Pontificia Universidad Javeriana Cali

Facultad de Ingeniería
Ingeniería de Sistemas y Computación

PROCESAMIENTO DE GRANDES VOLÚMENES DE DATOS

Integrantes:
Ana María García, Andrés Felipe Delgado, Leonardo Sáez, Katherine Camacho

Project 3 (Spark GraphX + Neo4j + MLlib)

Período 2020-2

1. Descripción de la colección Neo4j

- Lo primero que se hizo con el conjunto de datos fue realizarle el mismo preprocesamiento de los proyectos 1 y 2. A continuación, debido a que el conjunto de datos tiene 10 atributos, los cuales para la representación en un grafo son demasiados, se realizó una reducción de dimensionalidad sin perder información, con ayuda de la librería Sklearn.manifold.TSNE de Python. De este modo, el conjunto de datos quedó con dos atributos y el mismo número de registros (25.478)
- Posteriormente, con ayuda de la librería Sklearn.neighbors.KDTree se obtuvieron los 2 vecinos más cercanos de cada registro y se creó una tabla con las relaciones, solo se buscó a los dos vecinos más cercanos para evitar que el grafo tuviese conexiones innecesarias, convirtiéndolo en un grafo computacionalmente inmanejable
- Luego de esto, se crearon los nodos del grafo en Neo4j, donde cada nodo representa un registro del conjunto de datos. Luego se crearon las relaciones con la tabla creada a partir de los dos vecinos más cercanos de cada Avila. De está forma tenemos nodos con dos atributos internos conectados con sus dos vecinos más cercanos que en la gran mayoría de los casos van a pertenecer a un mismo autor.

CÓDIGO DE IMPLEMENTACIÓN DEL GRAFO NEO4J

```
load csv with headers from 'file:///nodos.csv' as row
merge (p:Pagina{id: row.idNodo, F1: row.caracteristica1, F2: row.caracteristica2})
return count(row)
```

Figura 1: Carga de Nodos

```
load csv with headers from 'file:///relaciones.csv' as row
match (p:Pagina)
where p.id = row.idNodo1
match (b:Pagina)
where b.id = row.idNodo2
merge (p)-[r:FRIEND_WITH]-(b)
return count(row)
```

Figura 2: Carga de Relaciones

REPRESENTACIÓN GRÁFICA DEL GRAFO NEO4J

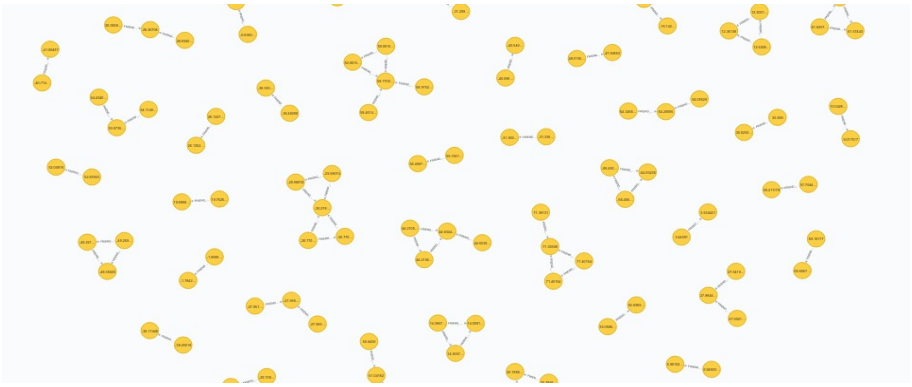

Figura 3: Grafo Neo4j

2. Medidas topológicas

- Triangle Counting / Clustering Coefficient: Permite detectar conjuntos de tres nodos, donde cada nodo tiene una relación con todos los demás nodos.Esto nos permite identificar nodos que con una mayor probabilidad pertenecen a un autor.
- Page Rank: Mide la importancia de cada nodo que está dentro del gráfico, entonces según el número de relaciones entrantes y según el nodo de donde proviene dicha relación se establece la importancia del nodo.
- Betweenness Centrality: Es una forma de detectar la influencia que tiene un nodo sobre el flujo de información en un grafo.
- Louvain:Permite detectar comunidades dentro de una red, para lo cual evalúa cuánto más densamente están los nodos conectados dentro de una comunidad, en comparación con qué tan conectados estarían en una red aleatoria.
- Strongly Connected Components: encuentra conjuntos máximos de nodos conectados en un grafo dirigido. Un conjunto se considera un componente fuertemente conectado si hay una ruta dirigida entre cada par de nodos dentro del conjunto. Es útil en nuestro caso debido a que se pueden relacionar los nodos dentro de un SCC como paginas pertenecientes a un mismo autor.

2.1. Comparación de las 5 medidas topológicas que se evaluaron

- Accuracy Random Forest - Triangle Counting: 0.638953040801

- Accuracy Random Forest - Page Rank: 0.643956889915
- Accuracy Random Forest - Betweenness Centrality: 0.652040030793
- Accuracy Random Forest - Louvain: 0.858352578907
- Accuracy Random Forest - Strongly Connected Components: 0.957467282525

2.2. ¿Qué medida es más útil para la predicción?

La medida más útil para la predicción de los autores fue Strongly Connected Components, pues esta medida fue la que tuvo el mejor desempeño al implementar el modelo de Random Forest, escogido por ser el modelo con mejor desempeño del proyecto 1. El exito de esta medida se debe a la forma en que se creó el grafo donde se buscó relacionar los nodos con sus vecinos mas cercanos aumentando la probabilidad de la existencia de componentes fuertemente conexos en el grafo ayudando mucho en el proceso de clasificación. Por este mismo motivo la medida Louvain tuvo un muy buen desempeño.

3. Comparación de los resultados de la predicción con y sin medidas topológicas

PREDICCIÓN SIN MEDIDAS TOPOLÓGICAS

Para esto se tomaron los resultados de la predicción del proyecto 1, donde no se realizaron medidas topológicas:

- Accuracy score de Regresión Logística = 0.5613997879109226
- Accuracy Score de Árbol de Decisión = 0.8598091198303287
- Accuracy Score de Bosque Aleatorio = 0.8914103923647932

Figura 4: Predicción Sin Medidas Topológicas

Como podemos ver, el modelo que mejor desempeño tuvo fue el Bosque Aleatorio, y por esa razón será el que se implementará en este proyecto con medidas topológicas, para posteriormente realizar la comparación.

PREDICCIÓN CON TODAS LAS MEDIDAS TOPOLÓGICAS

Después de leer todos los nodos del grafo con ayuda de Spark, se creo un dataframe con los atributos del grafo los cuales incluyen las medidas topológicas, para luego evaluar este conjunto de datos con el modelo de Random Forest". Los resultados obtenidos fueron los siguientes:

Accuracy Score of RandomForestClassifier is = 0.970554272517

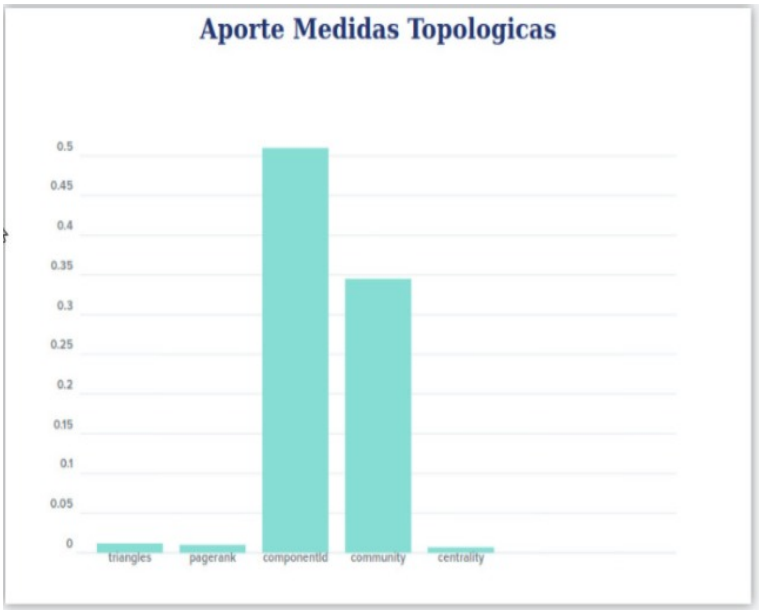

Figura 5: Aporte de Cada medida Topologica a la predicciòn cuando se usan todas las medidas a la vez. Community hace referencia a Louvain y componentID a Strongly Connected Components

4. Conclusiones

- en todos los proyectos realizados en el curso se pudo ver la relevancia del procesamiento de los datos, pudimos observar como un buen preprocesamiento tiene grandes impactos en la calidad de las predicciones que obtenemos.
- En cada proyecto se hizo uso de diversas tecnologias previstas para diferentes problemas, en el primer proyecto nos enfrentamos al problema de trabajar con grandes cantidades de datos y por este motivo se hizo uso de HDFS y spark, en el segundo batallamos con la necesidad de recibir baches de información en tiempo real y realizar predicciones con ellas y en este tercer y ultimo proyecto con las métricas interesantes que puede brindar el análisis de un grafo y que impacto tienen en las predicciones que hacemos, todo de la mano de neo4j.
- a nivel grupal consideramos que hemos expandido nuestras habilidades de

búsqueda de soluciones y lectura de documentación haciendo una abstracción de lo necesario para eliminar el problema.

- Fue muy interesante entender y poner en practica las diversas arquitecturas y tecnologías que existen para procesar datos y hacer predicciones sobre ellos,ampliando los contextos donde podamos aplicar nuestros conocimientos en machine learning.