Procesamiento de Grafos Usando la Librería SNAP y Gephi

Alejandro Herce
Tecnológico de Monterrey Campus Santa Fe
Carlos Lazo 100, Colonia Santa Fe.
Distrito Federal, México
A01021150@itesm.mx

ABSTRACT

Este documento demuestra los pasos realizados en la investigación para importar un grafo de ejemplo de Stanford Large Network Dataset Collection sobre la red social Twitter, que se importará en la librería de SNAP, donde posteriormente será procesado y se exportará en diferentes formatos incluyendo GraphML, GEFX, GDF y GraphSON, basado en JSON. Posteriormente, se importarán los grafos generados en Gephi para generar la visualización de estos.

Keywords

Grafo; JSON, Gephi, GraphML, GEFX, GDF, GraphSON

1. INTRODUCCIÓN

En computación, un grafo es una estructura de datos que consiste en un conjunto de objetos llamados nodos, unidos por aristas o arcos. Un grafo típicamente se representa como un conjunto de puntos *(nodos)* unidos por líneas *(aristas)*. Existen dos tipos de grafos, dirigidos y no dirigidos. Los grafos dirigidos son aquellos en los que las aristas o arcos apuntan en una dirección, y en los no dirigidos las aristas apuntan en ambas direcciones.

En esta investigación, utilizamos la librería de SNAP para procesar nuestro grafo, la cual fue creada por Stanford en 2004. La librería SNAP *(Stanford Network Analisis Platform)* es una librería para data mining en grafos escrita en C++ y que puede ser escalada a millones de nodos y aristas. La librería también permite manipular grafos de gran tamaño, calcular propiedades estructurales y generar grafos aleatoriamente.

2. TRABAJANDO CON LOS GRAFOS

Como ya habíamos mencionado, utilizaremos SNAP para importar y procesar nuestro grafo, que después se exportará a cuatro diferentes formatos. La información del grafo fue obtenida del *Stanford Large Network Dataset Collection*[1], en donde se pueden descargar varios datasets recolectados por Stanford sobre diferentes redes sociales.

Nuestro dataset elegido es el de la red social Twitter, que cuenta con 81,306 nodos y 1,768,149 aristas.

2.1 Importando el Grafo

Una vez seleccionado y descargado el dataset a utilizar, necesitamos importarlo a nuestro programa en donde será procesado por la librería *SNAP*. El archivo del dataset es un txt en donde se incluyen todos los nodos y aristas necesarios para generar el grafo, sólo se tiene que importar con nuestra librería. El archivo txt se tiene que ubicar en el mismo folder que nuestro main.cpp para que pueda ser leído por *SNAP*.

Después de inicializar un grafo en *SNAP*, tenemos que importar los datos del archivo txt. Para importar el dataset, sólo se requiere una línea de código:

TSnap::LoadEdgeList<Graph>("twitter.txt",0,1);

2.2 Procesando y Exportando el Grafo

Como habíamos mencionado, el grafo se exportará en cuatro diferentes formatos, GraphML, GEFX, GDF y JSON a través de la librería GraphSON. No tocaremos los detalles específicos de cada formato, pues la documentación es accesible para todos.

De forma muy general, al exportar nuestro grafo en realidad lo que estamos haciendo es guardar los nodos y aristas como si fuera un documento txt pero con una plantilla especifica para cada librería. Por ejemplo, los nodos en GraphML se guardan así: *<node id="nodo1">* y de la misma forma las aristas. Las librerías no son iguales, pero lo que importa es el concepto de la plantilla, que es muy parecido a hacer una lista de nodos y aristas, solo que con mas símbolos.

Para exportar el grafo, lo único que se tiene que hacer es un ciclo *for* donde se va recorriendo el grafo nodo por nodo y arista por arista y guardándolo en el archivo con la plantilla correspondiente al formato seleccionado.

2.3 Tiempos de Ejecución

Además de exportar el grafo, el programa también mide el tiempo de ejecución al exportar en cada formato. El tiempo se mide con la función *high resolution clock*, tomando el tiempo justo antes de la ejecución y justo al terminar para después hacer una resta y obtener el tiempo en milisegundos que tardó en ejecutarse cada función de exportación.

Para ejecutar el programa, se utilizó una maquina virtual de Ubuntu con 4 GB de RAM y 4 cores (2 reales y 2 en hyperthreading). Los tiempos para exportar 81,306 nodos y 1,768,149 aristas a los cuatro formatos se pueden apreciar

[1]Link: https://snap.stanford.edu/data/index.html

DOI: 10.1145/1235

Table 1: Tiempos de ejecución

Función	Tiempo (Milisegundos)
GraphML	775
GEFX	1536
GDF	677
GraphSON	1330

en la Tabla 1.

2.4 Importando en Gephi

Una vez exportado el grafo en los diferentes formatos, podemos importarlo en Gephi. Gephi es un programa multi plataforma para la visualización y exploración interactiva de todo tipo de redes y sistemas complejos, como grafos. Gephi permite importar, explorar y manipular grafos en tiempo real, además de poder exportarlos en varios formatos como PDF y PNG. Este programa es compatible con tres formatos en los que exportamos nuestro grafo: GraphML, GEFX y GDF.

Una vez importado el grafo en Gephi, se le aplicaron colores a los nodos en varios tonos de azul al blanco y después se le aplico la función Force Atlas 2. Dicha función es un algoritmo continuo para la espacialización de redes que funciona aplicando fuerzas, repeliendo los nodos unos de otros como imanes, pero al mismo tiempo las aristas actúan como resortes uniéndolos. Estas fuerzas crean un balance, lo que permite que el resultado final sea mas comprensible.

Los resultados del algoritmo aplicado al grafo se pueden ver en las Figuras 1 y 2.

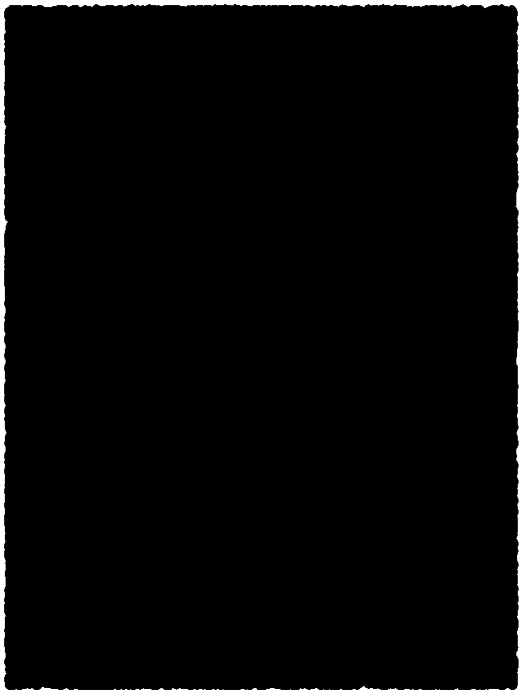

Figure 1: Antes de Force Atlas 2

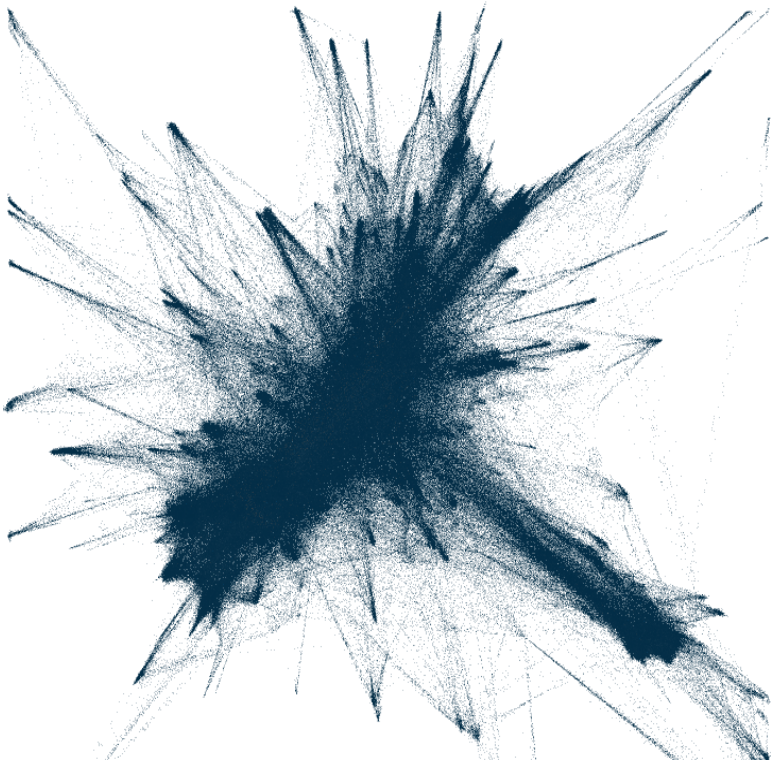

Figure 2: Después de Force Atlas 2

3. CONCLUSIONES

El proceso de importar y exportar el grafo en diferentes formatos fue relativamente simple. Es un simple archivo en el cual se inserta una lista de nodos y aristas. Los problemas surgieron cuando pasamos a la parte de Gephi, pues el programa es un poco lento y con la cantidad de nodos y aristas generados tardó mucho en importar y en aplicar el algoritmo Force Atlas 2.

Originalmente quería usar el dataset de Youtube con mas de 1 millón de nodos y casi 3 millones de aristas, pero Gephi no se dejó y se cerro varias veces. En la parte de exportarlo a los cuatro formatos, no tuve ningún problema y los tiempos de ejecución rondaban entre los mil y dos mil milisegundos, bastante decentes si consideramos el tamaño del grafo.

En general el proceso fue muy simple y Gephi ofrece muchas opciones para explorar los grafos, pero faltó un poco mas de tiempo para aprender un poco mas acerca de este programa.

4. GITHUB

https://github.com/alexherce/Algoritmos