Санкт-Петербургский Государственный Университет

Математико-механический факультет

Кафедра системного программирования

Курсовая работа на тему:

Выделение групп пользователей в социальных сетях

Никита Симонов, 361 группа

Научный руководитель:

Владимир Суворов

Санкт-Петербург

Май 2012

Оглавление

1. Введение

В последние годы все большей популярностью начинают пользоваться различные социальные сети, создаются новые сети с новыми целями и направлениями. С ростом популярности социальные сети собирают все больше данных о пользователях и связях между ними – главной составляющей подобных систем - которые представляют огромный интерес для анализа.

Так как связи между пользователями возникают не случайным образом, можно предположить, что если пользователи связаны между собой, то они либо имеют общие интересы, либо входят в одну общественную организацию, являются родственниками, коллегами по работе и т.п. Для анализа связей в социальных сетях обычно используется социальный граф пользователей: рассматриваются социальные отношения в терминах теории сетей, где пользователи являются вершинами графа, а соответствующие социальной сети отношения между пользователями рассматриваются как связи между вершинами.

Исходя из вышесказанного, ставится задача о выделении пользователей в группы на основе структуры полученного социального графа так, чтобы пользователи из одной группы имели общие параметры, связывающие их.

В данном документе описана курсовая работа, результатом которой является программный продукт, представляющий к рассмотрению пользователю его социальный граф и группы пользователей социальной сети для их последующего анализа.

2. Постановка задачи

В рамках данного проекта должно быть реализовано приложение, позволяющее для пользователя социальной сети выделить из его окружения группы пользователей связанных между собой. Таким образом, были поставлены следующие задачи:

- собрать информацию о пользователях из социальной сети;
- построить социальный граф пользователя;
- визуализировать социальный граф;
- разработать алгоритм поиска групп в социальном графе;
- отобразить полученные группы.

Актуальность данной работы обусловлена применимостью во многих областях, в задачах, требующих выделения целевой аудитории, как то: поиск кадров, таргетированная реклама и т.п. Кроме того, в ней используются социальные сети, содержащие в себе достаточный объем информации, представляющей интерес для анализа.

3. Области применения

Результат данной работы может быть применим к различным областям.

- В рамках таргетированной рекламы данная работа может использоваться для поиска целевой аудитории с целью вовлечения её в свои рекламные акции.
- Для поиска кадров кадровая служба может анализировать окружение пользователя с подходящими на целевую должность профессиональными данными и выделять из найденных групп ту, члены которой могут подходить по параметрам на эту должность.
- Для поиска экстремистски настроенных граждан достаточно проанализировать окружение нескольких представителей этого класса и получить группу потенциально опасных граждан.

4. Реализация

Реализацию приложения можно разделить на четыре этапа:

- сбор и хранение данных социальной сети;
- построение на основе собранных данных социального графа пользователя;
- выделение групп пользователей из полученного графа;
- визуализация графа и групп пользователей.

Для написания приложения выбран объектно-ориентированный язык программирования Java [1], являющийся кроссплатформенным и имеющим большое количество библиотек, расширяющих его функциональность и упрощающих разработку систем.

Далее более подробно будет рассмотрен каждый этап разработки приложения.

4.1 Сбор и хранение данных

Для решения поставленной задачи в качестве платформы для исследования была выбрана социальная сеть “ВКонтакте” [2], являющаяся на текущий момент самой посещаемой русскоязычной социальной сетью, насчитывающей на январь 2012 года и имеющей порядка 160 миллионов зарегистрированных пользователей и около 33 миллионов уникальных посетителей в день. Кроме того данная социальная сеть имеет хорошо документированный API (Application programming interface) [3], в том числе для standalone-приложений. Для решения данной задачи были использованы следующие методы API:

- запрос авторизации клиентских приложений;
- метод user.get для получения расширенной информации о пользователях;
- метод friends.get для получения списка id друзей пользователя.

Все полученные данные представляются в текстовом формате XML (eXtensible Markup Language). Для сериализации

данных в формате XML был использован фреймворк Simple [4] для языка Java.

Полученная информация записывается в базу данных, состоящую из двух таблиц. Первая таблица хранит в себе информацию о пользователях. Во второй таблице представляется бинарное отношение (дружба между пользователями).

4.2 Построение социального графа

Для работы с графами использовался фреймворк JUNG (Java Universal Network/Graph Framework) [5], представляющий собой программную библиотеку, обеспечивающую возможность использования общего и расширяемого языка для моделирования, анализа и визуализации данных, представленных в виде графа или сети. Он написан на языке программирования Java, что позволяет JUNG-приложениям использовать встроенные возможности Java API, а также другие подключаемые библиотеки Java. Сама архитектура JUNG разработана для поддержки представлений различных сущностей и их отношений, таких как ориентированные и неориентированные графы, мультиграфы, графы с параллельными ребрами и гиперграфы.

Для представления графа среди набора модулей фреймворка JUNG был выбран класс UndirectedSparseMultigraph, описывающий неориентированный разреженный мультиграф и поддерживающий основные методы работы с графом.

4.3 Выделение групп пользователей

Предполагается, что пользователи, входящие в одну группу, имеют общие интересы и общих друзей, тогда можно предположить, что связность пользователей в группе высокая. Самой высокой связностью неориентированного графа обладает клика. Клика – это полный подграф неориентированного графа, то есть подграф исходного графа, между всеми вершинами которого существуют ребра. Можно утверждать, что пользователи, входящие в одну клику, скорее всего, связаны

общими интересами, и тогда клика является кандидатом на осмысленную группу. Таким образом, возникает задача поиска всех клик в заданном графе.

Задача о клике имеет следующую формулировку: требуется определить, существует ли в заданном графе клика заданного размера (то есть состоящая из заданного количества вершин). Задача о клике относится к классу NP-полных задач. Одним из самых быстрых алгоритмов по поиску всех клик является алгоритм Брона-Кербоша, представляющий собой метод ветвей и границ. Реализация алгоритма в псевдокоде выглядит следующим образом:

```
BronKerbosch(R, P, X):
    if P and X are both empty:
        report R as a maximal clique choose a pivot vertex u in P ⋃ X
    for each vertex v in P \ N(u):
        BronKerbosch(R ⋃ {v}, P ⋂ N(v), X ⋂ N(v))
        P := P \ {v}
        X := X ⋃ {v}
```

В приведенном алгоритме используются следующие обозначения:

R – множество, содержащее полный подграф данного графа.

P – множество вершин, которые могут быть добавлены в R.

X – множество вершин, которые уже использовались для расширения R.

N(x) – функция, по вершине возвращающая множество вершин, являющихся соседними для нее.

В худшем случае данный алгоритм работает за $O(3^{n/3})$. Следовательно, для большого количества вершин алгоритм будет работать слишком долго, а, следовательно, появляется необходимость уменьшить начальный граф для поиска клик в нем. Было решено искать клики только в графе, содержащем только пользователя и его друзей, что называется “первым

кругом”, так как включение “второго круга” (друзья друзей) сильно увеличивает количество вершин в графе.

Далее на основе полученных клик образуются группы. Для этого последовательно совершается несколько итераций по добавлению соседних вершин к клике, если вершина связана с большей частью вершин принадлежащих текущей клике. Аналогично каждый раз после добавления вершин производится слияние клик, у которых большая часть вершин совпадает. Таким образом, группы расширяются, и уменьшается число повторов.

4.4 Визуализация графа и групп

Для отображения полученных результатов был также использован фреймворк JUNG, так как он предоставляет средства визуализации, упрощающие разработку инструментов для интерактивного изучения графа. Кроме того, среди поставляемых им возможностей существуют механизмы фильтрации, которые позволяют пользователям сосредоточить свое внимание и их алгоритмы на конкретных частях графа.

Средствами JUNG отображается граф, в котором вершины, представляющие друзей рассматриваемого пользователя социальной сети, отмечены красным цветом. Вершины, являющиеся друзьями друзей, отмечены сине-зеленым (cyan) цветом. При выборе группы из списка полученных, вершины входящие в нее выделяются синим цветом непосредственно на графе, а список входящих в группу пользователей отображается в левой части экрана, что позволяет пользователю оценивать полученные группы. Вариант представления результата изображен на рисунке 1.

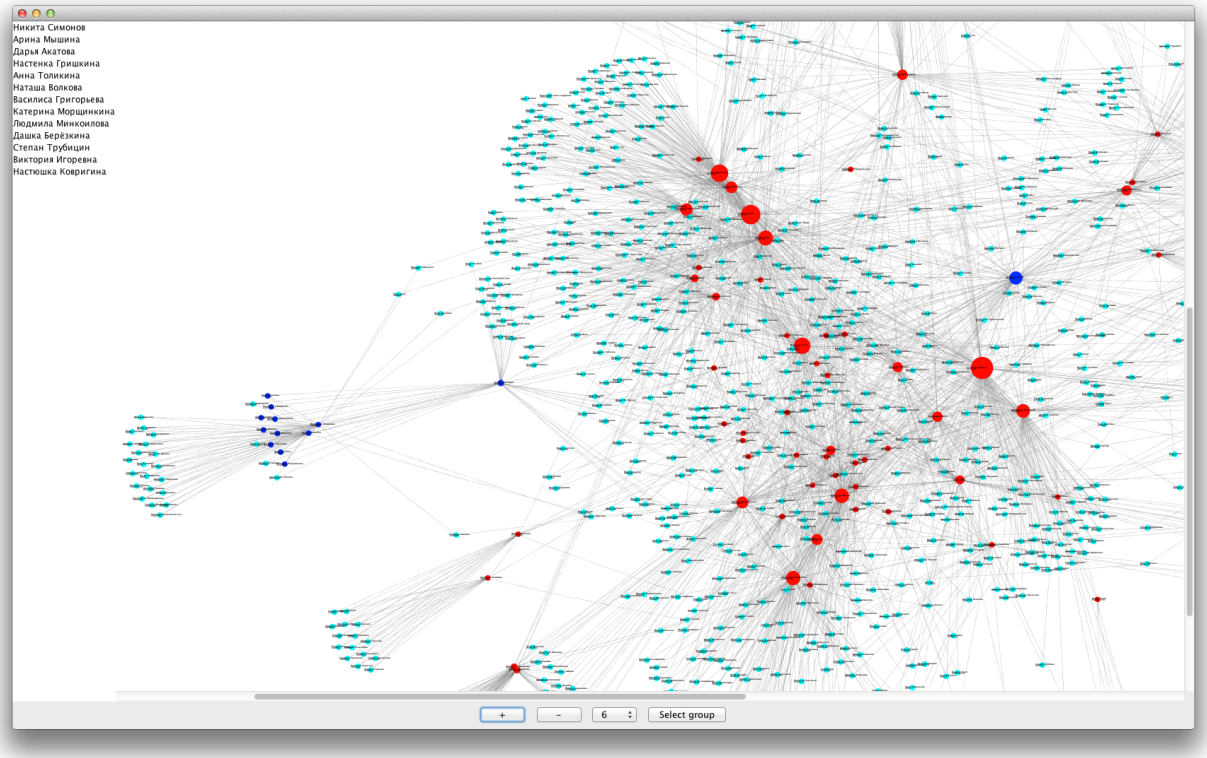

Рис. 1. Представление результата

5. Заключение

В ходе курсовой работы были изучены различные вспомогательные средства для визуализации графов, для работы с базой данных и социальной сетью, алгоритмы на графах. Кроме того в рамках данной работы разработано программное средство для выделения групп пользователей в социальных сетях.

Разработанное программное средство предоставляет возможность пользователю анализировать свой социальный граф и группы пользователей, связанные общими интересами, что достигается посредством визуального отображения социального графа и выделения пользователей на нем.

6. Дальнейшие планы

На данном этапе каждая полученная группа представляет из себя множество пользователей, лишь предлагаемых к рассмотрению и оценке пользователю приложению на степень обоснованности их включения в одну группу. На основе чего можно поставить задачу поиска общих параметров членов группы, которые могут их связывать. Для решения такой задачи следовало бы использовать информацию предоставленную пользователем о себе в рассматриваемой социальной сети (например, рассматривать всех членов группы и искать общие интересы или использовать группы, представленные в социальной сети, считая, что наиболее часто встречающийся интерес у пользователей из нашей группы является параметром, связывающим пользователей в группу).

В качестве еще одного направления дальнейшего развития может быть рассмотрена задача о расширении интересующей нас группы, то есть о возможности добавлении в нее других пользователей, связанных с пользователями данной группы и имеющих схожие параметры (например, перебирая членов целевой группы и, используя описанный в данной курсовой работе алгоритм, разыскивая среди его окружения группу с подходящими интересами, далее объединяя группы и продолжая расширение целевой группы).

7. Ссылки

[1] http://www.java.com/ru/ (официальный сайт языка программирования Java)

[2] http://vk.com (официальный сайт социальной сети ВКонтакте)

[3] http://vk.com/developers.php (документация API ВКонтакте)

[4] http://simple.sourceforge.net (официальный сайт фреймворка Simple)

[5] http://jung.sourceforge.net (сайт фреймворка JUNG)