Исследование данных и связей, в социальных сетях

М.И. Хотилин[a], А.В. Благов[a]

[a] *Самарский национальный исследовательский университет имени академика С.П. Королева, 443086, Московское шоссе, 34, Самара, Россия*

Аннотация

Данная работа посвящена анализу данных и связей в социальных сетях. Рассматривается подход представления социальной сети в виде графа. Исследованы и намечены к доработке алгоритмы отыскания сообществ и основных узлов («хабов»), являющихся аккаунтами, оказывающими наибольшее влияние на сообщества. Исследованы существующие программные среды по визуализации данных социальных сетей, разработан программный комплекс.

Ключевые слова: социальные сети; большие данные; граф; матрица смежности; SCAN-алгоритм; Gephi

1. Введение

За последнее десятилетие социальные сети стали играть огромную роль в жизни общества. Они, будучи предметом социализации людей, занимают одну из лидирующих позиций по производству «больших данных». Возможность выкладывать и делиться сообщениями, фотографиями, музыкой, видео с друзьями, а также создавать и проводить различные события, в том числе с целью продвижения бизнеса – всё это представляет собой колоссальный объем постоянно генерирующихся, устаревающих, обновляющихся данных.Большие объёмы данных, в том числе из социальных сетей, а также зависимости (связи) между ними необходимо представить в виде удобном для восприятия.

Зачастую, если речь идет об объектах представляющих собой сеть, например социальную, понятие визуализации данных тесно связано с понятием графов. Сеть, представленная в виде графа, проста для восприятия и дальнейшего анализа. Важной задачей является представление связей в социальных сетях для выявления различного рода зависимостей.

2. Сбор данных из социальной сети

Для представления социальной сети в виде графа может использоваться множество различных средств и инструментов. В рамках данной работы для решения этой задачи было использовано следующее: разработанное на языке C# приложение, позволяющее получить необходимые данные и выполнить их анализ; средство визуализации данных Gephi для представления в графической форме непосредственно самого построенного графа зависимостей (так называемого графа друзей пользователя).

Само программное средство визуально представляет собой форму авторизации, на которой вводятся логин пользователя и пароль учетной записи пользователя (рисунок 1).

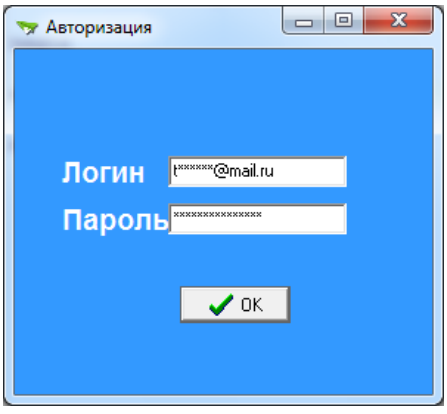

Рис. 1. Интерфейс программного средства.

После ввода логина и пароля, посредством открытого протокола авторизации OAuth версии 2.0 происходит авторизация пользователя в социальной сети и предоставление доступа к необходимой информации, а именно: к списку друзей пользователя, списку сообществ, фотографиям, сообщениям и т.д. В рамках данной работы нас интересовала возможность извлечь из социальной сети друзей пользователя, поэтому остальные пункты остались без внимания.

Каждый пользователь социальной сети имеет свой уникальный идентификатор, или иначе ID, что позволяет однозначно определить пользователя. Используя свойства встроенного API социальной сети Вконтакте, можно, зная IDпользователя, извлечь информацию о его друзьях, вплоть до N-ного уровня вложенности. Иными словами можно извлечь список друзей (N=1), друзей (N=2) и т.д. Нас интересовал список друзей до уровня вложенности N=2.

Список друзей, извлеченный из социальной сети и преобразованный, принимает вид текстового файла, содержащего IDпользователя, авторизовавшегося в социальной сети и далее в табличной форме ID пользователя и его имя (ФИО). Зная IDдруга пользователя, можно также узнать список и его друзей, что аналогично заносится в файл. Пример части выходного файла по пользователю и каждому из пользователей-друзей представлен на рисунке 2.

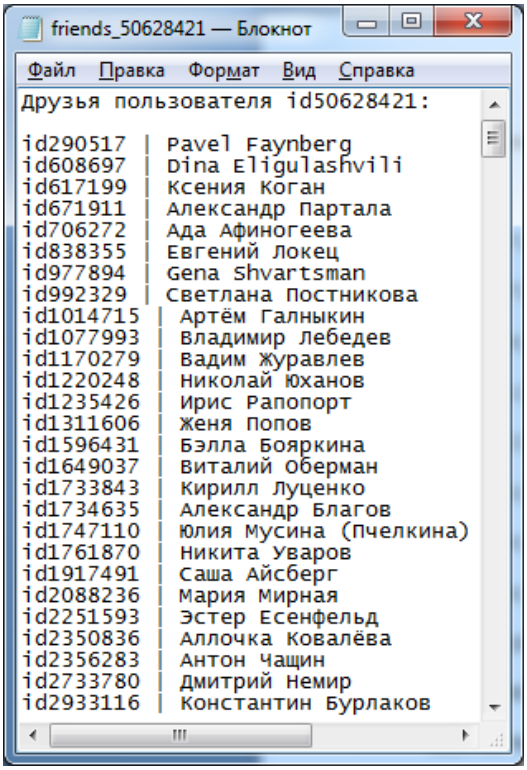

Рис. 2. Файл, содержащий информацию о друзьях пользователя.

Далее, путем конкатенации файлов, получается общий список всех друзей, по которому строится список всех друзей (размерности К), из которого организуется матрица смежности (размерности $K \times K$), по которой впоследствии строится граф зависимостей, посредством программного средства Gephi.

Матрица смежности, представляет собой матрицу размерности $K \times K$, содержащую по горизонтали и вертикали список друзей, а на пересечении строки и столбца стоит 0 или 1. Содержание ячейки матрицы-таблицы представляет собой 0, если пользователи не знакомы (не содержатся в списке общих друзей между пользователем и другом пользователя) и 1 иначе, если существуют отношение «дружба» между указанными пользователями. После построения, данная матрица сохраняется в формате .csv для дальнейшей загрузки в Gephi. Пример матрицы смежности приведен на рисунке 3.

	Air Sola	Ildar Khalitov	Igor Rytsa	Svetlana S	Alexey Sa	Anastasiy	Andrey M	Maksim R	Alexsand	Yuri Nagu
Air Sola	0	0	0	1	0	0	0	0	0	1
Ildar Khalitov	0	0	0	0	0	0	0	0	0	0
Igor Rytsarev	0	0	0	0	0	1	1	1	1	1
Svetlana Sukhanova	1	0	0	0	0	0	0	0	0	0
Alexey Satonin	0	0	0	0	0	0	0	1	0	0
Anastasiya Kireeva	0	0	1	0	0	0	1	1	1	1
Andrey Mukhataev	0	0	1	0	0	1	0	1	1	1
Maksim Raguzin	0	0	1	0	1	1	1	0	1	1
Alexsander Nagulov	0	0	1	0	0	1	1	1	0	1
Yuri Nagulov	1	0	1	0	0	1	1	1	1	0

Рис. 3. Матрица смежности графа друзей.

3. Построение графа, классификация вершин, нахождение наиболее значимых

Построенная на основе списка всех друзей матрица смежности будущего графа, загружается в программное средство Gephi, с целью дальнейшей визуализации графа зависимостей.

Gephi представляет собой, написанный на языке высокого уровня Java, программный продукт для сетевого анализа и визуализации данных[9].

Построенный Gehpi граф выглядит таким образом (рисунок 4):

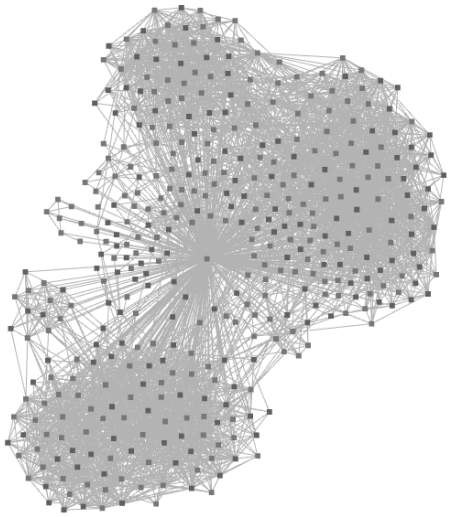

Рис. 4. Граф зависимостей пользователя.

В данном графе вершинами являются пользователи социальной сети, а ребрами – отношение «дружба» между пользователями.

Стоит заметить, что друзья друзей пользователя, не имеющие общих связей с пользователем, не интересовали нас в рамках данной работы, поэтому данные вершины-друзья друзей были удалены из графа.

Следующим этапом является классификация вершин графа. В работе предложена следующая классификация:

- core(ядро) – это вершина, содержащая в $\varepsilon -$ *окрестности*, по крайней мереμ вершин
- hub (хаб) – это отдельная вершина, соседи которой принадлежат двум или более различным кластерам;
- outlier (посторонний) – это отдельная вершина, все соседи которой принадлежат одному и тому же кластеру, или не принадлежат никакому кластеру [10].

Для осуществления подобной классификации используется SCAN алгоритм [7]. Надо отметить, что в SCAN алгоритме для кластеризации используется модифицированная метрика модулярности:

$$Q = \sum_{s=1}^{k} \left[\frac{l_s}{L} - \left(\frac{d_s}{2L} \right)^2 \right],$$

где L – количество ребер в графе, l_s - число ребер между вершинами в кластере s и d_s - сумма степеней вершин в кластере s.

Принцип работы SCAN алгоритма описан ниже.

Поиск начинается с начального посещения каждой вершины один раз [10], с целью нахождения структурно-связных кластеров, а затем посещения изолированных вершин, чтобы идентифицировать их (hub или outlier).

SCAN выполняет один проход сети и находит все структурно-связанные кластеры для заданного параметра. В начале все вершины помечены как неклассифицированные[10]. Алгоритм SCAN классифицирует каждую вершину либо как являющуюся членом кластера, либо как не являющуюся [10]. Для каждой вершины, которая еще не классифицирована, SCAN проверяет, является ли эта вершина ядром. Если вершина является ядром, новый кластер расширяется из этой вершины. В противном случае вершина помечается как не являющаяся членом кластера.

Чтобы найти новый кластер, SCAN начинается с произвольной ядраV и ищет все вершины, которые структурно-достижимы из V. Этого вполне достаточно, чтобы найти полный кластер, содержащий вершину V[10]. Генерируется новый IDкластера, который будет назначен всем найденным вершинам.

SCAN начинается, постановкой всех вершин в ε-окрестности вершины V в очередь. Для каждой вершины в очереди вычисляются все непосредственно достижимые вершины, и в очередь вставляются те вершины, которые до сих пор не классифицированы. Это повторяется до тех пор, пока очередь не опустеет[10].

Вершины, не являющиеся членами кластеров, могут быть дополнительно классифицированы как хабы или посторонние. Если отдельная вершина имеет ребра на два или более кластеров, она может быть классифицирована как хаб. В противном случае, это посторонний.

Отличительной особенностью является наличие параметров μ и ε, которые могут задаваться пользователем или экспертом. При этом нахождение оптимального значения данных параметров можно провести при помощи машинного обучения системы, используя определённые сегменты сети.

Поскольку Gephi является opensource платформой [9], одним из больших его преимуществ является возможность написания своих собственных модулей, реализующих различные алгоритмы. Таким образом, используя алгоритм из [7], был написал модуль, реализующий SCAN-алгоритм.

Результатом работы SCAN-алгоритма над построенным в Gephi графе, стал текстовый файл, содержащий список друзей пользователя и типизацию его как вершины графа (рисунок 5)

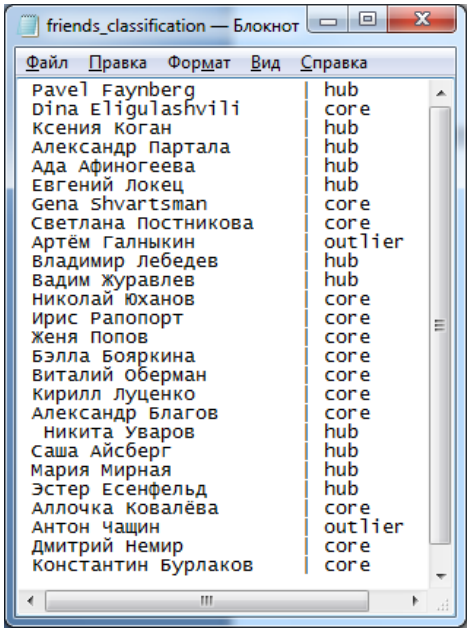

Рис. 5. Результат работы SCAN-алгоритма.

Стоит отметить, что SCAN-алгоритм имеет определенное ограничение на размерность используемого графа. На графах с высокой размерностью (N> 500) возникает погрешность в работе.

Наиболее вероятной причиной, объясняющей это является тот факт, что при большой размерности графа (порядка 10^4узлов), функционал модулярности, лежащий в основе SCAN, дает погрешность и возникают значительные вычислительные трудности, связанные с временными затратами при вычислении значения данного функционала.

Одним из путей разрешения данной проблемы является модификация алгоритма модулярности для параллельных вычислений [11]. Идея данной модификации состоит в том, чтобы разбить множества сообществ на подмножества между процессорами. Однако, здесь следует учесть, что для балансировки данные подмножества должны иметь примерно одинаковую сумму квадратов размеров сообщества.

Авторами поставлена задача создания распределенной модификации SCAN-алгоритма и реализации параллельного варианта алгоритма модулярности для графов сверхвысокой размерности.

4. Заключение

Социальные сети и связи в них являются предметом исследования в данной научно-исследовательской работе. Подход, основанный на представлении социальных сетей в виде графа, предоставляет возможность применять алгоритмы кластеризации графов высокой размерности. Описанные в работе алгоритмы позволяют производить классификацию сегментов социальной сети, а также находить элементы, представляющие наибольший интерес, например, пользователей, влияющих на все элементы одного сообщества. Данные алгоритмы планируются к доработке для последующего применения в решении практических задач отыскания сообществ в сегменте социальных сетей Самарской области.

Исследована среда разработки Gephi, позволяющая реализовать визуализацию социальных сетей, разработано программное средство, позволяющее представлять данные в необходимом для исследования виде.

Литература

[1] Tan, W. Social-network-sourced big data analytics/ W. Tan, M.W. Blake, I.Saleh., S. Dustdar //IEEE Internet Computing. – 2013. – №. 5. – P. 62-69.
[2] How people describe themselves on Twitter / K. Semertzidis, E. Pitoura, P. Tsaparas //Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks. – 2013. – P. 25-30.
[3] Big Data Instruments for Social Media Analysis / A. Blagov, I. Rytcarev, K. Strelkov, M. Khotilin//Proceedings of the 5th International Workshop on Computer Science and Engineering. – 2015. – P. 179-184.
[4] Иванов, П.Д. ТехнологииBigDataиразличныеметоды представлениябольшихданных / П.Д. Иванов, А.Г. Лопуховский// Инженерныйжурнал: наукаиинновации,вып. 9. – 2014.
[5] Gastner, M. T. Optimal design of spatial distribution networks / Michael T Gastner, M. E. J. Newman // Phys. Rev. E. – 2006. – T.74. – C. 016117.
[6] Newman, M. E. J. Finding and evaluating community structure in networks / M. E. J. Newman, M. Girvan // Phys. Rev. E. – 2004. – T. 69. – C. 026113.
[7] Scan: a structural clustering algorithm for networks / Xu X. et al. //Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM, 2007. – P. 824-833.
[8] Newman, M. E. J. Fast algorithm for detecting community structure in networks / M. E. J. Newman // Phys. Rev. E. – 2004. – T. 69. – C.066133.
[9] Открытая платформа по представлению данных в виде графов GEPHI [Электронный ресурс]. – URL: https://gephi.org
[10] Хотилин, М.И. Визуальное представление и кластерный анализ социальных сетей. / М.И. Хотилин, А.В. Благов // Сборник материалов Международной конференции и молодежной школы «Информационные технологии и нанотехнологии» - 2016. - С. 1067-1072.
[11] Drobyshevskiy M.D., Korshunov A.V., Turdakov D.Y. Parallel modularity computation for directed weighted graphs with overlapping communities //Труды института системного программирования РАН. – 2016. – Т. 28. – №. 6. – С. 153-170.