Отчет по летней практике: «Анализ сообществ ВКонтакте»

Выполнили: студентки 1 курса ОП ПАД ФКН ВШЭ
Краузе Наталья, Павлеева Мария, Смотрова Кристина

НИУ ВШЭ г.Москва,
2019 год

Содержание:

Постановка задачи

В данной работе нашей задачей был анализ групп социальной сети ВКонтакте с использованием графовых алгоритмов. В качестве сообщества для анализа мы взяли группу «Квестчн», в которой на момент написания работы было 442 пользователя, большинство из которых - студенты ФКН ВШЭ.
Глобально задачу можно разделить на следующие составляющие: получение данных о пользователях группы с использованием VK API, построение графа по полученным данным, запуск и анализ работы алгоритмов на графе пользователей. Дополнительно была реализована визуализация графа пользователей с помощью библиотеки NetworkX.

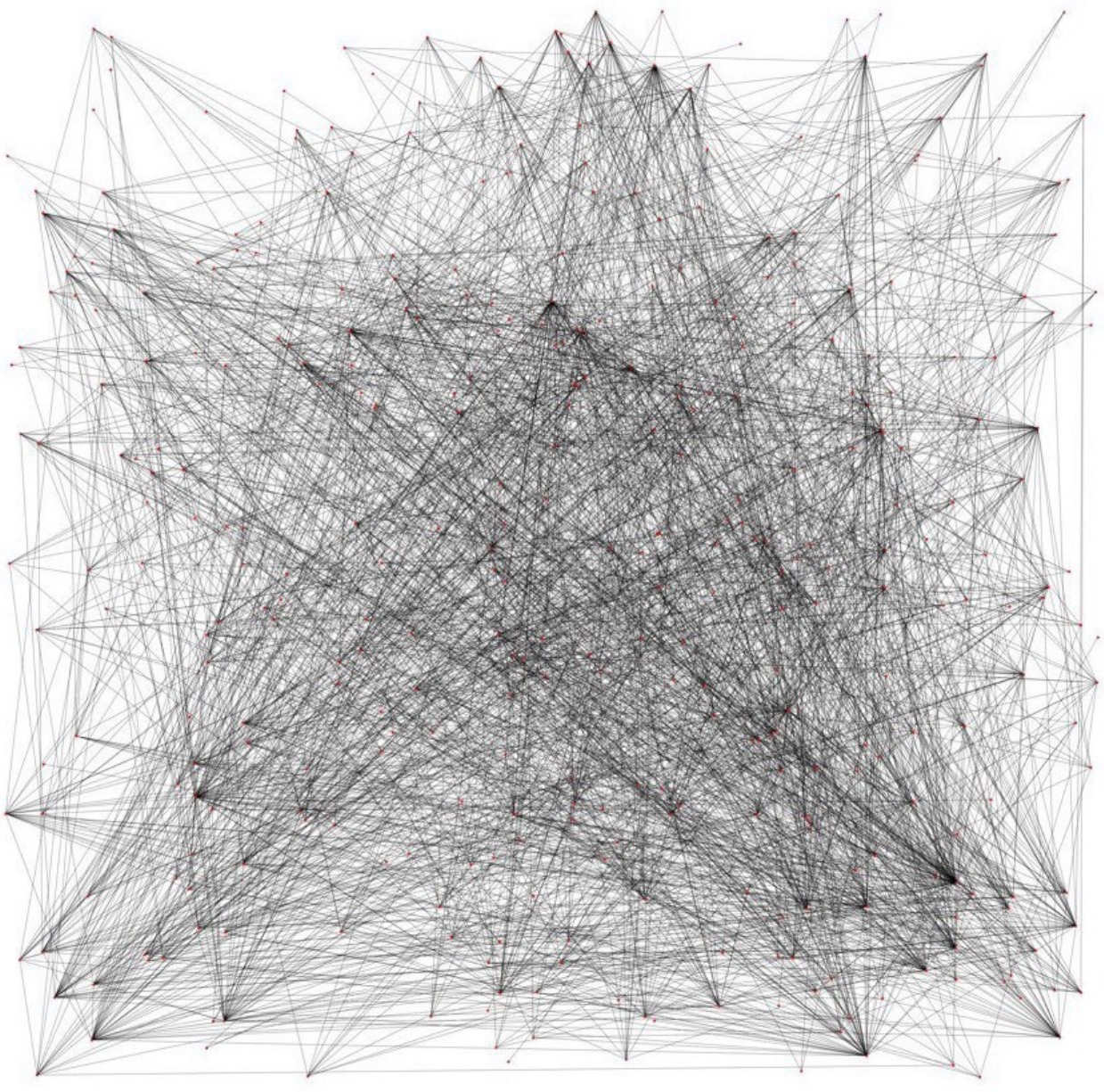
Получение данных

Для того, чтобы получить данные, мы использовали запросы к VK API с помощью библиотеки requests. Сначала мы получили доступ к списку участников группы: это делает метод getMembers. После этого для каждого из пользователей необходимо было получить список всех его друзей, чтобы затем проверять для всех пользователей группы, входят ли они в список друзей пользователя (такой перебор занимает линейное от количество пользователей время, так как список пользователей группы мы поместили в неупорядоченное множество, где операция поиска выполняется за константное время), следовательно, общая асимптотика получения информации о связях между пользователями требовала квадратичное время. В процессе получения списка друзей возникли следующие проблемы: у некоторых пользователей закрытые аккаунты, поэтому информацию об их друзьях получить невозможно. Ещё часть пользователей была заблокирована администрацией ВКонтакте или имела удаленные страницы, откуда тоже нельзя было получить информацию. Так же имелись ограничения на количество запросов в секунду (не больше 3), поэтому нам пришлось вводить программу «в сон» на 1 секунду, чтобы она не завершала свою работу аварийно.

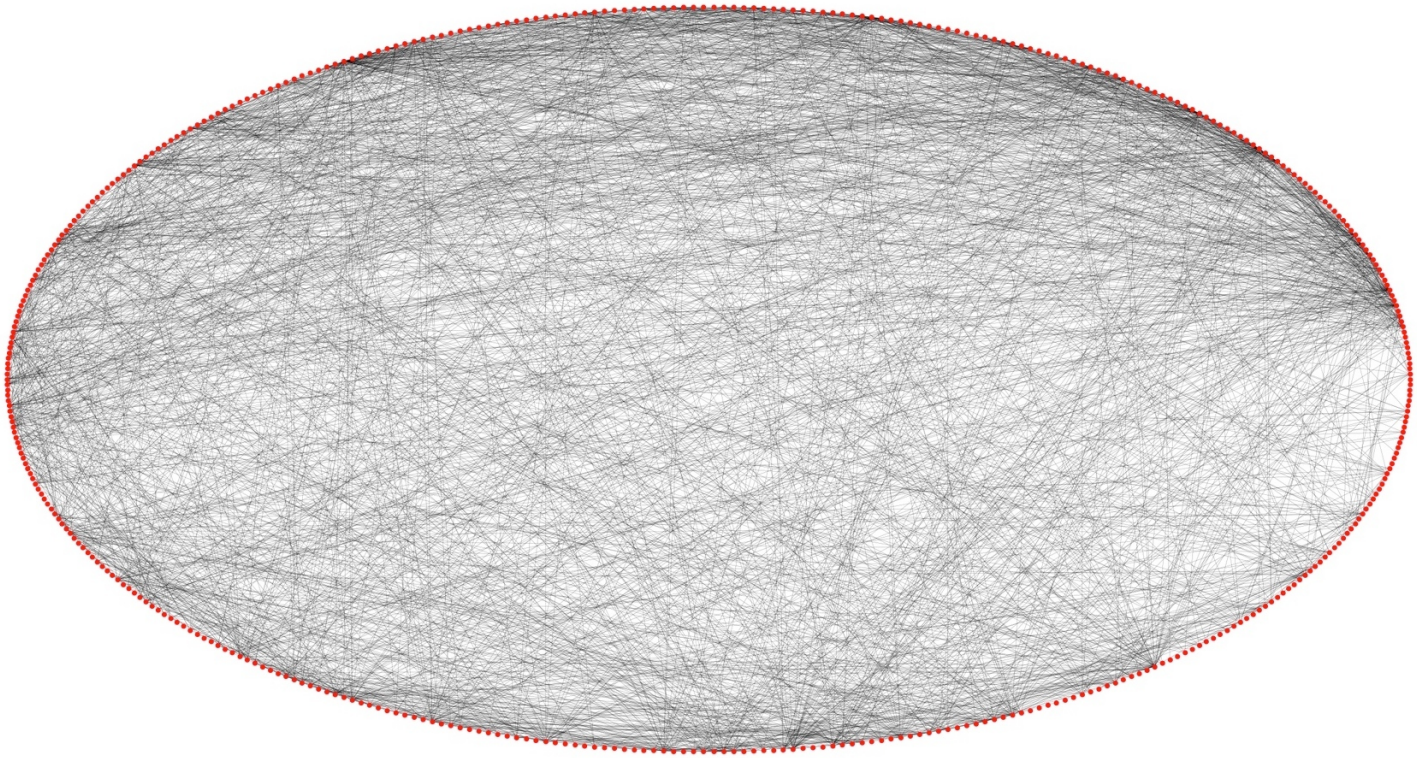
Составление и визуализация графа

На основе полученных данных мы построили несколько представлений графа: в виде матрицы смежности, в виде списка смежных вершин и в виде списка рёбер. Вершинами в данном графе выступали пользователи, а рёбрами соединялись пользователи, которые находились друг у друга в друзьях. полученный граф оказался разреженным(в нём было порядка 2300 рёбер из 97000 возможных). Далее мы решили визуализировать граф несмотря на его

размер для того, чтобы лучше представлять, как устроены взаимосвязи между пользователями. Для этого мы использовали библиотеку NetworkX. Мы построили несколько визуализаций: со случайным расположением вершин на плоскости и с расположением их по овальовидной форме.



Визуализация графа со случайным расположением вершин на плоскости



Визуализация графа с оваловидным упорядочением вершин на плоскости

Анализ графа

Как было сказано ранее, полученный граф оказался достаточно разреженным. Но это не главная проблема. Изначально планировалось проанализировать граф, используя алгоритм Каргера для поиска минимального разреза. Но это имеет смысл только для связных графов. Наш же граф получился не связным. Поэтому мы решили провести поиск компонент связности, базируясь на алгоритме DFS. В итоге в нашем графе оказалось порядка 65 компонент связности: одна большая (около 300 пользователей) и очень много маленьких (в основном это или изолированные вершины, или отдельные рёбра между парой вершин). Большая компонента связности состояла из студентов ФКН, остальные же являлись или заблокированными пользователями, или фейковыми аккаунтами, или парами людей с других факультетов, которые, судя по всему, нашли группу в рекомендациях и их заинтересовала тематика постов.
В конце мы попробовали применить алгоритм Карьера к нашей большой компоненте связности и получили, что размер минимального разреза составляет 8, что означает, что наша компоненты разбивается на 2 больших кластера пользователей, между которыми есть всего 8 связей(ребер).

Вывод

В процессе работы мы освоили много новых инструментов и библиотек: использовали запросы к VK API с помощью библиотеки request, строили визуализации графа пользователей, используя NetworkX, изучили и реализовали новые алгоритмы, а именно: алгоритм Каргера и алгоритм поиска компонент связности, основанный на поиске в глубину(DFS). Так же мы научились интерпретировать результаты работы алгоритмов с практической точки зрения.