INFORME PROYECTO

Overlapping community detection with graph neural networks

Introducción

Existen grupos, clústers o comunidades en redes complejas. Frecuentemente son definidas como grupos de nodos que están densamente conectados, más conectados entre ellos que con el resto de la red. Detectar estas comunidades es necesario para la descripción de segmentos de una población, detección de datos atípicos, diseño de políticas dentro de una dinámica, etc.
Este proyecto pretende encontrar comunidades dentro de un conjunto de autores de artículos de investigación que publican en conjunto, esto es, una red de coautoría. Con el advenimiento de las redes neuronales de grafos podemos aprovecharnos de su estructura previamente estipulada, en principio, podría ayudar a la porción de la red profunda encargada de la extracción de características.

Estructura del repositorio

La estructura del repositorio es bastante simple:

1. 01-community_detection_gnn.ipynb es el notebook que contiene todo lo necesario para cargar datos, crear el modelo, entrenarlo y evaluarlo.
2. data/ es una carpeta con los datos necesarios.
    a. colombia_openalex_adj.npz matriz dispersa de scipy con la matriz de adjacencia.
    b. colombia_openalex_att.npz matriz dispersa de scipy con los atributos de los nodos de la red.
    c. colombia_openalex_classes.npz matriz diospersa de scipy con las clases a las que pertenecen los nodos de la red.
    d. colombia_openalex_mappings.pkl archivo pickle binarizado con el mapeo de los números de las matrices de atributos y clases a palabras (No es necesario para la ejecución del notebook).
3. ENTREGA1.PDF es el documento con el anteproyecto
4. INFORME_PROYECTO.PDF es este archivo.

Datos

Los datos se encuentran en el repositorio o en el link a zenodo: https://zenodo.org/record/7420261.

Estos datos tienen las siguientes caracteristícas:

1. El grafo completo en formato gml que puede ser importado por muchas librerías y software para en análisis de redes complejas (Fig. 1).

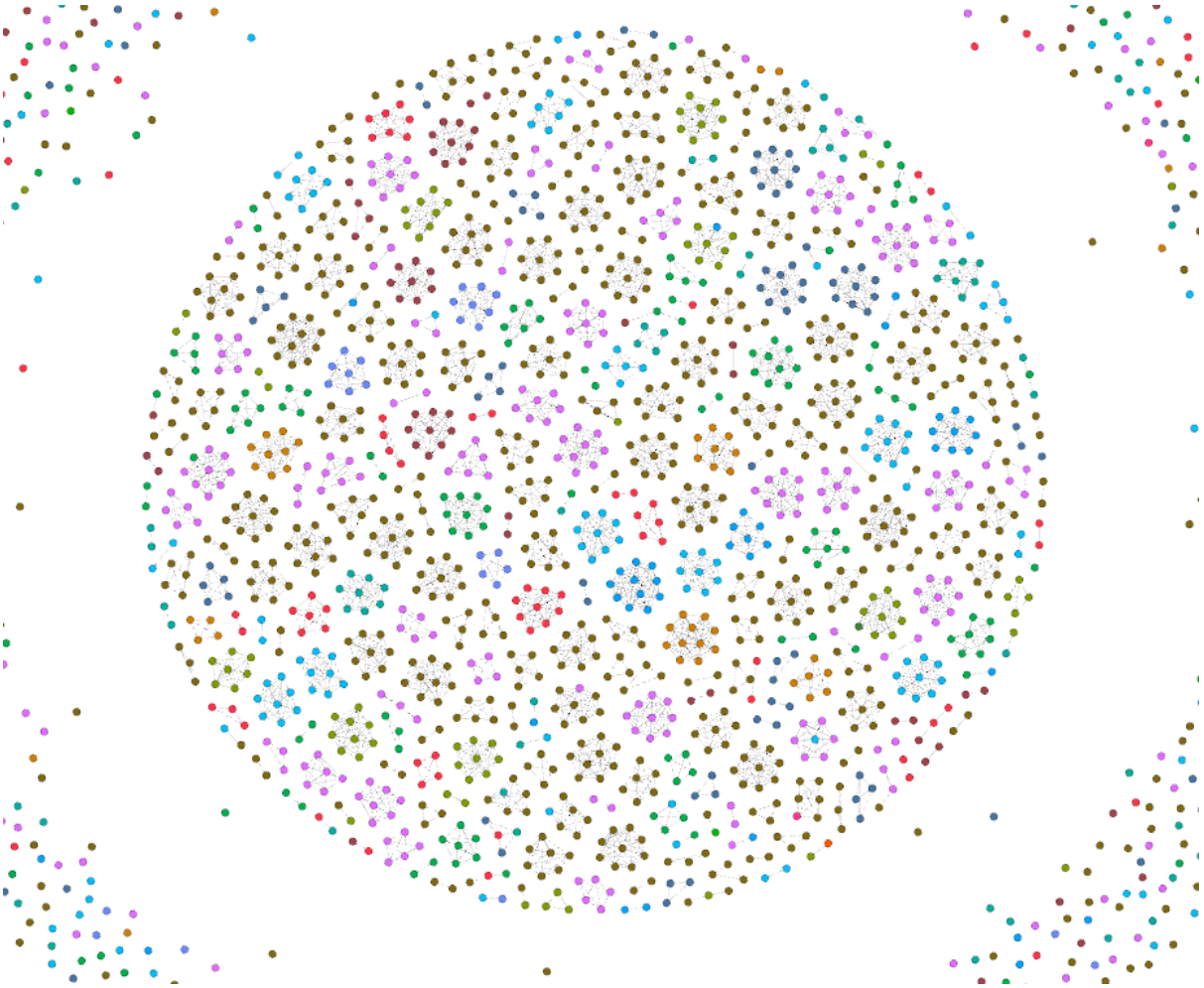

Fig. 1. Red de colaboración de autores colombianos. Fuente: Elaboración propia.

2. La matriz de adyacencia que contiene tantas filas y columnas como autores. Cada elemento de la matriz contiene un 1 si el par de autores (fila, columna) ha publicado un artículo en conjunto, si no, el elemento es 0 (Fig. 2).

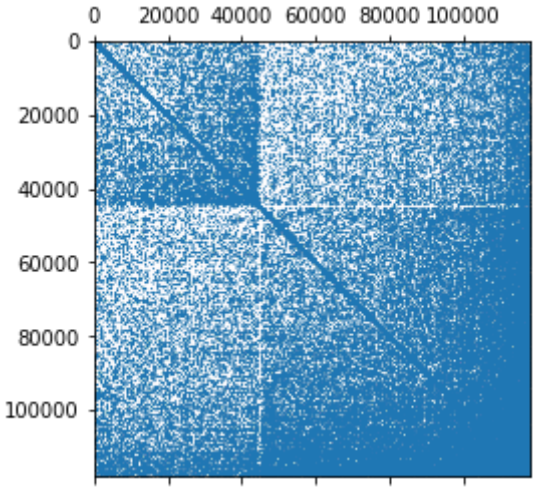

Fig. 2. Matriz de adyacencia de la red de coautoría. Fuente: Elaboración propia

3. La matriz de características en la que se condificaron 200 palabras aleatorias de entre los títulos y abstracts de los artículos de los investigadores cada columna contiene un elemento booleano que indica si el autor uso dicha palabra en alguno de sus artículos. Adicionalmente se codificó también la institución a la que pertenece el autor.
4. La matriz de clases que contiene los conceptos de nivel 0 de OpenAlex con los que ha sido etiquetado cada artículo de cada autor. Los conceptos son:
    a. Medicine
    b. Computer science
    c. Engineering
    d. Environmental science
    e. Materials science
    f. Chemistry
    g. Biology
    h. Physics
    i. Psychology
    j. Geology
    k. Philosophy
    l. Geography
    m. Art
    n. Political science
    o. Sociology
    p. Economics
    q. Mathematics
    r. Business
    s. History

Estructura de la solución

Modifiqué en varios aspectos la solución propuesta en el artículo de Shchur and Günnemann (2019):

- El conjunto de datos no parte de Microsoft Academics sino de su sucesor, OpenAlex.
- El conjunto de datos fue extraído con los autores colombianos de los trabajos en OpenAlex.

Presento un resumen de lo que logré implementar del artículo. El modelo consiste en una red neuronal convolucional de grafos con las siguientes características:

- Dos capas escondidas convolucionales de tamaños 128 y 64 con la operación **A** @ (**X** @ **W**) + **b**. Dónde **A** es la matriz de adyacencia, **X** es la matriz de caracteristicas, **W** son los pesos de la capa y **b** es el sesgo.
- Una capa con la misma convolución de la anterior pero de tamaño igual a la cantidad de comunidades a detectar, i.e. la capa de salida.
- Función de dropout especial que soporte operaciones con matrices dispersas después de cada capa.
- Función de activación ReLU en cada capa.
- La función de loss de Bernoulli-Poisson pero sin garantizar distribuciones uniformes en sus partes.

Iteraciones

El conjunto de datos es lo suficientemente grande para ser abordado por un modelo de deep learning. No hay grandes cantidades de redes de coautoría académica para entrenar con todos ellos. Usualmente las redes neuronales de grafos se valen de una extracción inteligente de sus partes para diseñar las particiones con las que se realizará el proceso de entrenamiento.
En proceso de entrenamiento y validación se realizan las siguientes tareas:

- Se extrae una porción de los datos, es decir, un subgrafo. Este proceso debe garantizar que el grafo no esté completamente conectado, o sea que de entre los nodos escogidos haya algunos conectados y otros que no.
- Con esta porción se predicen las comunidades a las que pertenecen los autores del subgrafo.
- Se ponen los gradientes en cero pues pytorch acumula los gradientes en la propagación hacia atrás.
- Con las comunidades obtenidas se calcula la función de loss de Bernoulli-Poisson.
- Se propaga hacia atrás.
- Se calcula la métrica NMI.

Resultados

Con los parámetros:

- hidden_sizes = [128,64]
- dropout = 0.5
- lr = 1e-3
- max_epochs = 500
- display_step = 50
- batch_size = 100
- threshold = 0.5

Se obtuvo un decrecimiento en la función de loss y un crecimiento en la métrica NMI como es de esperarse (Fig. 3)

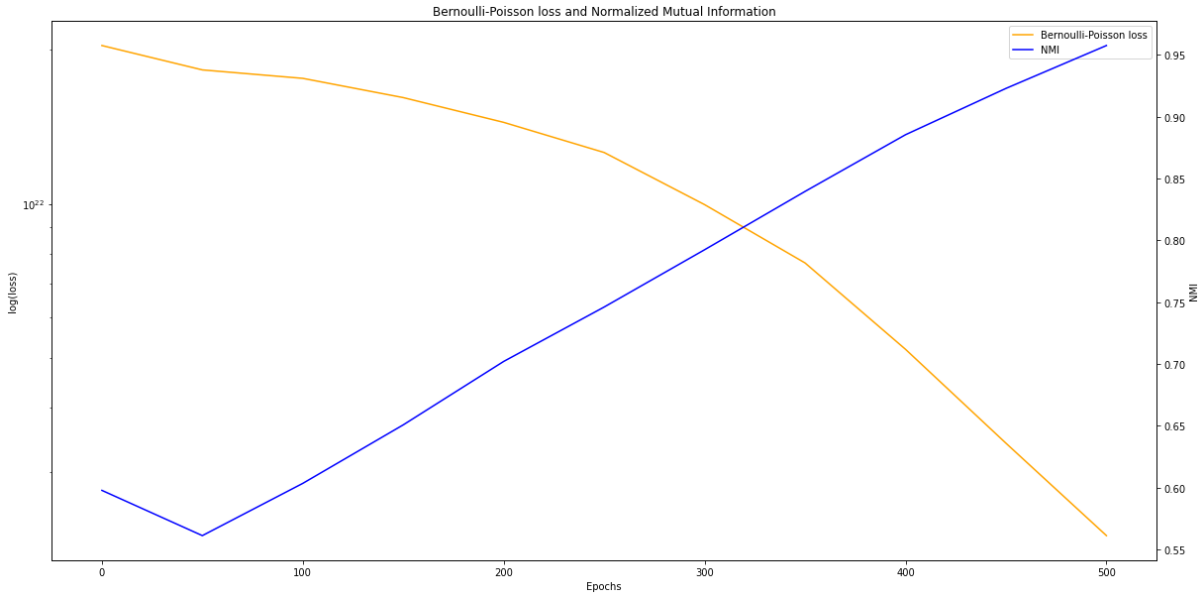

Fig. 3. Función de loss Bernoulli-Poisson en escala logarítmica (naranja) y métrica Normalized Mutual Information en escala lineal (azul). Fuente: Elaboración propia.

Trabajos a futuro

El modelo podría mejorar si:

- Se mejoran las variables descriptoras de los investigadores.
- Se hace normalización de dichas variables.
- Se implementa una normalización por lotes.
- Se mejora la función de loss para compensar el desbalance.
- Se implementa un término de regularización.
- Se mejora el extractor de subgrafos.
- Se expande las afiliaciones de los autores pues pueden estar afiliados a más de una institución.

References

McDaid, A. F., Greene, D., & Hurley, N. (2011). Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*.

Shchur, O., & Günnemann, S. (2019). Overlapping community detection with graph neural networks. *arXiv preprint arXiv:1909.12201*.