Redes Neuronales de Grafos para Sistemas de Recomendación

Denis Parra
Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Graph Neural Networks para Recomendación

1. Conceptos iniciales
2. Aprendizaje Inductivo y Transductivo
3. Sistemas de Recomendación Basados en GNN
4. Clases de GNN
5. GraphSAGE
6. LightGCN

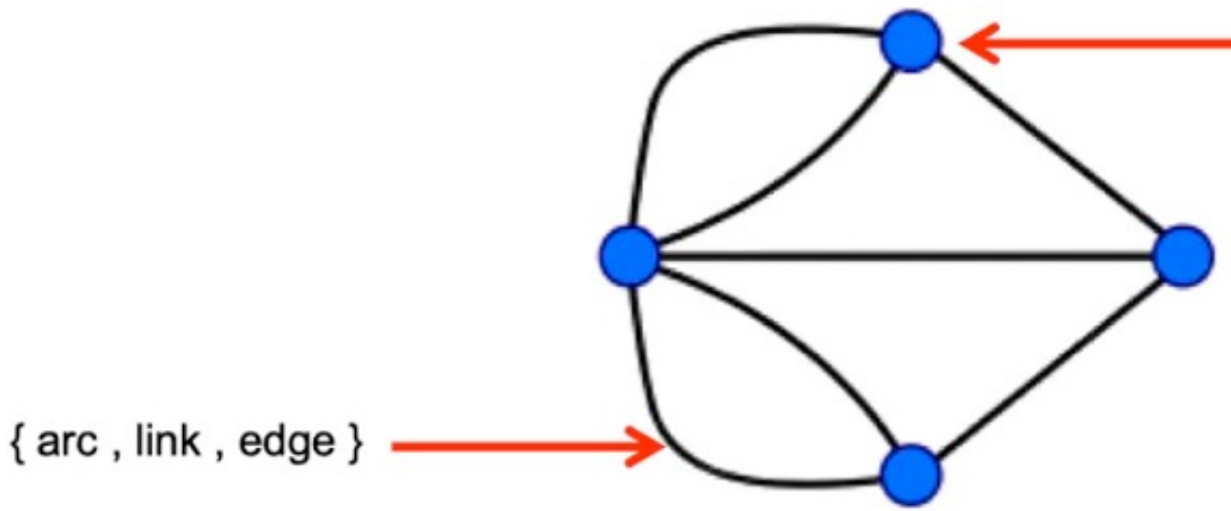
Conceptos básicos de Grafos

- Kleinberg y Easley definen **red** como:

Un concepto abstracto de "cualquier colección de objetos en los cuales alguno de los pares de objetos están conectados"

- Luego, **grafo** es una forma de representar una red. Consiste en un conjunto de objetos, llamados nodos, con ciertos pares de esos objetos conectados por líneas llamadas enlaces.

Conceptos básicos de Grafos

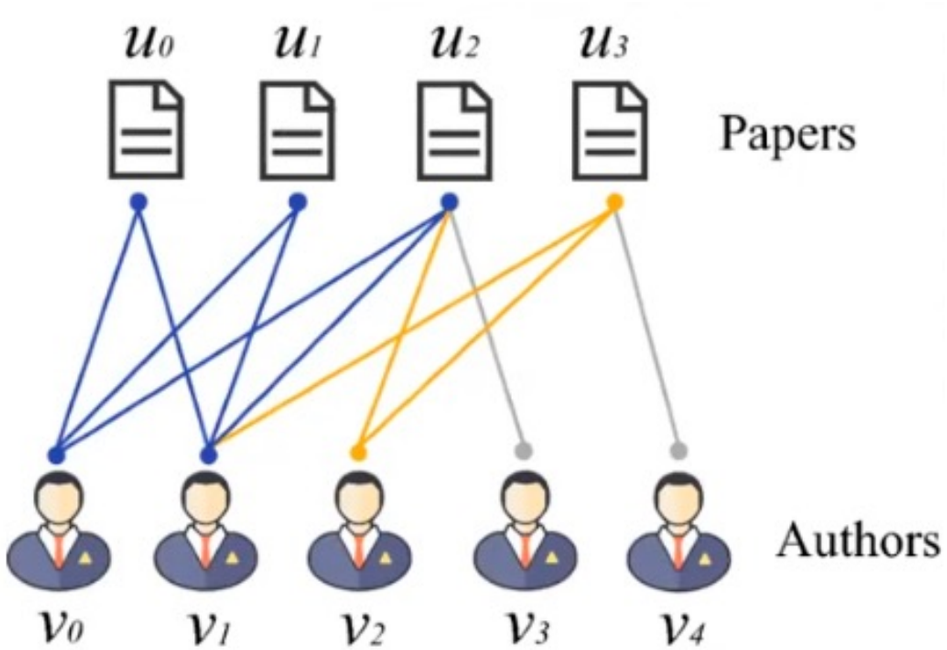


points	lines	
vertices	edges, arcs	math
nodes	links	computer science
sites	bonds	physics
actors	ties, relations	sociology

Source: L. Adamic SNA class @coursera

Grafos en Sistemas de Recomendación

- Es natural modelar el problema de recomendación como un problema de grafos
- Por ejemplo, predecir los enlaces en un grafo bipartito que aún no ha sido completado



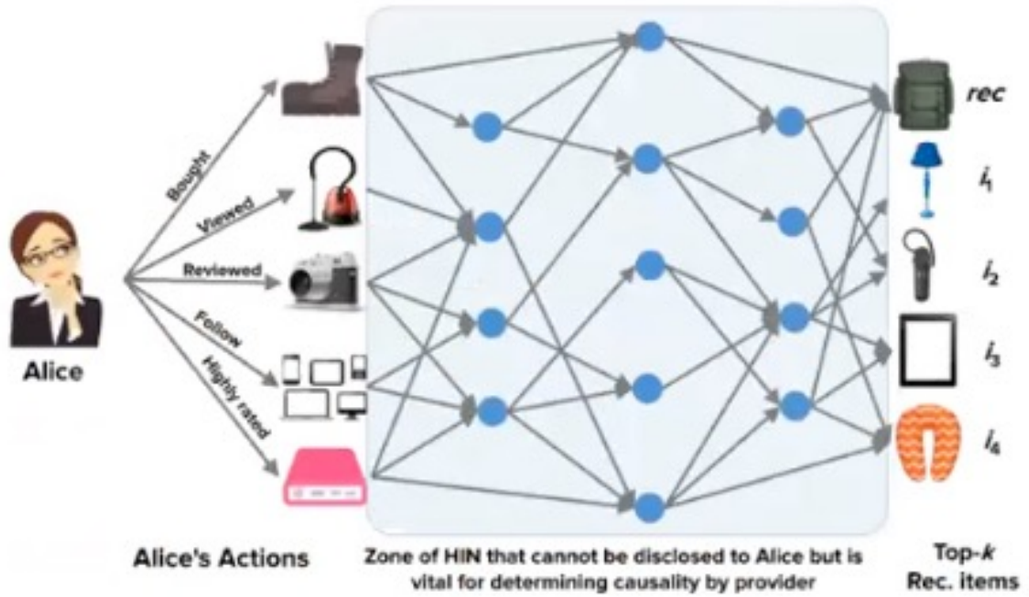
Wang, K., Lin, X., Qin, L., Zhang, W., & Zhang, Y. (2021). Towards efficient solutions of bitruss decomposition for large-scale bipartite graphs. *The VLDB Journal*, 1-24.

Antecedentes

- Existen trabajos de más de una década que muestran el potencial de la representación de grafos para modelar sistemas de recomendación
  - Jeh, G., & Widom, J. (2002). Simrank: a measure of structural-context similarity
  - Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering
  - Liben-Nowell, D., & Kleinberg, J. (2003). The link prediction problem for social networks.

Antecedentes: Explicabilidad

- O para proveer explicaciones a recomendaciones



Ghazimatin, A., Balalau, O., Saha Roy, R., & Weikum, G. (2020). PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (pp. 196-204).

Graph Neural Networks

- La aparición de las redes neuronales como forma de aprender representaciones de imágenes, texto, secuencia, videos y audio se extendió a los grafos.
- El concepto de red neuronal de grafo fue propuesto por Scarselli et al. (2008)

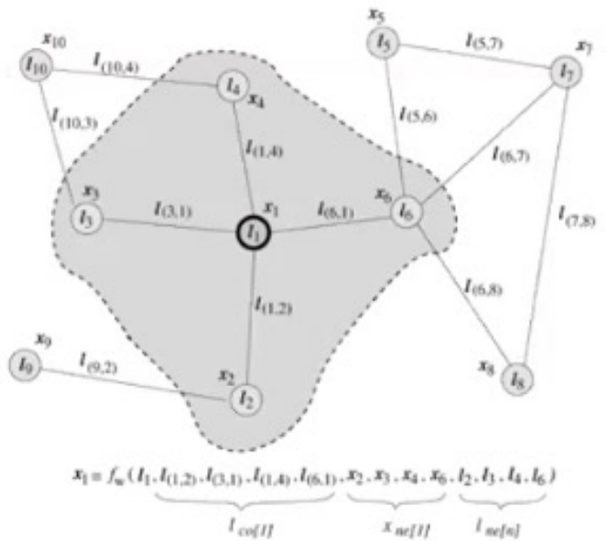
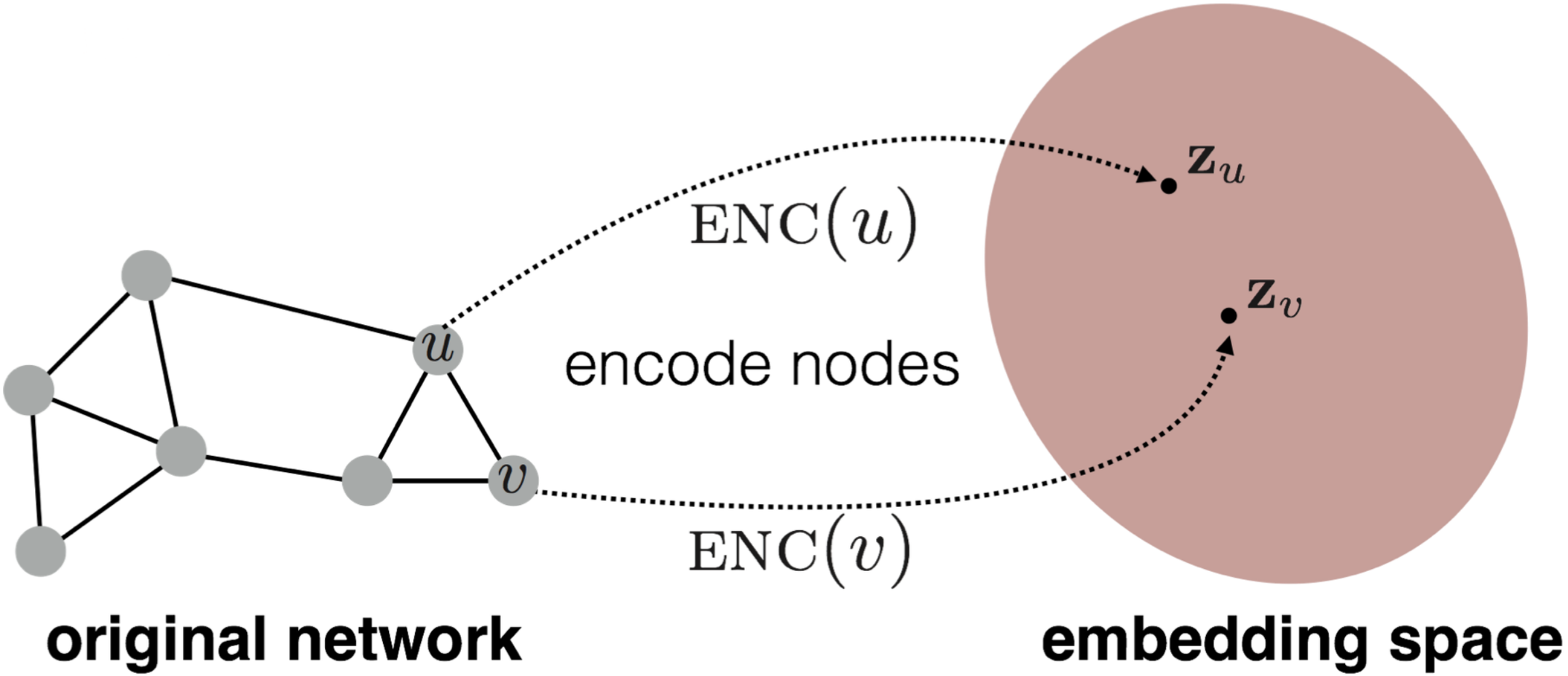


Fig. 2. Graph and the neighborhood of a node. The state x_1 of the node 1 depends on the information contained in its neighborhood.

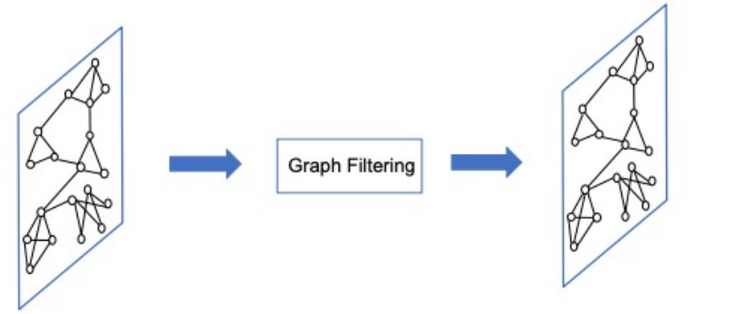
Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, *20*(1), 61-80.

GNN: Representation Learning

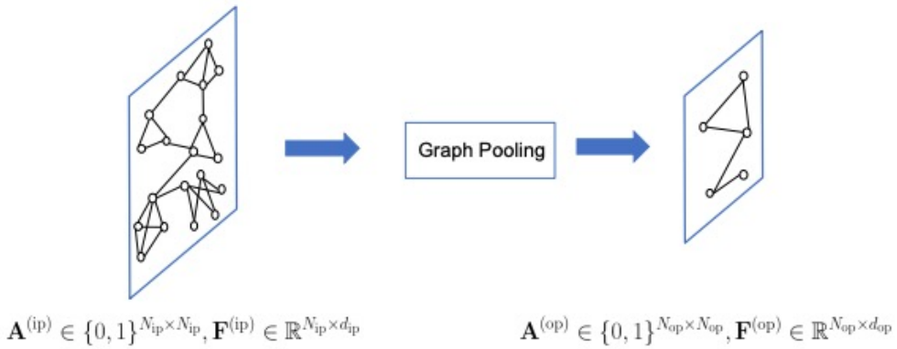


Conceptos de GNNs: filtro y selección

- **Filtro**: cambios en atributos de nodos en base a operaciones del grafo o vecinos del nodo activo

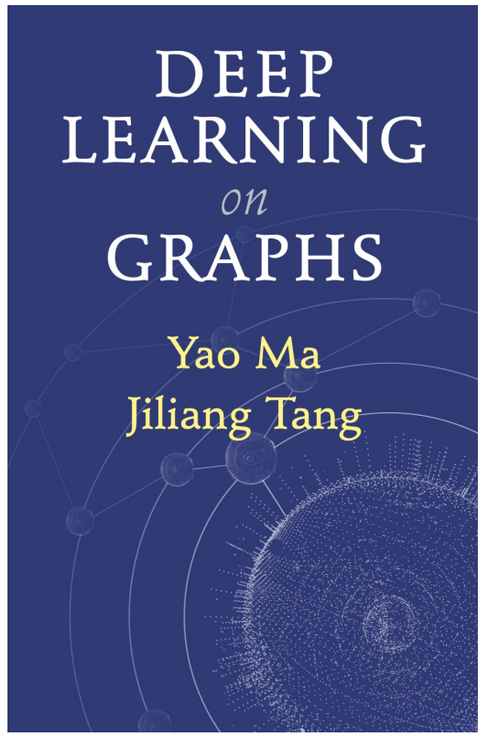


- **Selección**: cambios en el grafo producto de operaciones como pooling



Graph Neural Networks: Representation Learning

- Deep Learning on Graphs (Ma & Tang, 2021)
- https://web.njit.edu/~ym329/dlg_book/



PART 1 Foundations

- Chapter 2 Foundations of Graphs [Introduction]
- Chapter 3 Foundations of Deep Learning [Introduction]

PART 2 Methods

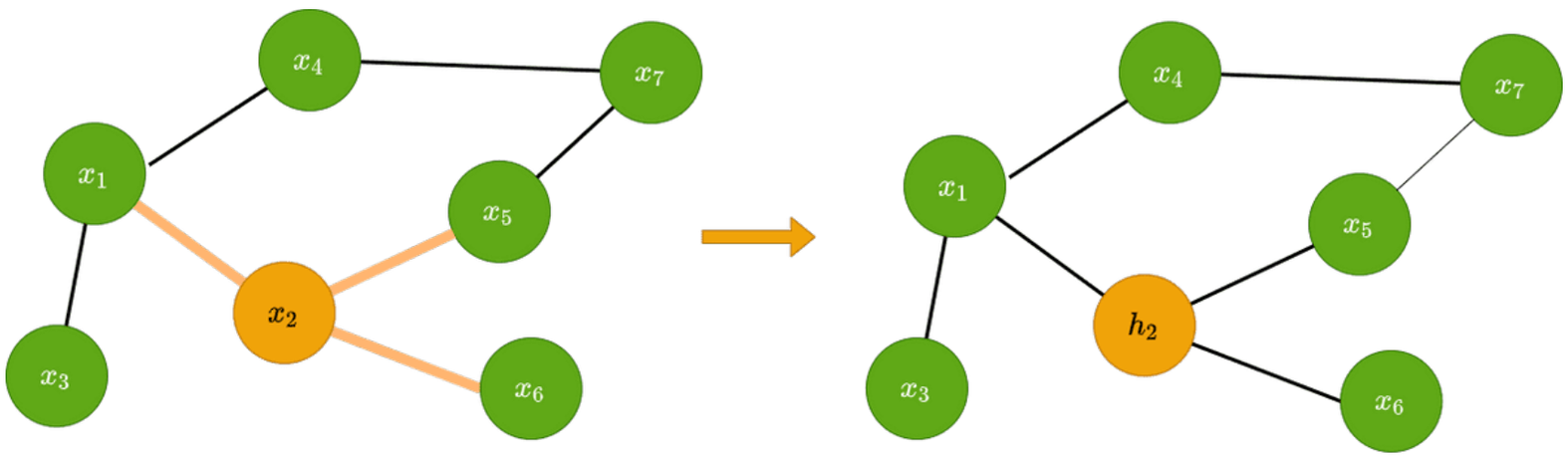
- Chapter 4 Graph Embedding [Introduction]
- Chapter 5 Graph Neural Networks [Introduction]
- Chapter 6 Robust Graph Neural Networks [Introduction]
- Chapter 7 Scalable Graph Neural Networks [Introduction]
- Chapter 8 Graph Neural Networks on Complex Graphs [Introduction]
- Chapter 9 Beyond GNNs: More Deep Models on Graphs [Introduction]

PART 3 Applications

- Chapter 10 Graph Neural Networks in Natural Language Processing [Introduction]
- Chapter 11 Graph Neural Networks in Computer Vision [Introduction]
- Chapter 12 Graph Neural Networks in Data Mining [Introduction]
- Chapter 13 Graph Neural Networks in Biochemistry and Healthcare [Introduction]

Convolución en Grafos

- Convoluciones sobre un grafo: permiten obtener features o representación de un nodo a partir de las características de los vecinos.



$$h_2 = g(x_1, x_5, x_6)$$

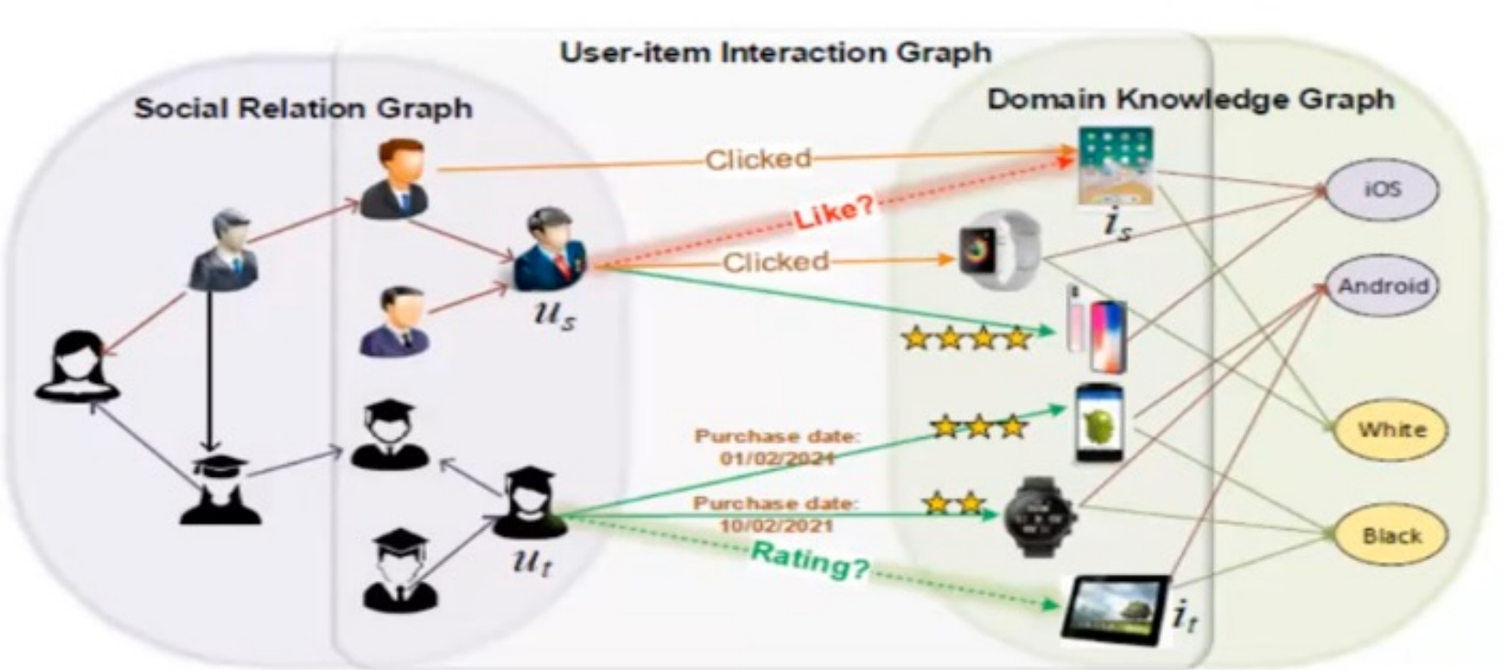
Aprendizaje Inductivo vs. Transductivo

- En ***aprendizaje transductivo***, el modelo ya ha visto tanto datos de entrenamiento como de test. Costo: si se agrega un nodo nuevo el gráfico, se debe volver a entrenar con el modelo.

- En ***aprendizaje inductivo***, el modelo sólo ve los datos de entrenamiento. Por lo tanto, el modelo se utilizará para predecir etiquetas para datos no vistos en el grafo.

Sistemas de Recomendación basados en GNN

- Facilidad para representar interacciones y conexiones de elementos mediante grafos
- Capacidad de aprender relaciones complejas



Clases de GNN

- Con las GNN se utilizan métodos de redes neuronales sobre información en grafos para generar recomendaciones:
  - **Graph Convolutional Neural Networks**: utilizar pooling y convoluciones para considerar la información de los nodos vecinos.
  - **Gated Graph Neural Networks**: Se introduce una GRU para aprender representaciones de nodos absorviendo iterativamente la influencia de otros nodos.
  - **Attention Network (**GAT**)**: Se utilizan mecanismos de atención para aprender las distintas relevancias de tienen algunos usuarios (o ítems) sobe otros.

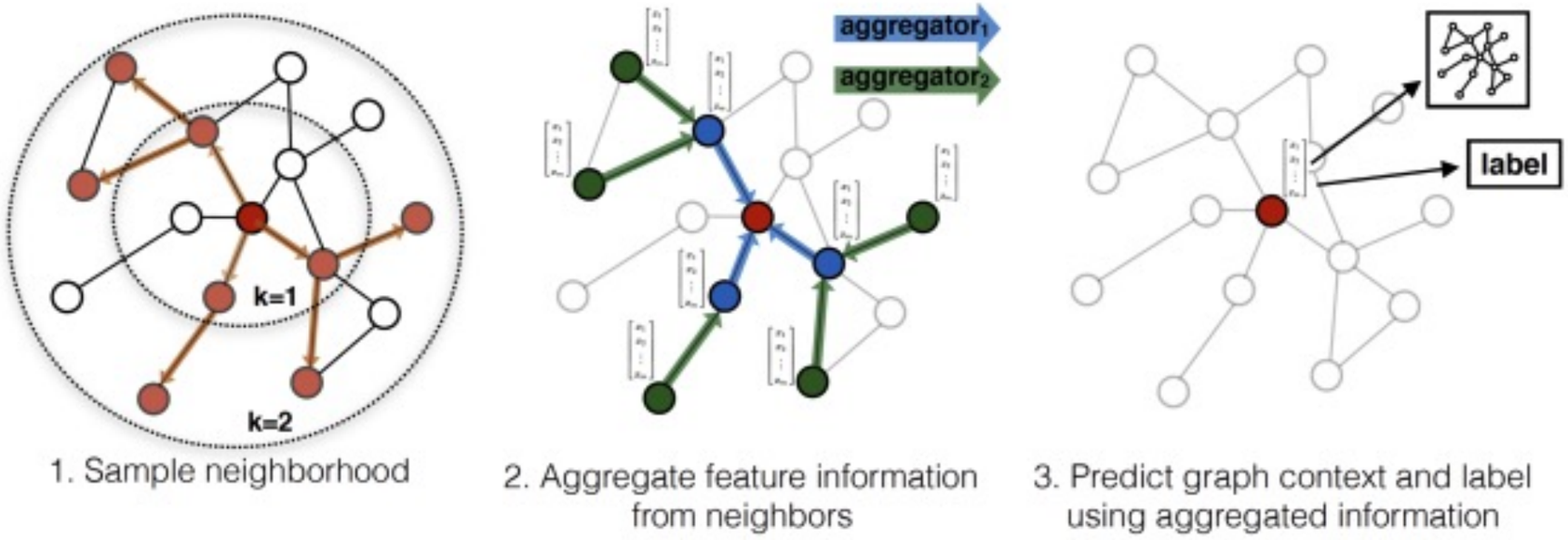
GraphSAGE

- Método inductivo (puede generalizar a nodos no observados)
- "**Inductive Representation Learning on Large Graphs**", https://arxiv.org/abs/1706.02216
- Inventado por Hamilton, Ying y Leskovec (2017) para generar representaciones de nodos de grafos.
- Fue adaptado como PinSAGE para hacer recomendaciones en Pinterest.

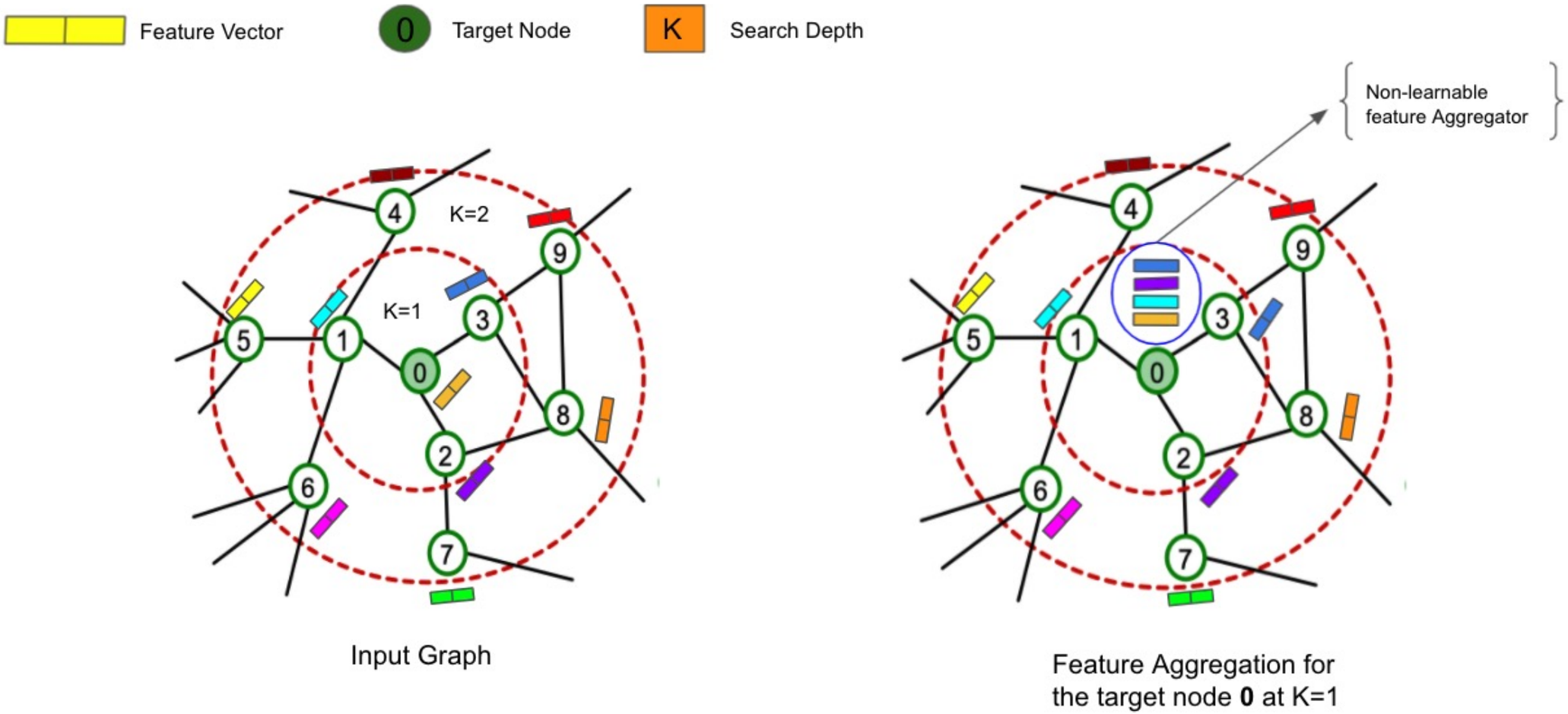
Principios de GraphSAGE

- PASO 1: Muestrear vecindario de los nodos

- PASO 2: Aprender funciones de agregación para cada profundidad de búsqueda:

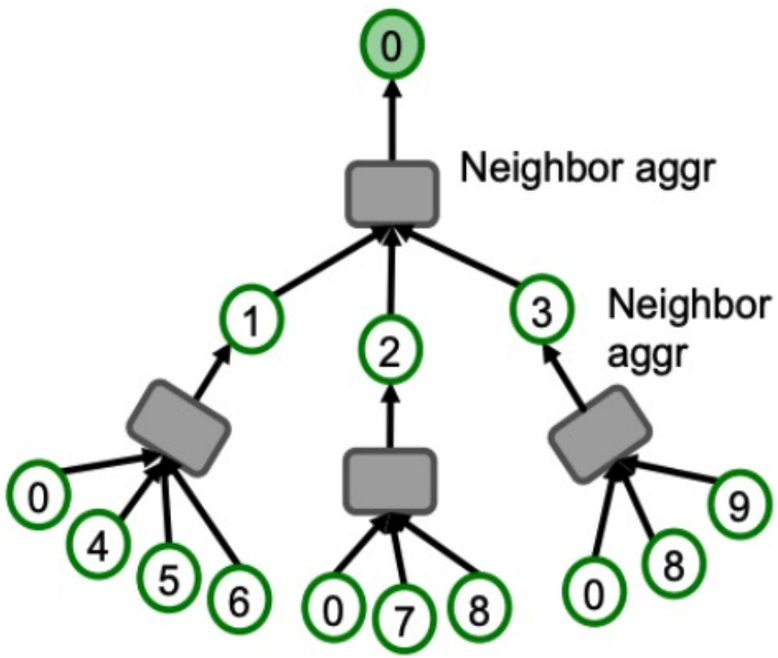


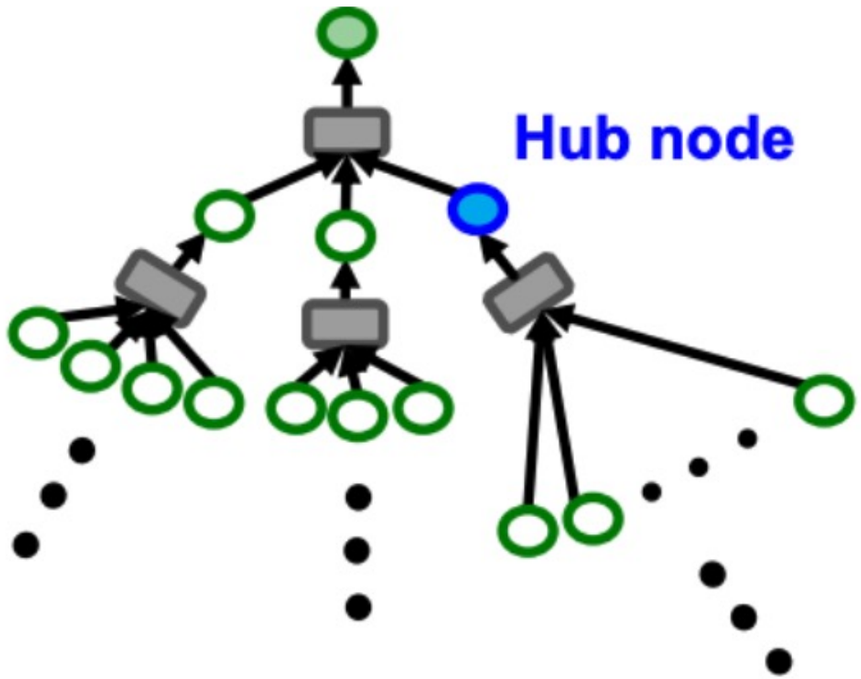
Muestreo de nodos en GraphSAGE



Muestreo de nodos en GraphSAGE II

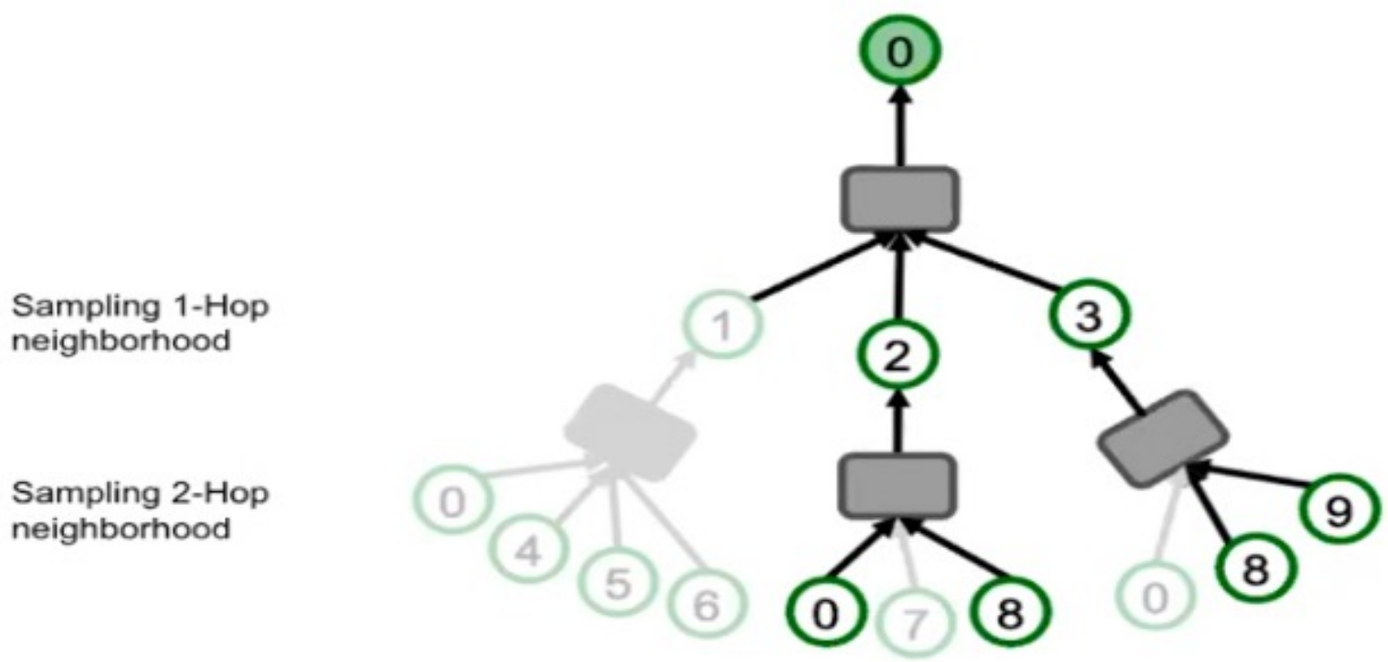
- El muestreo en GraphSAGE permite escalar el aprendizaje de embeddings para los nodos a millones, pero:
  - Para un K grande, crece exponencialmente el número de cálculos
  - Podemos encontrarnos con la maldición de “nodos celebridades”





Muestreo de nodos en GraphSAGE III

- Muestreo en GraphSAGE permite escalar aprendizaje de embeddings para los nodos a millones o miles de millones

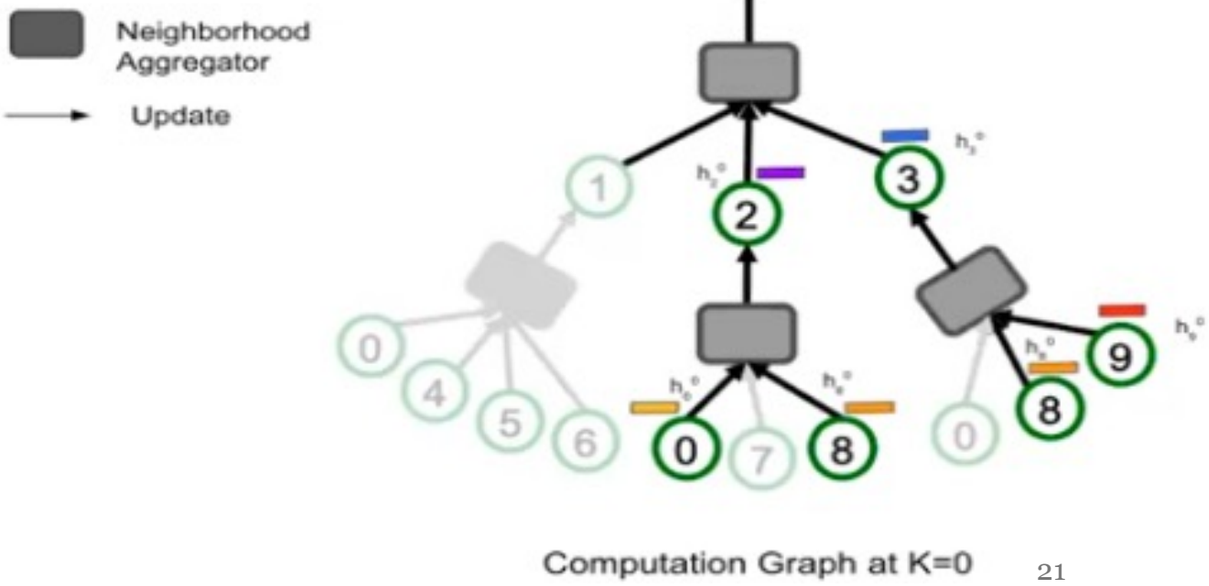


Aprendizaje en GraphSAGE: Init y Concat

- Se crean nodos agregadores
- Los nodos se inicializan con características originales
- El proceso iterativo se llama message passing

- Al finalizar K=0 tendremos

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot \text{MEAN}(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$$



Computation Graph at K=0

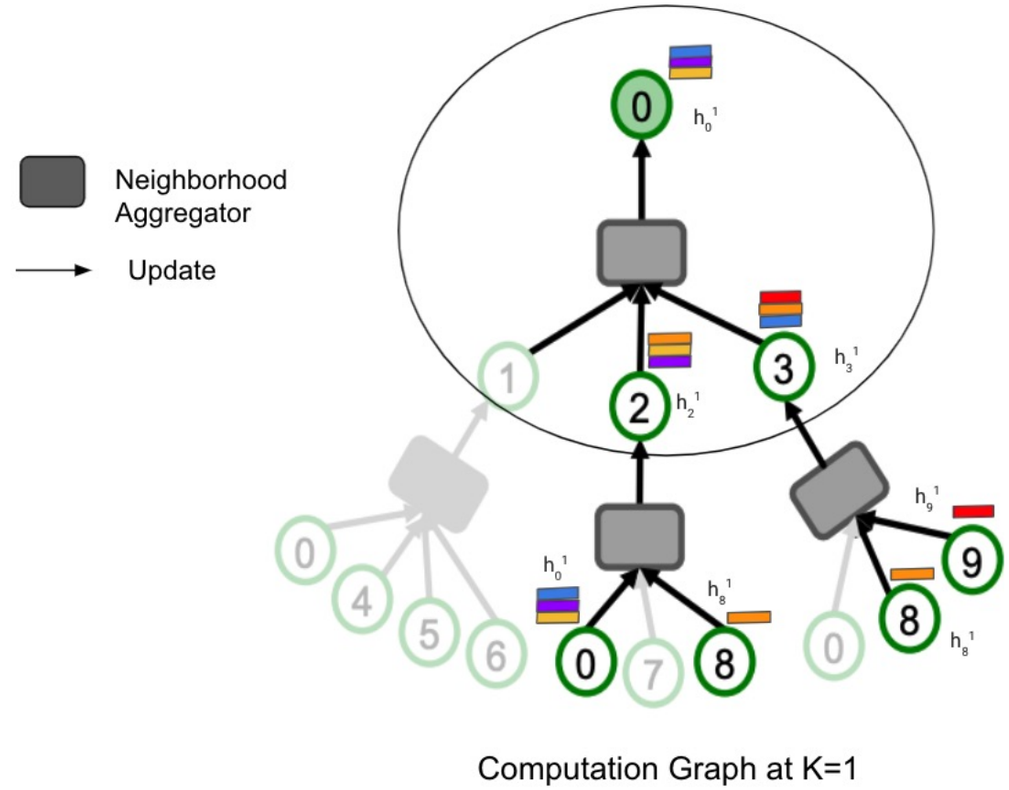
Aprendizaje en GraphSAGE: Update

- Una vez que tiene representaciones a cada nivel K, actualiza la representación de los nodos con la ecuación

$$\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$$

- Adicionalmente, se normaliza para mantener la distribución de node embeddings

$$\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$$



Computation Graph at K=1

Pseudocódigo de GraphSAGE

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k, \forall k \in \{1, ..., K\}$; non-linearity σ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
for $k = 1...K$ **do**
  for $v \in \mathcal{V}$ **do**
    $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
    $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
  end
  $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

Pérdida: Supervisada y No Supervisada

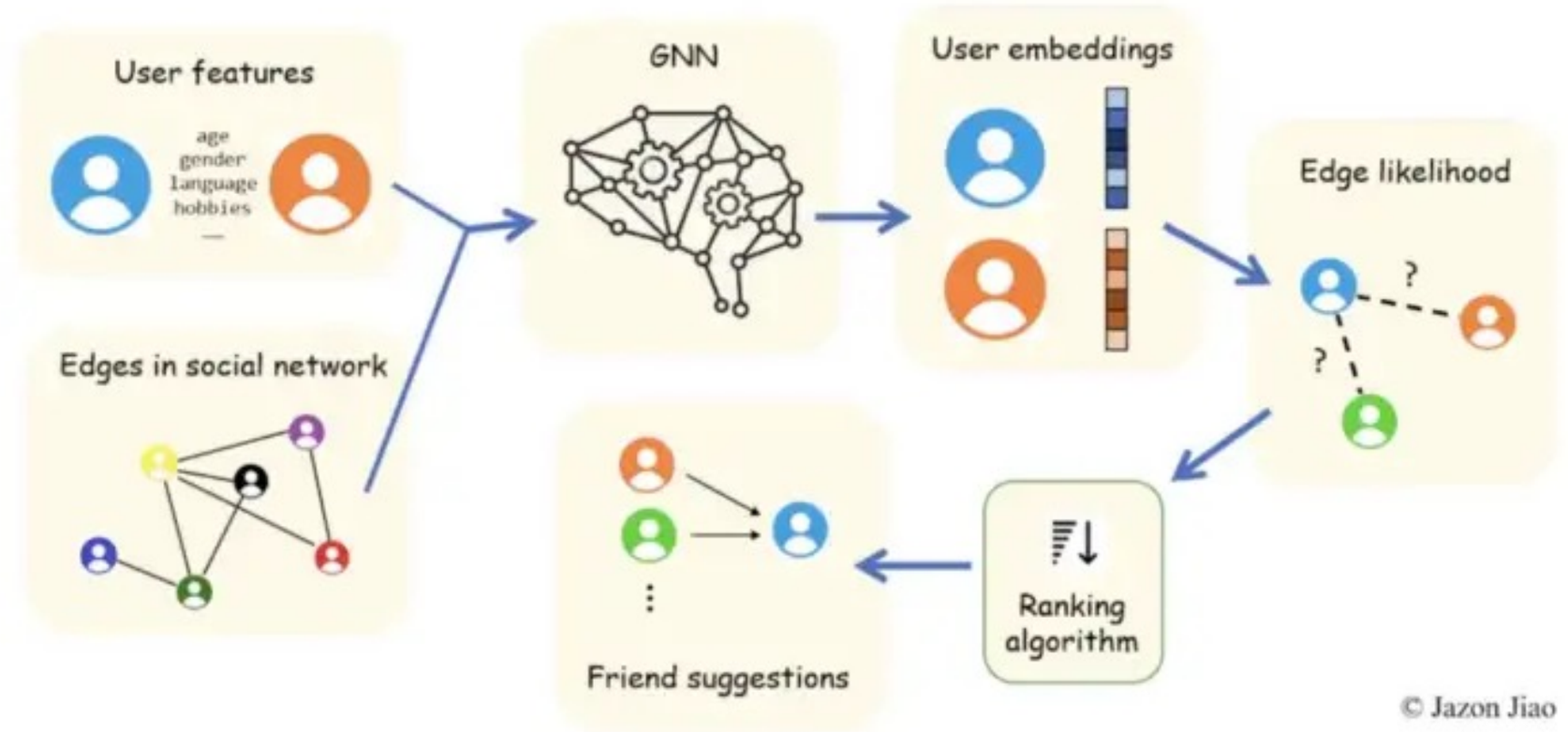
- **No supervisada**: Similar a BPR, nodos conectados deben quedar más cerca entre ellos que nodos no conectados (e.g. Hinge loos)

- **Supervisada**: Si hay etiquetas de los nodos, se puede supervisar el aprendizaje vía clasificación:

$$J_{\mathcal{G}}(\mathbf{z}_u) = -\log\left(\sigma(\mathbf{z}_u^\top \mathbf{z}_v)\right) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log\left(\sigma(-\mathbf{z}_u^\top \mathbf{z}_{v_n})\right)$$

Aplicación

- **Recomendación de contactos en red social**

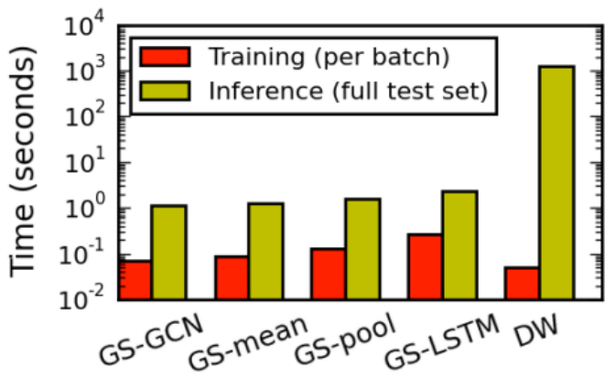


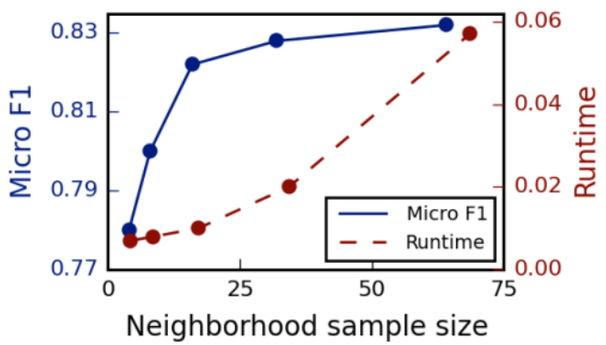
https://medium.com/stanford-cs224w/friend-recommendation-using-graphsage-ffcda2aaf8d6

Resultados GraphSAGE

Table 1: Prediction results for the three datasets (micro-averaged F1 scores). Results for unsupervised and fully supervised GraphSAGE are shown. Analogous trends hold for macro-averaged scores.

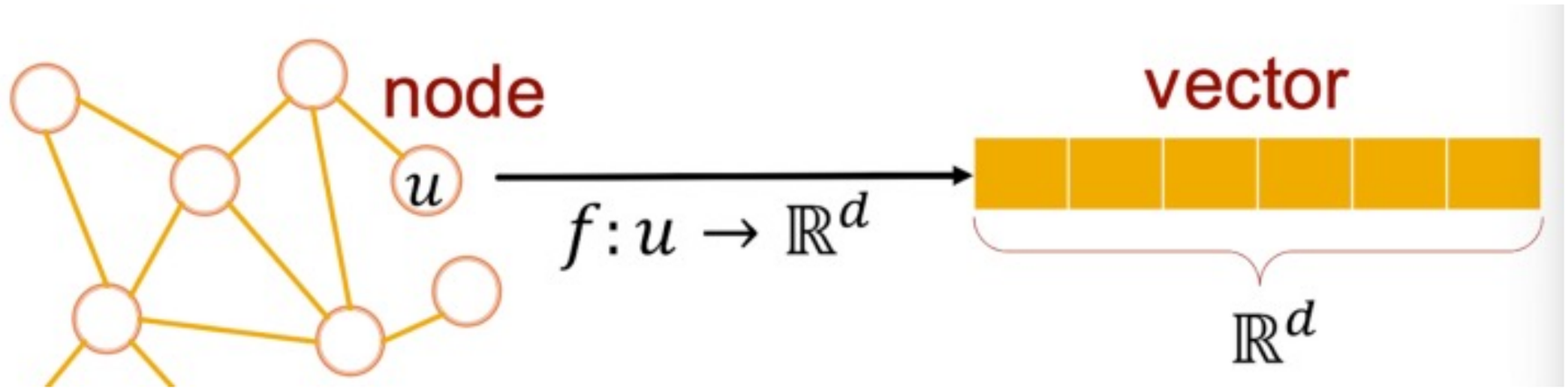
	Citation		Reddit		PPI	
Name	Unsup. F1	Sup. F1	Unsup. F1	Sup. F1	Unsup. F1	Sup. F1
Random	0.206	0.206	0.043	0.042	0.396	0.396
Raw features	0.575	0.575	0.585	0.585	0.422	0.422
DeepWalk	0.565	0.565	0.324	0.324	—	—
DeepWalk + features	0.701	0.701	0.691	0.691	—	—
GraphSAGE-GCN	0.742	0.772	**0.908**	0.930	0.465	0.500
GraphSAGE-mean	0.778	0.820	0.897	0.950	0.486	0.598
GraphSAGE-LSTM	0.788	0.832	**0.907**	**0.954**	0.482	**0.612**
GraphSAGE-pool	**0.798**	**0.839**	0.892	0.948	**0.502**	0.600
% gain over feat.	39%	46%	55%	63%	19%	45%





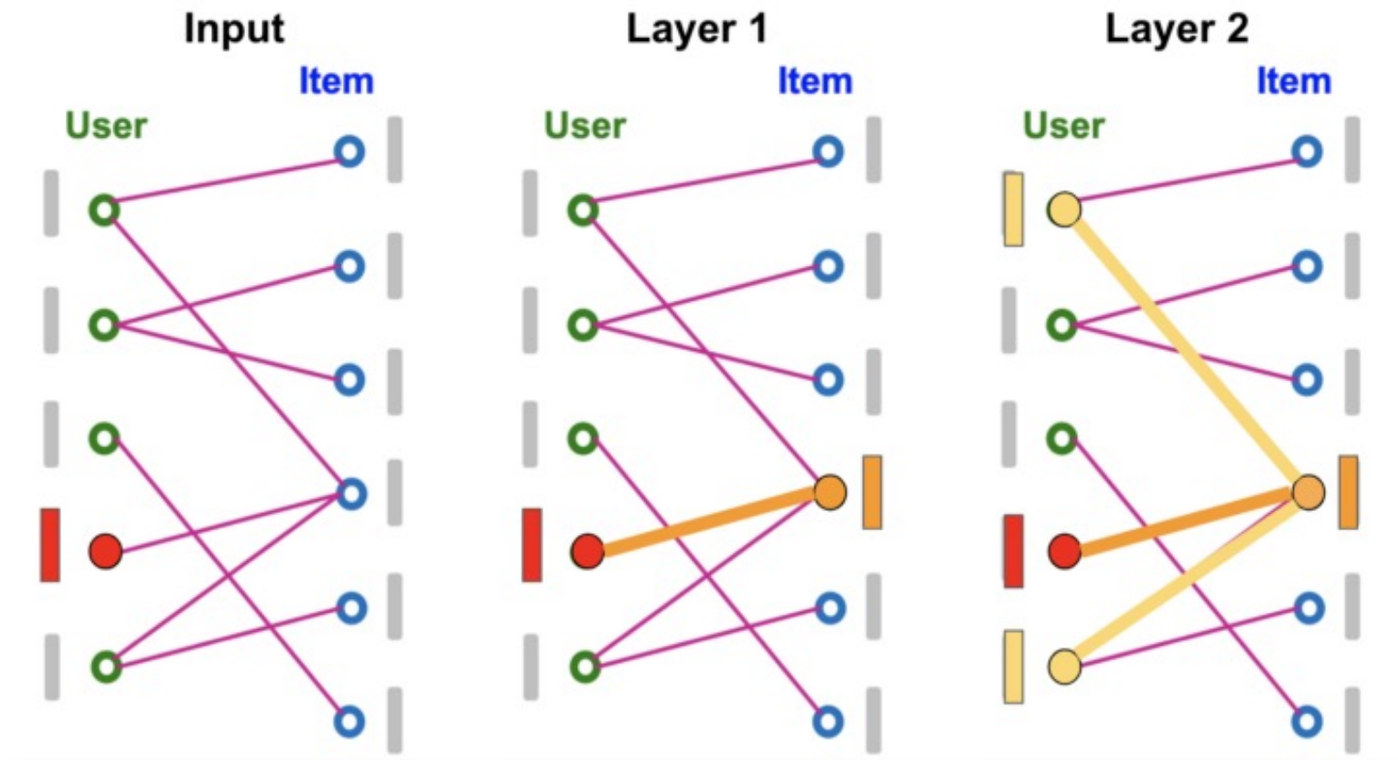
NGCF y LightGCN

- En NGFC y LightGCN retomamos la idea de representar nodos como vectores



NGCF y LightGCN: Usando información de vecinos

- El filtrado colaborativo tradicional no captura explícitamente información de los nodos a k-hops

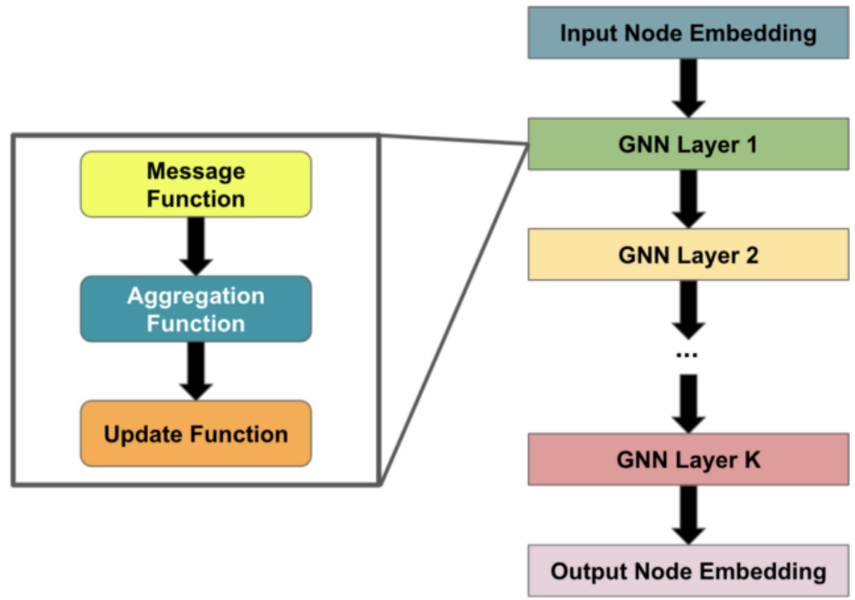


NGCF

- En cada capa de NGCF, el modelo aplica transformaciones de características que se aprenden:

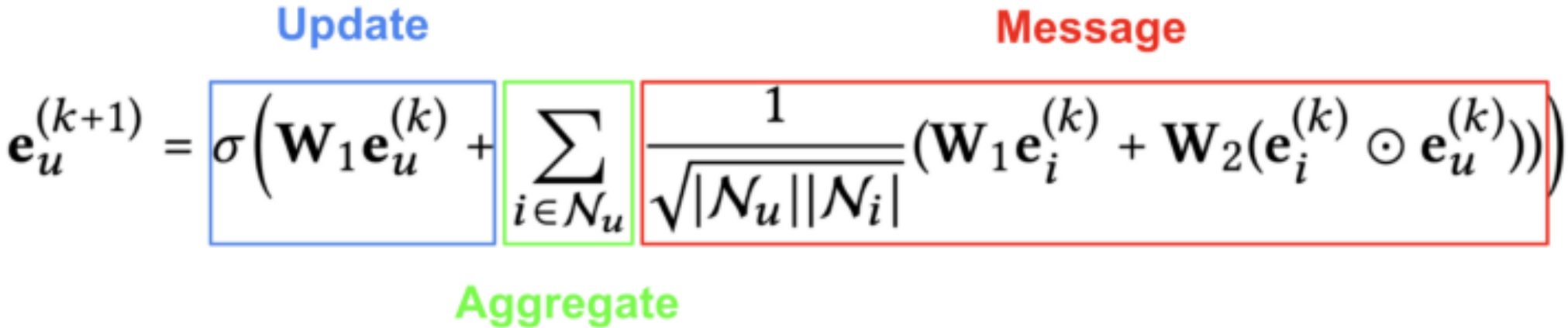
$$\mathbf{e}_u^{(k+1)} = \sigma\left(\mathbf{W}_1\mathbf{e}_u^{(k)} + \sum_{i\in\mathcal{N}_u}\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}(\mathbf{W}_1\mathbf{e}_i^{(k)} + \mathbf{W}_2(\mathbf{e}_i^{(k)}\odot\mathbf{e}_u^{(k)}))\right)$$

$$\mathbf{e}_i^{(k+1)} = \sigma\left(\mathbf{W}_1\mathbf{e}_i^{(k)} + \sum_{u\in\mathcal{N}_i}\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}(\mathbf{W}_1\mathbf{e}_u^{(k)} + \mathbf{W}_2(\mathbf{e}_u^{(k)}\odot\mathbf{e}_i^{(k)}))\right)$$



NGCF - detalle

- Se ve en la regla de actualización el paso de mensaje, la agregación y la actualización



LightGCN

- Inspirada en NGCF, elimina varios parámetros de las capas de la red neuronal bajo el supuesto que los embeddings de usuarios e ítems son suficientemente expresivos

$$\mathbf{e}_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(k)},$$

$$\mathbf{e}_i^{(k+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(k)}.$$

LightGCN II

- Notar que se eliminan las matrices W_i
- La actualización se hace directa de la agregación

Message

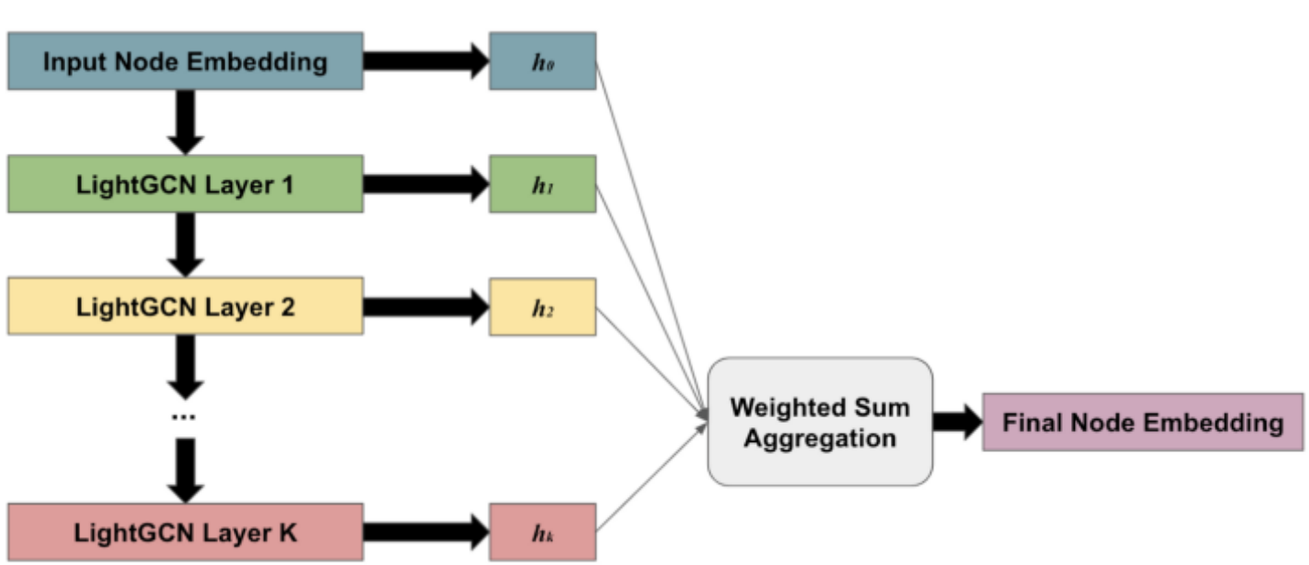
$$\mathbf{e}_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(k)}$$

Aggregate

LightGCN III: Agregación final

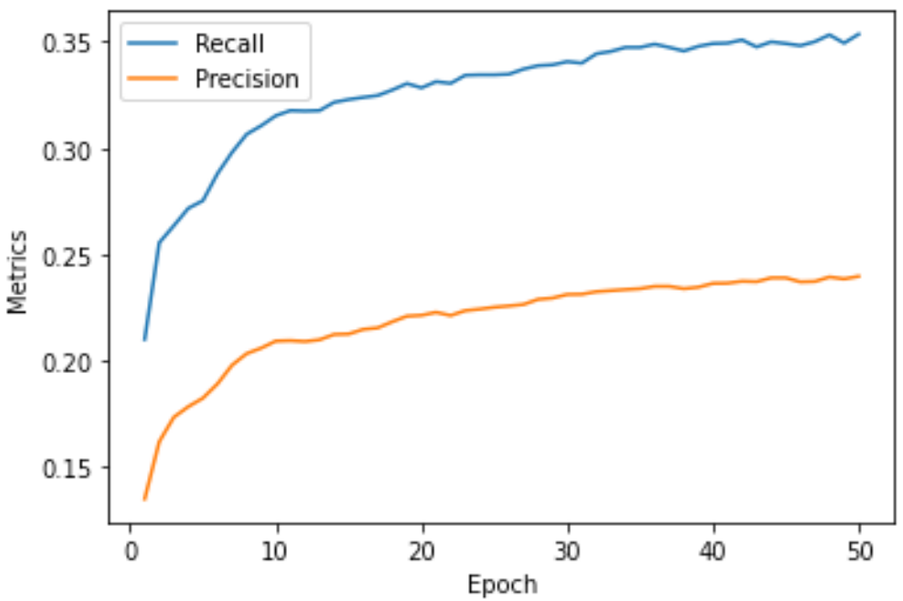
- La agregación final se hace a través de suma ponderada de los embeddings de cada capa

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{e}_u^{(k)}; \quad \mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \mathbf{e}_i^{(k)};$$

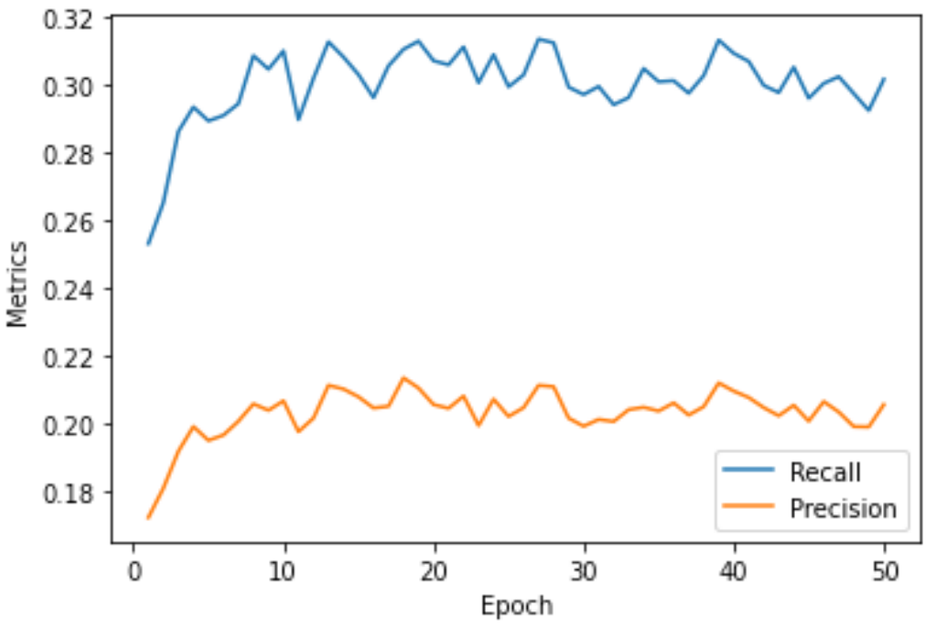


Resultados

- https://medium.com/stanford-cs224w/recommender-systems-with-gnns-in-pyg-d8301178e377



LightGCN precision@20 and recall@20 per training epoch.



NGCF precision@20 and recall@20 per training epoch.

References

- Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, *30*.
- Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 974-983).
- He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval* (pp. 639-648).