Дослідження засобів кластеризації графів у графовій СУБД Neo4j для виявлення співтовариств у соціальних мережах

Єлизавета Мелешко
Кафедра кібербезпеки та програмного забезпечення,
Центральноукраїнський національний технічний університет,
м. Кропивницький, Україна
elismeleshko@gmail.com

Abstract. In this work, the research to the main graph clustering methods and also the means for graph clustering, which are provided by the graphical DBMS Neo4j, was carried. The software in the Python programming language for testing graph clustering methods for detecting communities in social networks was developed. The main graph clustering means of DBMS Neo4j, using developed software, were tested.

Ключові слова: кластеризація графів, графові бази даних, соціальні мережі, аналіз соціальних мереж, модулярність.

Однією з основних задач аналізу соціальних мереж (СМ) та соціальних графів є задача виділення співтовариств (кластерів).

Методи кластеризації графу (КГ) за принципом роботи можна поділити на наступні: засновані на оптимізації модулярності [1], засновані на спектральних особливостях графу та засновані на оцінці ентропії системи [2]. За результатами роботи дані методи можна поділити на такі, що розбивають граф на кластери, які не перетинаються (Edge Betweenness, Label Propagation, FastGreedy, WalkTrap, Leading Eigenvector, MultiLevel, тощо), та на ті, що розбивають граф на кластери, які перетинаються (k-Clique Perlocation, BigCLAM, DEMON, CONGO, тощо) [3].

Метою даної роботи є дослідження засобів кластеризації графів СМ наявних у графовій СУБД Neo4j. В ході проведення дослідження було здійснене тестування даних методів КГ з застосуванням програмного забезпечення розробленого на мові програмування Python.

Основна частина

Neo4j – це система управління базами даних типу NoSQL, заснована на представлені даних у вигляді графів [4]. Вона має вбудовану бібліотеку Graph algorithms з розпаралеленими алгоритмами для роботи з графами. Для кластеризації графів дана бібліотека містить реалізації наступних алгоритмів [4]:

– **Louvain** (функція algo.louvain) – алгоритм кластеризації графів, заснований на оптимізації модулярності. Вузли об'єднуються у кластери так, щоб збільшити модулярність. Є одним з найшвидших алгоритмів на основі модульності, і добре працює з великими графами.

– **Label Propagation** (функція algo.labelPropagation) – кластеризує граф, використовуючи лише його структуру. Кожна вершина в графі поміщається в той кластер, якому належить більшість його сусідів. Якщо ж таких кластерів декілька, то вибирається випадково одне з них. У початковий момент часу всім вершинам ставиться у відповідність окреме співтовариство.

– **Triangle Counting / Clustering Coefficient** (функція algo.triangleCount) – визначає кількість трикутників, що проходять через кожен вузол у графі. Трикутник являє собою набір з трьох вузлів, в якому кожен вузол має зв'яки з усіма іншими вузлами. На основі одержаних даних визначає коефіцієнт кластеризації. Хоча розробники СУБД Neo4j відносять даний алгоритм до алгоритмів кластеризації, слід зазначити, що це не зовсім коректно, адже знаходяться трикутники, а не кластери у графі,

а трикутник лише частковий випадок кластера.

Також серед реалізацій методів КГ у документації до бібліотеки Graph algorithms Neo4j [4] пропонуються до використання **Connected Components** та **Strongly Connected Components** які знаходять зв'язані підграфи незв'язного графу для неорінтованих та орієнтованих графів відповідно, та **Balanced Triads** – алгоритм оцінки структурного балансу графу СМ, що знаходить збалансовані та незбалансовані тріади у мережі. Оскільки ці алгоритми не розбивають граф на кластери, а виконують дещо інші функції, їх тестування у рамках даного дослідження не проводилося.

Для реалізації інших методів КГ необхідно розробляти власні функції або запити до Neo4j, як показало дослідження ця СУБД надає досить зручний функціонал для таких розробок.

У розроблюваній системі для роботи з Neo4j була використана бібліотека neo4j.v1 для мови Python. Для тестування методів КГ було згенеровано випадковий граф з властивостями подібними до властивостей графів СМ.

Результати роботи методу Louvain наведені у табл.1.

Таблиця 1. Результат виклику функції Louvain для графу з 2348 вузлами та 10762 зв'язками

Кількість кластерів	Кількість ітерацій	Час Завантажен-ня даних, мс	Час роботи алгоритму, мс	Час запису результатів у граф, мс
11	2	18	34	93

Результати роботи методу labelPropagation наведені у табл. 2.

Таблиця 2. Результат виклику функції labelPropagation для графу з 2348 вузлами та 10762 зв'язками

Кількість кластерів	Кількість ітерацій	Час Завантажен-ня даних, мс	Час запуску алгоритму, мс	Час запису результатів у граф, мс
3	3	25	1	3

Результати роботи методу triangleCount наведені у табл. 3.

Таблиця 3. Результат виклику функції triangleCount для графу з 2348 вузлами та 10762 зв'язками

Кількість трикутників	Середній коефіцієнт кластеризації	Час Завантажен-ня даних, мс	Час роботи алгоритму, мс	Час запису результатів у граф, мс
10630	0.1756	18	3	7

Як видно з таблиць 1-3 дані методи КГ діють зовсім по різному та дають різні результати. Метод Louvain знаходить більше кластерів, ніж метод labelPropagation, а метод triangleCount взагалі шукає не кластери, а трикутники. LabelPropagation на зашумлених графах часто робить об'єднання всіх вершин в один кластер, або невелику кількість кластерів, що в даному випадку під час тестування і відбулося.

Висновки

Було проведено дослідження методів кластеризації графів. Досліджено можливості графової СУБД Neo4j для реалізації даних методів. Для виділення кластерів СУБД Neo4j пропонує декілька реалізованих у її бібліотеці Graph algorithms методів, а саме Louvain, Label Propagation та Triangle Counting. Інші методи кластеризації графів треба реалізовувати самостійно, але Neo4j надає багато зручних інструментів для цього. Як показало дослідження, серед засобів кластеризації графів у СУБД Neo4j для виділення співтовариств у СМ найкраще підходить функція Louvain.

Література

[1] Д. Форман "Много цифр. Анализ больших данных при помощи Excel", Альпина Паблишер, 464 с., 2016.

[2] М.И. Коломейченко, И.В. Поляков, А.А. Чепов-ский, А.М. Чеповский "Выделение сообществ в графе взаимодействующих объектов", Фундаментальная и прикладная математика, Т. 21, № 3,С. 131–139, 2016.

[3] Е.С. Никишин "Методы выделения сообществ в социальных графах", 2016, URL: http://www.machinelearning.ru/wiki/images/8/8a/Nikishin_coursework_community_detection.pdf

[4] "neo4j", 2019, URL: https://neo4j.com/