



MASTER EN DATA SCIENCE

Curso Académico 2018/2019

Trabajo Fin de Master

Divider Greedy algorithm for performing community detection in social networks

Autor : Michel Maes Bermejo

Tutor : Jesús Sánchez-Oro Calvo

Resumen

En este proyecto abordaremos la creación de un algoritmo capaz de detectar comunidades en redes sociales, un problema en el que los algoritmos tradicionales no ofrecen soluciones rápidas. Para ello usaremos un enfoque metaheurístico basado en la división de comunidades e intentaremos mejorar sus prestaciones a lo largo de varias iteraciones.

Índice general

1. Introducción y motivación	1
2. Objetivos	2
3. Tecnologías, Herramientas y Metodologías	3
3.1. Tecnologías	3
3.1.1. Java	3
3.2. Herramientas	4
3.2.1. Control de versiones: Git	4
3.2.2. Entorno de desarrollo: IntelliJ	4
3.3. Metodologías	4
4. Descripción informática	5
5. Estudio comparativo	6
6. Conclusiones del proyecto y trabajos futuros	7
Bibliografía	8

Capítulo 1

Introducción y motivación

Desde su nacimiento, Internet ha logrado conectar a las personas de forma sencilla. El ejemplo más claro son las redes sociales, que han crecido de forma contundente los últimos años. La rapidez con la que se transmite la información ha provocado que muchas personas las utilicen como medio de información de referencia frente a los tradicionales. Esto ha desencadenado un creciente interés de diferentes marcas e incluso de otros medios de aprovechar este fenómeno para su beneficio, aprovechando la estructura de red (o grafo) en la que se basa.

Desde el punto de vista de un científico de datos, resulta interesante estudiar la estructura que forman estas redes, que constituyen un ejemplo perfecto de un grafo. Una de las áreas más interesantes que utiliza como base este tipo de redes es la detección de comunidades, que cuenta con multitud de aplicaciones.

Actualmente existen multitud de algoritmos que abordan el problema de la detección de comunidades. Los más tradicionales y exactos no son viables para abordar los grandes volúmenes de datos que generan las redes sociales, por lo que se ha optado por soluciones heurísticas, no tan exactas, pero mucho más rápidas.

La motivación de este proyecto nace de la idea de crear una solución heurística diferente a las preexistentes (normalmente basadas en algún algoritmo destructivo), optando por un enfoque destructivo.

Capítulo 2

Objetivos

El objetivo principal de este proyecto será la creación de un algoritmo de detección de comunidades. Para ello utilizaremos un enfoque destructivo (comenzar con todos los nodos del grafo en la misma comunidad e ir haciendo particiones) contrario a un enfoque más utilizado, constructivo (comenzar con cada nodo en su propia comunidad e ir agrupandolos). Durante la construcción, usaremos la modularidad como métrica para evaluar nuestra solución. Realizaremos una serie de iteraciones en las cuales iremos mejorando el algoritmo con el fin de obtener la mejor versión del mismo.

Finalmente, compararemos nuestro algoritmo con otros preexistentes para valorar si es efectivo el enfoque elegido. En esta fase usaremos otras métricas sobre las particiones obtenidas como la conductancia o cobertura.

Capítulo 3

Tecnologías, Herramientas y Metodologías

3.1. Tecnologías

3.1.1. Java



Figura 3.1: Logo Java

Java (Figura 3.1) es un lenguaje de programación de propósito general, concurrente y orientado a objetos. Su sintaxis deriva en gran medida de C y C++. Uno de los principales atractivos de Java es su máquina virtual (JVM) que nos permite ejecutar nuestro código Java en cualquier dispositivo, independientemente de la arquitectura.

3.2. Herramientas

3.2.1. Control de versiones: Git



Figura 3.2: Logo Git

Git¹ (Figura 3.2) es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

3.2.2. Entorno de desarrollo: IntelliJ



Figura 3.3: Logo IntelliJ

IntelliJ² (Figura 3.3) es un IDE para Java desarrollado por JetBrains ideado para mejorar la productividad del programador.

3.3. Metodologías

Dado que pretendemos obtener la mejor versión de nuestro algoritmo, el desarrollo seguirá una metodología iterativa-incremental³, dónde en cada iteración se propondrán y evaluarán diferentes mejoras sobre el algoritmo.

¹<https://git-scm.com/>

²<https://www.jetbrains.com/idea/>

³<https://proyectosagiles.org/desarrollo-iterativo-incremental/>

Capítulo 4

Descripción informática

Capítulo 5

Estudio comparativo

Capítulo 6

Conclusiones del proyecto y trabajos futuros

estigmérgicos [1].

Bibliografía

- [1] J. Sánchez-Oro and A. Duarte. Iterated greedy algorithm for performing community detection in social networks. 2018.