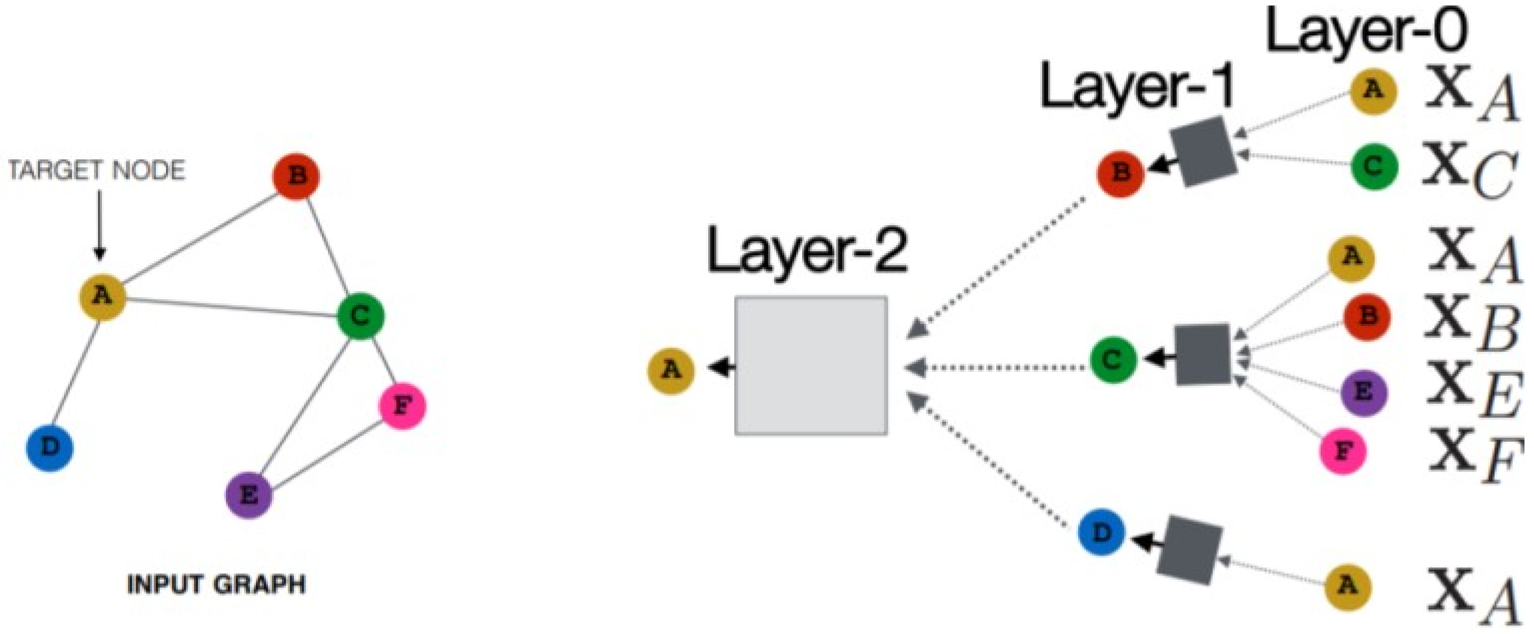
$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right)$$

Los embeddings de GIN pueden ser usados directamente para node classification o link prediction. De hecho, los usamos para clasificación de grafos en el dataset de IMDB, el cual era de clasificación binaria.

[1] Hornik, K., Stinchcombe, M., White, H. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359-366, 1989.

Graph Isomorphic Networks (GIN)

Usar una MLP a la salida de la función de update y agregación tiene este efecto:



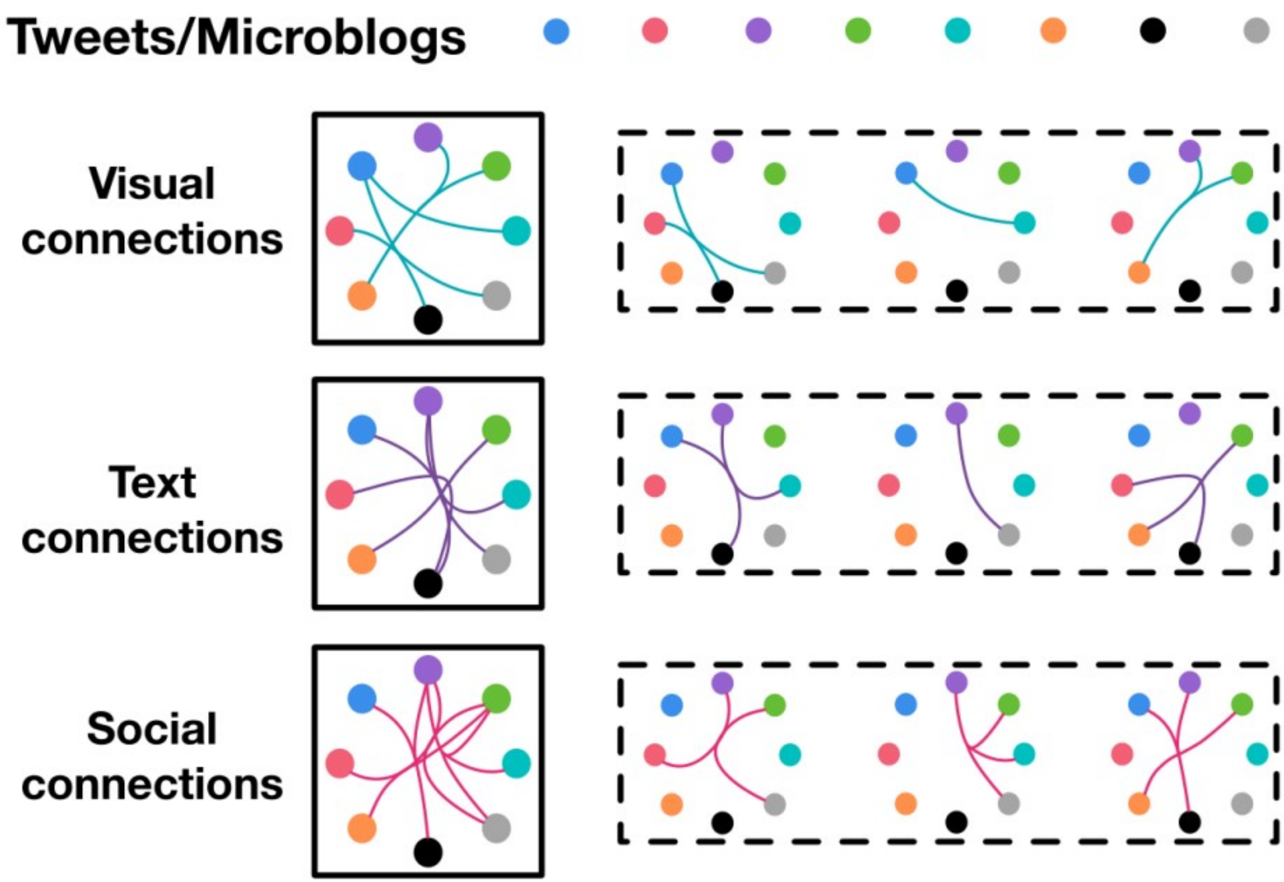
Importante notar que como la MLP es dependiente de cada capa ($MLP^{(k)}$), en la misma capa los pesos se comparten, pero estos difieren de los que se usarán en la siguiente capa. Es decir, usar más capas dota de mayor expresividad a la red, como mostró el paper del modelo GIN.

- Hypergraph Neural Networks (HGNN) -

Hipergrafos

Un hipergrafo es un grafo en el cual los enlaces pueden conectar simultáneamente mas de dos nodos.

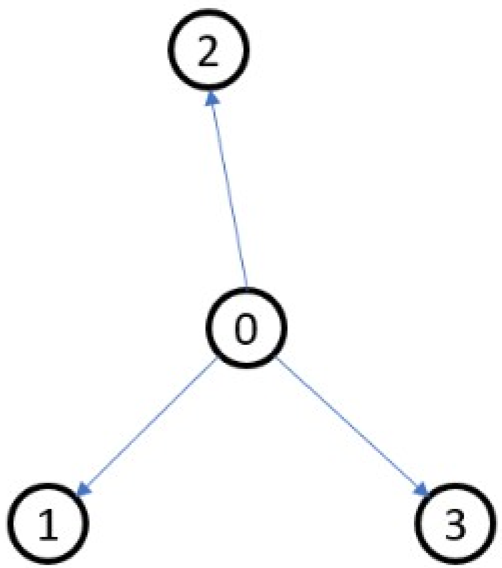
El hecho de que un hipergrafo permite enlaces de orden mayor a dos establece una representación a partir de la cual podemos identificar se puede codificar correlaciones de orden superior, algo muy frecuente en datos multimodal.Veamos un ejemplo:



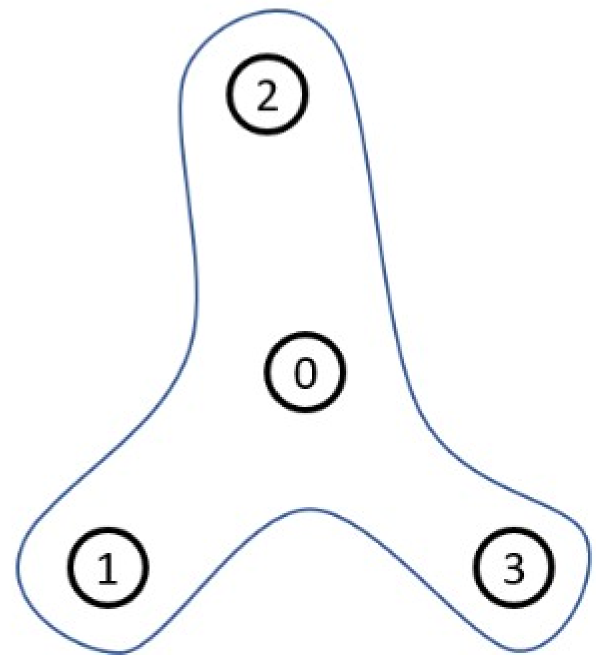
Una red social define distintos tipos de interacciones, y en realidad, tenemos redes superpuestas.

Hipergrafos

Una forma de modelar lo anterior es usando hiperenlaces, es decir, definir, enlaces que conecten varios nodos a la vez.



Enlaces diádicos



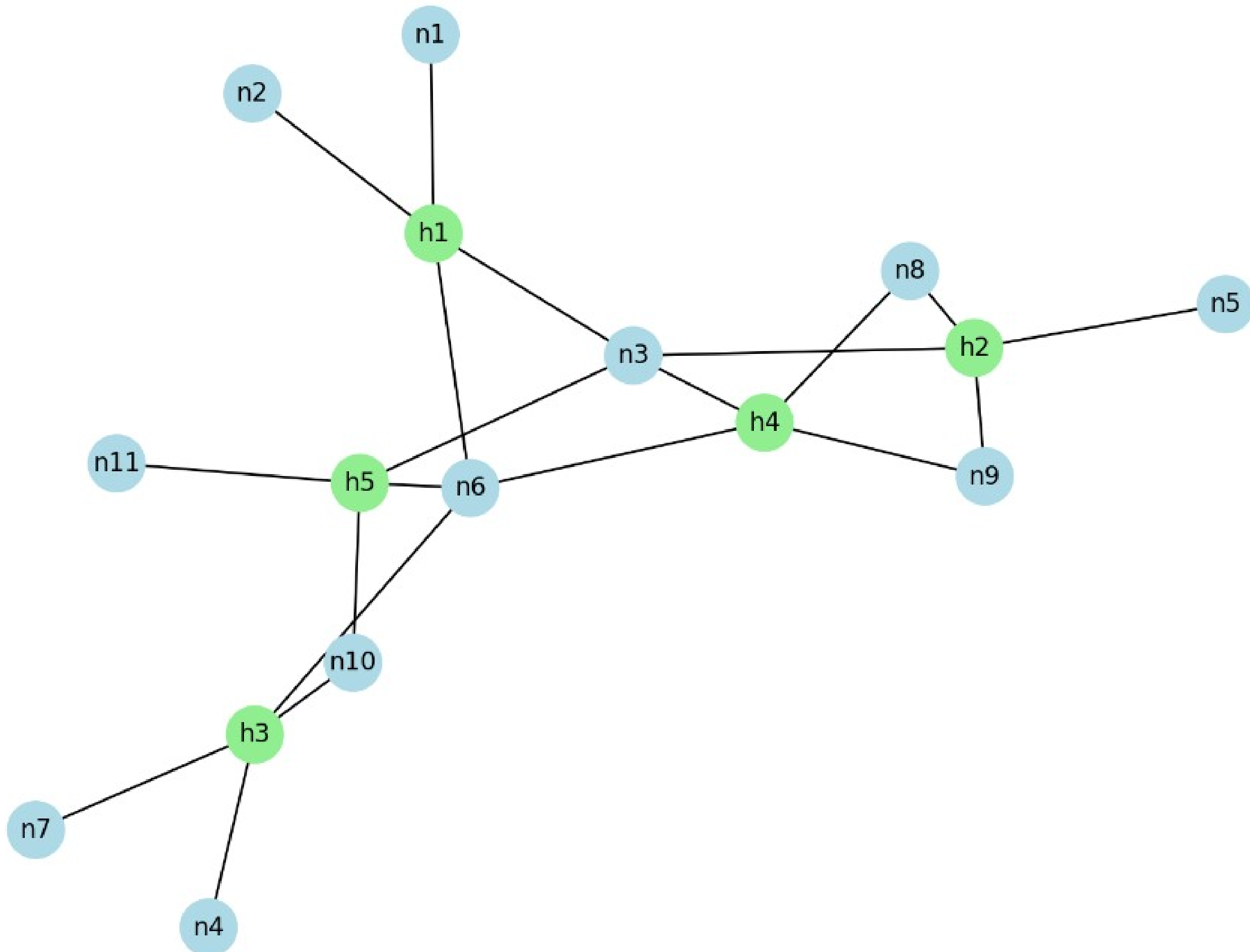
Hiperenlace

Una forma sencilla de reprsentar este tipo de grafos es a través de una **matriz de incidencia**, esto es, cada fila representa un nodo y cada columna un hiperenlace. Cada columna puede tener varios 1s, es decir, puede vincular a varios nodos a las vez.

Hipergrafos

Matriz de incidencia

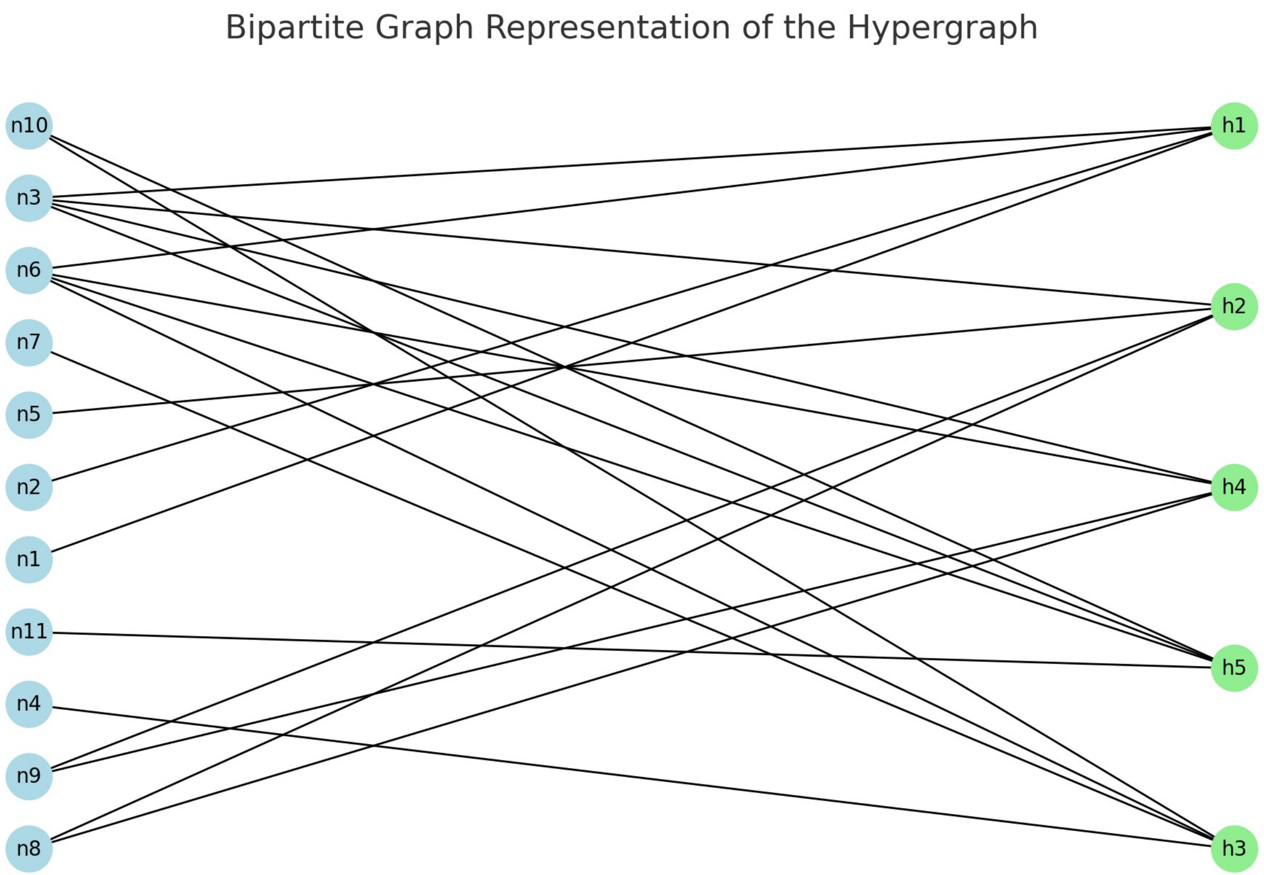
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Algo interesante de esta representación es que permite definir hasta cierto punto grafos bipartitos, ya que surgen dos tipos de nodos y los nodos del mismo tipo no tienen conexiones directas entre sí.

Hipergrafos

Un hipergrafo puede verse como una grafo bipartito de nodos e hiperenlaces.



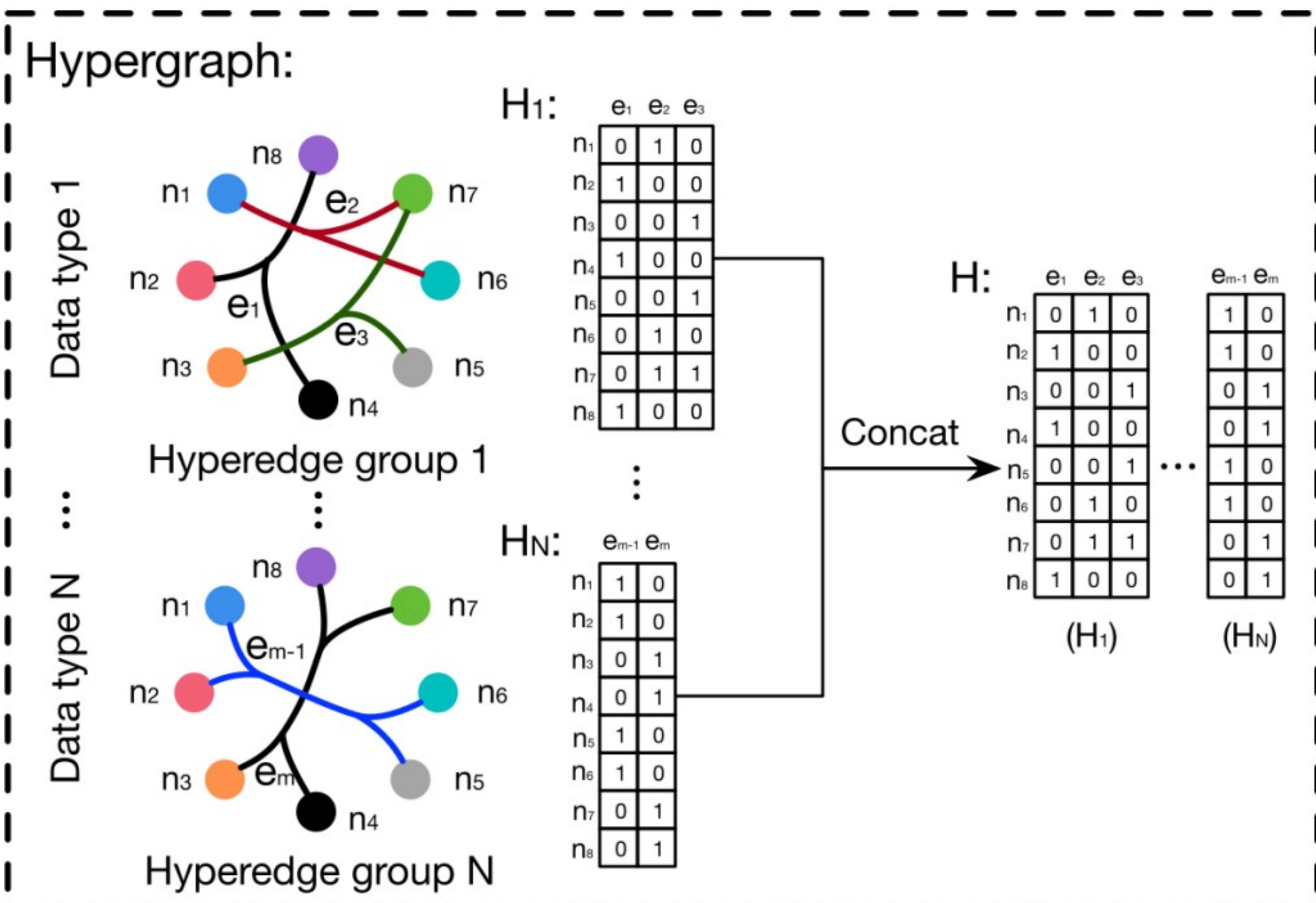
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Esta representación es bastante útil ya que se conecta con sistema recomendadores, en los cuales el grafo bipartito modela usuarios y productos.

Hypergraph Neural Networks

Vamos a estudiar un modelo que permite representar N hipergrafos que operan sobre los mismos nodos, esto, para permitir modelar datos multimodales. Esto es interesnate ya que es lo suficientemente flexible como para modelar distintos tipos de datos en una misma prerepsentación.

En el caso de un hipergrafo estándar, simplemente usando este modelo de forma unimodal. Para hacer esto, las HGNN operan sobre una concatenación de matrices de incidencia:



Hypergraph Neural Networks

Las HGNN definen una matriz de pesos para los hiperenlaces. Cada hiperenlace tiene una matriz diagonal de pesos que modela la relevancia de ese hiperenlace para el grafo (peso global de e en G). En base a esto, se definen las funciones de grado de nodos y de hiperenlaces:

$$d(v) = \sum_{e \in \mathcal{E}} \omega(e) h(v, e)$$

$$\delta(e) = \sum_{v \in \mathcal{V}} h(v, e)$$

Los h( ) representa las entradas de la matriz de incidencia.

El aprendizaje de representaciones en una HGNN se hace a niveld e nodos. Típicamente la tarea considera una fase de minimzación de un funcional con dos componentes:

$$\arg \min_{f} \{\mathcal{R}_{emp}(f) + \Omega(f)\}$$

donde f representa la función de clasificación (por ejemplo, clasificación de nodos), R la función de pérdida empírica supervisada (típicamente transductiva), y omega es la componente de regularización.

Hypergraph Neural Networks

El factor de regularización se minimiza si, dos nodos conectados por un hiperenlace, se corresponden según f:

$$\Omega(f) = \frac{1}{2} \sum_{e \in \mathcal{E}} \sum_{\{u,v\} \in \mathcal{V}} \frac{w(e)h(u,e)h(v,e)}{\delta(e)} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2$$

Este factor tiene una relación con el Laplaciano de un hipergrafo, el cual se define a partir de:

$$\boldsymbol{\Theta} = \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2}$$

y entonces el Laplaciano es:

$$\boldsymbol{\Delta} = \mathbf{I} - \boldsymbol{\Theta}$$

El factor de regularización normalizado se escribe según:

$$\Omega(f) = f^\top \boldsymbol{\Delta}$$

Hypergraph Neural Networks

La conexión de las HGNN con las GCN es evidente. De hecho, adolece de las mismas limitaciones. Si recordamos, la convolución espectral en base a la transformada de Fourier de grafo era muy costosa de computar. Luego, las HGNN para ser eficientes, simplifican el operador de convolución a lo siguiente:

$$\mathbf{g} \star \mathbf{x} \approx \theta_0 \mathbf{x} - \theta_1 \mathbf{D}^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_{\mathbf{e}}^{-1} \mathbf{H}^\top \mathbf{D}_{\mathbf{v}}^{-1/2} \mathbf{x}$$

donde g es el kernel de convolución, definido a partir del grafo. Sin aproximación, g debiera ser el kernel de convolución de los vectores propios del laplaciano del grafo.

Como g se aproxima, se usa lo siguiente:

$$\begin{cases} \theta_1 = -\frac{1}{2}\theta \\ \theta_0 = \frac{1}{2}\theta \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2} \end{cases}$$

Luego, en simple, la convolución corresponde a:

$$\begin{aligned} \mathbf{g} \star \mathbf{x} &\approx \frac{1}{2}\theta \mathbf{D}_v^{-1/2} \mathbf{H}(\mathbf{W} + \mathbf{I}) \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2} \mathbf{x} \\ &\approx \theta \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2} \mathbf{x} \end{aligned}$$

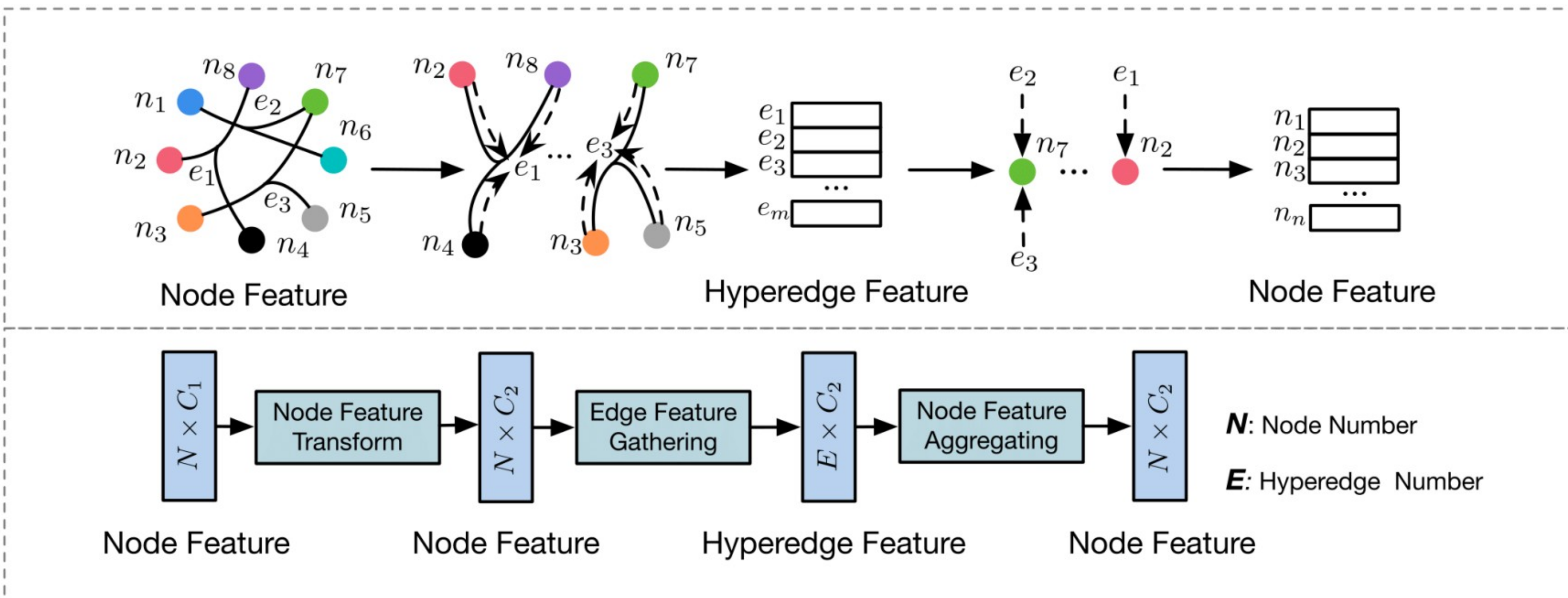
Hypergraph Neural Networks

Finalmente, la HGNN opera usando capas convolucionales sobre el hipergrafo de la siguiente forma:

$$\mathbf{X}^{(l+1)} = \sigma(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{(l)}\mathbf{\Theta}^{(l)})$$

Actúa a nivel de hiperenlace

Luego de la función de
activación, vuelve a los nodos



Hypergraph Neural Networks

Los experimentos muestran que la HGNN es similar a la GCN en grafos convencionales.

Method	Cora	Pubmed
DeepWalk (Perozzi, Al-Rfou, and Skiena 2014)	67.2%	65.3%
ICA (Lu and Getoor 2003)	75.1%	73.9%
Planetoid (Yang, Cohen, and Salakhutdinov 2016)	75.7%	77.2%
Chebyshev (Defferrard, Bresson, and Vandergheynst 2016)	81.2%	74.4%
GCN (Kipf and Welling 2017)	81.5%	79.0%
HGNN	**81.6%**	**80.1%**

Sin embargo, su utilidad está en su capacidad para modelar hipergrafos, por ejemplo en dataset multimodales, algo que la GCN no puede hacer de forma nativa.

Method	Classification Accuracy
PointNet (Qi et al. 2017a)	89.2%
PointNet++ (Qi et al. 2017b)	90.7%
PointCNN (Li et al. 2018)	91.8%
SO-Net (Li, Chen, and Lee 2018)	93.4%
HGNN	**96.7%**

👉 Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y. Hypergraph Neural Networks, AAAI 2019 https://arxiv.org/pdf/1809.09401