УДК 519.17, 519.237

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ, АНАЛИЗ ЭФФЕКТИВНОСТИ И ОЦЕНКА КАЧЕСТВА АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ ГРАФОВЫХ МОДЕЛЕЙ СОЦИАЛЬНЫХ СЕТЕЙ

М. С. Ионкин, М. В. Огнева

Ионкин Михаил Сергеевич, аспирант, Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, 410012, Россия, Саратов, Астраханская, 83, msionkin@gmail.com

Огнева Марина Валентиновна, кандидат физико-математических наук, заведующий кафедрой информатики и программирования, Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, 410012, Россия, Саратов, Астраханская, 83, ognevamv@gmail.com

Рассматривается задача поиска сообществ (кластеров) в неориентированных графах (задача кластеризации). Кластеризация — объединение в группы схожих объектов –- является одной из фундаментальных задач в области анализа данных. Список прикладных областей, где она применяется, широк: сегментация изображений, маркетинг, борьба с мошенничеством, прогнозирование, анализ текстов и многие другие. На сегодняшний момент не существует универсального эффективного решения данной задачи. Число методов объединения в группы, максимально похожих друг на друга объектов, довольно велико –- несколько десятков алгоритмов и еще больше их модификаций. В статье описаны алгоритмы решения данной задачи, приведены оценки их асимптотической сложности, традиционные метрики и функционалы качества, необходимые для оценки результатов их работы. Предложен вариант решения проблемы, противоположной проблеме resolution limit, — нахождение мелких относительно всего графа сообществ. Выполнена программная реализация алгоритма Smart Local Moving, который является улучшением известного алгоритма Louvain. Приведено экспериментальное сравнение эффективности рассмотренных алгоритмов для больших разреженных графов, содержащих несколько сотен тысяч вершин и ребер и соответствующих реальным данным сайтов YouTube, Amazon, Live Journal. Сравнительный анализ выполнялся на этих трех «обезличенных» наборах данных с заранее известным разделением на сообщества, а также на наборе данных со всей доступной информацией о вершинах (пользователях) из социальной сети Вконтакте. Выполнялось сравнение друг с другом сообществ, найденных разными алгоритмами на одном и том же наборе данных. Оценивались такие характеристики, как время выполнения алгоритмов, показатели модулярности и нормализованной взаимной информации.

Ключевые слова: кластеризация, поиск сообществ, графовые модели, анализ данных.

DOI: 10.18500/1816-9791-2017-17-4-441-451

ВВЕДЕНИЕ

Одной из основных задач анализа данных является задача кластеризации — выделение сообществ (кластеров) разных объектов. Например, с помощью нее можно находить группы пользователей с похожими предпочтениями, что, в свою очередь, помогает определить, какая информация будет для них наиболее интересна. Методы кластеризации применяются для снижения размерности в задачах машинного обучения, в маркетинговых исследованиях (выделение сегментов пользователей), в

области компьютерного зрения (для сегментации изображений, распознавания образов) и т. д. [1, 2].

Несмотря на актуальность задачи кластеризации, она до сих пор не решена окончательно. Существуют различные алгоритмы для решения данной проблемы [1, 3–6], но каждый имеет свои ограничения, преимущества и недостатки.

В связи с этим актуальной является задача оптимизации, анализа и сравнения алгоритмов и их оценок, выявления их преимуществ и недостатков, что, возможно, в дальнейшем приведет к созданию универсального алгоритма.

1. ОБЗОР АЛГОРИТМОВ

Для анализа были выбраны следующие алгоритмы: Infomap, Walktrap, Label Propagation, Fastgreedy, Edge Betweenness, Louvain и Smart Local Moving. Первые пять из них — довольно популярные алгоритмы, реализованные в различных библиотеках для анализа данных. Два последних были изобретены относительно недавно (2008 и 2013 гг.) и в соответствии с теоретическими оценками работают быстрее первых пяти.

Далее по тексту за n будем обозначать количество вершин в графе, а за m — количество ребер.

Infomap. Метод поиска сообществ, основанный на случайном блуждании (на каждом шаге процесса блуждающий объект находится в вершине и перемещается в другую вершину, выбранную случайным равновероятным образом из соседних вершин; последовательность посещенных вершин является марковской цепью, состояния которой являются вершинами графа). Каждое сообщество и каждая вершина имеет свой уникальный бинарный код, причем вершины из разных сообществ могут иметь одинаковый код. Авторы интерпретируют задачу выделения сообществ в графе как задачу кодирования пути, который пройдет «блуждатель», и пытаются минимизировать длину кода, получающегося во время прохода. Время выполнения этого алгоритма авторы в своей статье не приводят [7].

Walktrap. Данный метод, как и метод Infomap, основан на механизме случайных блужданий и использует идею о том, что короткие случайные блуждания не приводят к выходу из текущего сообщества [8]. На вершинах определенным образом вводится метрика и с помощью матрицы вероятностей перехода вершин между сообществами определяется, какие вершины нужно объединить в один кластер. Сложность такого метода в лучшем случае имеет оценку $O(n^2 \log n)$, а в худшем — $O(mn^2)$ [8].

Label Propagation. Метод основан на эвристике, согласно ей вершина относится к тому сообществу, что и большинство ее соседей. Изначально каждая вершина является одним сообществом. Далее на каждой итерации вершины перемешиваются случайным образом и по очереди обновляются метки каждой вершины в зависимости от меток ее соседей. Данные действия продолжаются до тех пор, пока происходят изменения [9]. Из-за рандомизации алгоритм при нескольких запусках может выдать разные результаты, то есть является в некотором смысле неустойчивым. Авторы метода предлагают агрегировать результаты простым пересечением кластеров. Метод прост и интуитивен, а также является вычислительно эффективным. Вычислительная сложность данного метода почти линейная и сравнима с $O(m)$ [9].

Fastgreedy. Данный метод заключается в жадной оптимизации модулярности (понятие модулярности будет рассмотрено ниже). Происходит инициализация сообществ в каждой вершине, а далее, перебирая пары смежных вершин, алгоритм пы-

тается максимизировать значение модулярности путем перемещения вершин в определенные сообщества. В данный алгоритм легко добавить априорную информацию о составе кластеров, например, если мы знаем, что какие-то конкретные вершины должны лежать в одном кластере для увеличения качества кластеризации. Метод содержит в себе все недостатки, свойственные жадным методам, и сходится не к самому лучшему решению. Часто алгоритм порождает одно большое сообщество с большинством вершин графа в нем и множество маленьких. Сложность метода $O(mn)$ [10].

Edge Betweenness. Метод разработан Гирваном и Ньюманом [11]. Алгоритм работает по следующей схеме: сначала подсчитываются коэффициенты центральности по посредничеству (количество кратчайших путей между всеми парами вершин, проходящих через данное ребро) на всех ребрах графа. Далее происходит поочередное удаление ребер с самым большим коэффициентом, а сообществами считаются оставшиеся компоненты связности. Процедура удаления связей завершается, когда достигает максимума модулярность результирующего разбиения. Данный алгоритм имеет множество модификаций, которые сводятся к подсчету других реберных коэффициентов либо замене модулярности другим схожим функционалом. Его главный недостаток — время работы, так как подсчет коэффициентов на ребрах является вычислительно сложной задачей. Сложность метода $O(m^2n)$ [11].

Louvain. Алгоритм основывается на жадной оптимизации модулярности и использует метод «local moving heuristic» (LMH, эвристика локального перемещения). LMH перемещает отдельные вершины графа из одного сообщества в другое так, чтобы каждое такое перемещение увеличивало значение модулярности. LMH проходит по вершинам графа в случайном порядке, а завершает свою работу, когда не останется таких вершин, перемещение которых увеличивает модулярность.

В начале алгоритма Louvain каждая вершина образует отдельное сообщество. Итерация алгоритма состоит из двух фаз. На первой фазе для каждой вершины выполняется метод LMH для всех смежных вершин таким образом, что перебор всех вершин продолжается до тех пор, пока происходит хотя бы одно перемещение вершины. На второй фазе происходит сжатие графа: вершины, входящие в одно сообщество, образуют новую супервершину (или метавершину) с соответствующим преобразованием ребер. Алгоритм останавливается, когда граф перестает изменяться, то есть когда модулярность достигла своего локального максимума. Отметим, что алгоритм зависит от порядка перебора вершин на его первом этапе. Эксперименты, которые провели авторы метода, позволяют говорить о том, что порядок не сильно влияет на результат работы метода (точнее, на значение функционала), но может значительно влиять на время выполнения, которое в среднем составляет $O(n \log n)$ [12].

Smart Local Moving. Данные об алгоритме Smart Local Moving (SLM) есть в работе [13], он представляет собой оптимизацию алгоритма Louvain. Так же, как и в алгоритме Louvain, в начале каждая вершина образует отдельное сообщество и далее для увеличения значения модулярности используется LMH. После этого идет шаг, который отсутствует в алгоритме Louvain: из каждого найденного сообщества строится подграф. Все вершины в одном подграфе снова распределяются в свои собственные сообщества, для этого подграфа запускается метод LMH. После того как каждый подграф был разбит на сообщества, алгоритм SLM выполняет сжатие графа, как и в алгоритме Louvain. В сжатом графе каждая вершина соответствует сообществу одного из подграфов, а все вершины, соответствующие сообществам из

одного подграфа, приписываются к одному и тому же сообществу в сжатом графе. Поэтому для каждого подграфа существует только одно сообщество в сжатом графе. Далее все эти шаги повторяются для получившегося сжатого графа, а весь алгоритм SLM остановится тогда, когда невозможно будет сжать очередной сокращенный граф. Оценки времени выполнения авторы алгоритма не приводят, но так как это оптимизация алгоритма Louvain, то величина оценки не должна превосходить $O(n \log n)$.

2. ОЦЕНКА КАЧЕСТВА КЛАСТЕРИЗАЦИИ

После работы алгоритма разбиения на сообщества необходимо оценить качество получившегося результата. Для этого используются функции потерь и функционалы качества. В зависимости от ситуации их можно разделить на две группы. В том случае, когда неизвестно истинное разбиение на сообщества, для оценки качества чаще всего используется значение функционала модулярности. А если истинное разбиение известно (такое возможно в случае модельных данных или для графов, в которых мы можем самостоятельно разметить сообщества), то возможно применение такой метрики, как нормализованная взаимная информация (NMI) [14].

Модулярность — это скалярная величина из отрезка $[-1; 1]$, которая количественно описывает неформальное определение структуры сообществ [11]:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i,\ C_j), \tag{1}$$

где A — матрица смежности графа, A_{ij} — (i, j) элемент матрицы, d_i — степень i-вершины графа, C_i — метка вершины (номер сообщества, к которому относится вершина), m — общее количество ребер в графе, $\delta(C_i, C_j)$ — дельта-функция: равна единице, если $C_i = C_j$, иначе — нулю.

Модулярность достаточно просто интерпретируется. Она показывает, насколько при заданном разбиении графа на группы плотность внутригрупповых связей больше плотности межгрупповых связей. Модулярность возможно эффективно пересчитывать при небольших изменениях в сообществах. Однако функционал не является непрерывным, и задача его оптимизации — дискретная, поэтому для поиска глобального оптимума используют приближенные схемы.

Нормализованная взаимная информация — метрика, основанная на теории информации и использующая идею о том, что если одно разбиение похоже на другое, то необходимо малое количество информации, чтобы восстановить одно разбиение из другого [15].

Рассмотрим два разбиения графа на сообщества x_i и y_i, где i — это номер вершины, а x_i и y_i — метки сообществ из разбиений. Предполагается, что метки x и y являются значениями двух случайных величин X и Y, которые имеют совместное распределение $P(x, y)$. Нормализованная взаимная информация определяется по формуле 2 [15]:

$$I_{norm} = \frac{I(X, Y)}{\sqrt{H\,(X)\,H(Y)}}, \tag{2}$$

$$I\,(X; Y) = H\,(X) - H\,(X|Y) = \sum_{x \epsilon X} \sum_{y \epsilon Y} p\,(x, y) \log \frac{p(x, y)}{p(x)p(y)}, \tag{3}$$

$$H(X) = -\sum_{k=-K}^{K} p_k \log p_k, \tag{4}$$

$$H(X|Y) = -\sum_{x \epsilon X} \sum_{y \epsilon Y} p(x,y) \log p(x,y), \tag{5}$$

где формула (3) — взаимная информация, (4) — энтропия случайной величины X, (5) — условная энтропия случайной величины X при наблюдении случайной величины Y.

Область значений NMI лежит в отрезке $[0;1]$. Если значение близко к нулю, то это означает, что два разбиения значительно отличаются друг от друга, а если величина равна единице, то два разбиения полностью совпадают. Величина взаимной информации не зависит от перестановок меток кластеров (разбиений) и для ее вычисления не делается никаких предположений относительно структуры кластера. Поэтому она может использоваться для сравнения результатов работы различных алгоритмов поиска сообществ в графах. К недостаткам данной метрики можно отнести то, что ее вычисление требует наличия заранее известных классов (ground truth classes) разбиения, которые практически никогда не доступны в реальных бизнес-задачах или которые могут быть размечены вручную, что является трудоемкой задачей.

3. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Проблема, обратная к resolution limit проблеме. Один из самых известных на данный момент алгоритмов поиска сообществ Louvain использует в своей основе модулярность. Обычно алгоритмы, использующие модулярность, не замечают мелких (относительно всего графа) сообществ и объединяют их в одно сообщество, что часто не является верным с точки зрения значения получающихся сообществ. Эта проблема получила название resolution limit. Существует несколько работ, посвященных исследованию этой проблемы и способам ее решения (например, [16,17]). Но в ходе исследования, проведенного авторами данной работы, выяснилось, что помимо обозначенной выше проблемы, на небольших объемах данных алгоритм Louvain делит их на слишком мелкие сообщества, т. е. возникает проблема, которая является прямо противоположной проблеме resolution limit. Для ее решения было предложено учитывать веса вершин графа при подсчете модулярности во время работы алгоритма Louvain, для чего в формуле подсчета модулярности вместо матрицы смежности была использована матрица весов. Был проведен эксперимент, в результате которого, расставив веса вершин у графа определенным образом, удалось сократить количество кластеров в получающемся разбиении, что позволило улучшить качество разбиения.

Сравнение и анализ алгоритмов. Для сравнения использовались традиционные алгоритмы, рассмотренные выше, и самостоятельно выполненная реализация алгоритма SLM.

Для анализа качества получающихся разбиений были выбраны наборы данных, у которых известны их ground-truth сообщества, то есть сообщества, сформированные по какому-либо признаку (например, группа в социальной сети, в которой состоят поклонники какого-нибудь артиста).

Были взяты общедоступные данные трех сайтов (YouTube, Amazon и Live Journal) [18], которые представляют собой большие разреженные графы. Все дан-

ные «обезличены» и представляют собой набор идентификаторов. Описание данных приведено в табл. 1.

Было проведено сравнение результатов работы алгоритмов по их времени выполнения и по значению модулярности для каждого набора данных. Кроме того, с помощью метрики NMI было проведено сравнение получившихся разбиений с ground-truth-сообществами. Полученные результаты представлены в табл. 2 (в ней не представлена информация по алгоритмам, время работы которых было больше суток).

Таблица 1 / Table 1

Описание наборов данных для анализа алгоритмов

Description of data sets for analysis of algorithms

Набор данных	Описание	Количество вершин	Количество ребер
YouTube	Вершины — пользователи. Ребро между двумя пользователями означает, что они подписаны друг на друга. Ground-truth-сообщество — группа пользователей по интересам	52 675	318 432
Amazon	Вершины — продукты. Ребро между двумя продуктами означает, что они часто покупаются вместе. Ground-truth-сообщество — продукты одной категории (например, ноутбуки)	317 194	872 935
Live Journal	Вершины — пользователи. Ребро между двумя пользователями означает, что они подписаны друг на друга. Ground-truth-сообщество — группа пользователей по интересам.	1 147 948	16 899 734

Таблица 2 / Table 2

Результаты работы алгоритмов / Results of work of algorithms

Алгоритм и оценка времени выполнения	Набор данных	Время выполнения, с	Модулярность	NMI
Label Propagation, $O(m+n)$	YouTube	4.1	0.5833	0.1342
	Amazon	38.74	0.7884	0.5451
	Live Journal	163	0.7124	0.159
Fastgreedy, $O(n\log^2 n)$	YouTube	280	0.5814	0.6885
	Amazon	1249	0.8771	0.6069
Infomap, $O(n(m+n))$	YouTube	4007.53	0.5785	0.1908
Walktrap, $O(n^2\log n)$	YouTube	760.18	0.5594	0.1572
	Amazon	1839.7	0.8543	0.4878
Louvain, $O(n\log n)$	YouTube	6	0.6676	0.1339
	Amazon	7	0.9348	0.3972
	Live Journal	240	0.7367	0.1812
SLM, $O(n\log n)$	YouTube	18	0.6957	0.1321
	Amazon	21	0.9403	0.4046
	Live Journal	818	0.7570	0.1775

Из табл. 2 видно, что время выполнения всех алгоритмов совпало с теоретическими оценками их времени выполнения.

Все алгоритмы, кроме Louvain и SLM, дают примерно одинаковые значения модулярности от 0.55 до 0.58 для данных YouTube и от 0.78 до 0.87 для данных Amazon. При этом алгоритмы Louvain и SLM в среднем дают модулярность выше на 0.1 пункта. Известно, что графы с более высокой модулярностью имеют более плотные соединения между вершинами внутри сообществ и менее плотные соединения между вершинами из разных сообществ. Это означает, что алгоритмы Louvain и SLM позволяют производить разбиение графа на сообщества с более плотной структурой, а это особенно важно для больших графов, состоящих из тысяч вершин и ребер.

Теперь обратимся к метрике NMI. Видно, что на наборе данных YouTube и Live Journal все алгоритмы, кроме Fastgreedy, принимают значения от 0.13 до 0.19. Это говорит о том, что эти алгоритмы делят графы на сообщества, мало совпадающие с группами, которые создают сами пользователи. Это можно объяснить тем, что алгоритмы работают, основываясь на структуре графа, на связях между вершинами (в данных двух случаях — на дружеских связях между пользователями), а группы создаются на основе какого-либо признака (например, группа программистов на языке Python), который в большинстве случаев не связан с отношением, заданным на ребрах графа (здесь — не связан с дружбой между пользователями). Тем не менее алгоритму Fastgreedy удается выполнить разбиение более близкое к ground-truth сообществам: на данных YouTube его значение NMI равно 0.68. Примерно так же обстоит дело со значениями NMI для данных Amazon. Все алгоритмы показали средний результат от 0.39 (у Louvain) до 0.6 (у Fastgreedy). То есть, основываясь на частоте покупок одного товара с другим, алгоритм распределил товары по группам наиболее часто покупаемых вместе товаров, и эти группы в среднем совпадают с реальными категориями товаров.

Так как у некоторых алгоритмов были получены примерно одинаковые значения модулярности и NMI, то необходимо выяснить, являются ли разбиения одинаковыми или это другие разбиения, но с похожей модулярностью. Для этого было проведено сравнение разбиений алгоритмов между собой с помощью метрики NMI. Это показало, например, что алгоритмы Louvain и SLM, а также Walktrap и Label Propagation находят наиболее похожие разбиения независимо от набора данных.

Анализ, который проводится на «обезличенных» данных, не позволяет понять, какой смысл несут в себе получившиеся разбиения. Поэтому было решено использовать данные социальной сети «ВКонтакте» (далее по тексту — набор данных Vk). Для этого был написан модуль для загрузки и предварительной обработки информации о пользователях данной социальной сети.

Рассматриваемый граф является моделью отношения «является другом» среди пользователей социальной сети. Был взят один пользователь социальной сети и список всех его друзей и построен неориентированный граф, в котором вершинами являются друзья этого пользователя, и между двумя вершинами проводится ребро, если эти два человека являются друзьями. Далее получившийся граф был вручную поделен на ground-truth-сообщества. Был проведен анализ результатов работы алгоритмов на этом наборе данных с помощью модулярности и NMI (табл. 3).

Таблица 3 / Table 3

Результаты работы алгоритмов для набора данных Vk
The results of the algorithms for the Vk data set

Алгоритм и оценка времени выполнения	Модулярность	NMI
Label Propagation, $O(m+n)$	0.1079	0.3219
Fastgreedy, $O(n\log^2 n)$	0.4718	0.0899
Edge Betweenness, $O(m^2 n)$	0.3859	0.2439
Infomap, $O(n(m+n))$	0.4899	0.1115
Walktrap, $O(n^2 \log n)$	0.4587	0.1319
Louvain, $O(n \log n)$	0.5144	0.6297
SLM, $O(n \log n)$	0.5523	0.8965

Из табл. 3 видно, что самые высокие показатели модулярности у алгоритмов Infomap, Louvain и SLM.

Согласно метрике NMI алгоритмы Lovain и SLM смогли достаточно точно определить социальные группы среди друзей пользователя, совпадающие с нашими ground-truth-сообществами, а наиболее похожими получились разбиения у алгоритмов Walktrap, Fastgreedy, Edge Betweenness и Infomap, но их разбиения довольно сильно отличаются от наших ground-truth-сообществ.

Таким образом, можно сделать вывод, что, основываясь только на знании о структуре графа, алгоритмы способны разбивать его на сообщества, содержащие информацию, которая не была доступна до начала эксперимента ни исследователю, ни самим алгоритмам.

ЗАКЛЮЧЕНИЕ

В работе проведен анализ наиболее популярных алгоритмов поиска сообществ в графах, изучены особенности их выполнения и свойства, дано краткое описание их работы. Предложен вариант решения проблемы, противоположной проблеме resolution limit — нахождение мелких относительно всего графа сообществ. Для этого учитывались веса вершин графа, что помогло подсчитывать модулярность более точно и, таким образом, производить более качественное разделение графа на сообщества.

Для того, чтобы выполнить анализ выбранных алгоритмов в единой среде, на языке Python был реализован алгоритм Smart Local Moving, который является улучшением алгоритма Louvain в максимизации модулярности.

Анализ проводился на трех «обезличенных» наборах данных с ground-truth-сообществами, содержащими несколько сотен тысяч вершин и ребер, представляющих собой большие разреженные графы, и на данных, взятых из социальной сети «ВКонтакте» и содержащих всю доступную информацию о вершинах. Оценивались такие характеристики, как время выполнения алгоритмов, показатели модулярности и нормализованной взаимной информации (NMI). По времени выполнения и максимизации значения модулярности явными лидерами являются алгоритмы Louvain и Smart Local Moving, однако они не всегда хорошо находят ground-truth-сообщества. С этой задачей в двух тестах хорошо справился алгоритм Fastgreedy, однако он не

смог найти «правильную» структуру для данных социальной сети «ВКонтакте». Таким образом, только при тщательном подборе алгоритма к соответствующей задаче можно извлекать ценную информацию из получающихся разбиений графа.

Библиографический список

1. *Aggarwal C. C., Charu C., Reddy C. K.* Data clustering. Algorithms and applications. N. Y. : CRC Press, 2014. 652 p.
2. *Jain A. K., Murty M. N., Flynn P. J.* Data clustering: a review // ACM Computing Surveys. 1999. Vol. 31, iss. 3. P. 264–323. DOI: 10.1145/331499.331504.
3. *Newman M. E. J.* Detecting community structure in networks // The European Physical Journal B – Condensed Matter and Complex Systems. 2004. Vol. 38, iss. 2. P. 321–330. DOI: 10.1140/epjb/e2004-00124-y.
4. *Leskovec J., Rajaraman A., Ullman J.* Mining of massive datasets. 2nd ed. Cambridge Univ. Press, 2014. 511 p.
5. *Fortunato S.* Community detection in graphs // Phys. Rep. 2010. Vol. 486, iss. 3–5. P. 75–174. DOI: 10.1016/j.physrep.2009.11.002.
6. Информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. URL: http://www.machinelearning.ru (дата обращения: 12.02.2017).
7. *Rosvall M., Axelsson D., Bergstrom C. T.* The map equation // The European Physical Journal – Special Topics. 2009. Vol. 178, iss. 1. P. 13–23. DOI: 10.1140/epjst/e2010-01179-1.
8. *Pons P., Latapy M.* Computing communities in large networks using random walks // Computer and Information Sciences – ISCIS 2005. 2005. P. 284–293. DOI: 10.1007/11569596_31.
9. *Raghavan U. N., Albert R., Kumara S.* Near linear time algorithm to detect community structures in large-scale networks // Phys. Rev. E. 2007. Vol. 76, iss. 3. P. 036106. DOI: 10.1103/PhysRevE.76.036106.
10. *Clauset A., Newman M. E. J., Moore C.* Finding community structure in very large networks // Phys. Rev. E. 2004. Vol. 70, iss. 6. P. 066111. DOI: 10.1103/PhysRevE.70.066111.
11. *Girvan M., Newman M. E. J.* Community structure in social and biological networks // Proc. National Academy of Sciences. 2002. Vol. 99, № 12. P. 7821–7826. DOI: 10.1073/pnas.122653799.
12. *Blondel V. D., Guillaume J., Lambiotte R., Lefebvre E.* Fast unfolding of communities in large networks // Journal of Statistical Mechanics : Theory and Experiment. 2008. Vol. 2008, № 10. P. P10008. DOI: 10.1088/1742-5468/2008/10/P10008.
13. *Waltman L., Eck N. J.* A smart local moving algorithm for large–scale modularity–based community detection // The European Physical Journal B. 2013. Vol. 86, iss. 11. P. 471. DOI: 10.1140/epjb/e2013-40829-0.
14. *Romano S., Bailey J., Nguyen V., Verspoor K.* Standardized mutual information for clustering comparisons: one step further in adjustment for chance // Proc. 31st Intern. Conf. on Machine Learning. Beijing, China : PMLR, 2014. Vol. 32, № 2. P. 1143–1151. URL: http://proceedings.mlr.press/v32/romano14.pdf (дата обращения: 25.04.2017).
15. *Хайкин С.* Нейронные сети : полный курс. М. : ИД «Вильямс», 2006. 1104 с.
16. *Fortunato S., Barthelemy M.* Resolution limit in community detection // Proc. National Academy of Sciences. 2007. № 104. P. 36–41. DOI: 10.1073/pnas.0605965104.

17. *Traag V. A., Dooren P. V., Nesterov Y.* Narrow scope for resolution-limit-free community detection // Phys. Rev. E. 2011. Vol. 84, iss. 1. P. 016114. DOI: 10.1103/PhysRevE.84.016114.
18. Stanford Large Network Dataset Collection. URL: https://snap.stanford.edu/data (дата обращения: 25.04.2017).

Образец для цитирования:
Ионкин М. С., Огнева М. В. Программная реализация, анализ эффективности и оценка качества алгоритмов кластеризации графовых моделей социальных сетей // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2017. Т. 17, вып. 4. С. 441–451. DOI: 10.18500/1816-9791-2017-17-4-441-451.

Implementation, Efficiency Analysis and Quality Evaluation of Clustering Algorithms for Graph Models of Social Networks

M. S. Ionkin, M. V. Ogneva

Michael S. Ionkin, orcid.org/0000-0002-4726-8245, Saratov State University, 83, Astrakhanskaya Str., Saratov, Russia, 410012, msionkin@gmail.com
Marina V. Ogneva, orcid.org/0000-0002-9828-7681, Saratov State University, 83, Astrakhanskaya Str., Saratov, Russia, 410012, ognevamv@gmail.com

The article deals with the community detection problem (the clustering problem) for undirected graphs. The clustering (grouping together of similar objects) is one of the fundamental tasks in the data analysis. This task is applied in a wide range of areas: image segmentation, marketing, anti-fraud, forecasting, text analysis and much more. At the moment, there is no universal and effective solution of this problem. There are several dozens of methods and there are many modifications of them which group objects that are as similar as possible to each other. The article describes algorithms for solving this task, presents their asymptotic complexity estimates, traditional metrics and quality functionals needed to evaluate the results of their work. The authors propose a solution to the problem which is the opposite of the resolution limit problem (algorithms find communities that are quite small in relation to the entire graph). The authors implemented the Smart Local Moving algorithm which is an improvement of the well-known Louvain algorithm. Performed an experimental comparison of the considered algorithms efficiency on large sparse graphs containing several hundreds of thousands of vertices and edges which corresponding to real data from YouTube, Amazon, Live Journal. The comparative analysis was performed on these three “impersonal” data sets with a previously known division into communities (ground-truth communities), as well as on a data set with all available information about the vertices (users) from the social network “Vkontakte”. The communities found by different algorithms on the same data set were also compared with each other. The authors examined such characteristics as the execution time of algorithms, values of modularity and normalized mutual information.

Key words: clustering, community detection, graph models, data analysis.

References

1. Aggarwal C. C., Charu C., Reddy C. K. *Data clustering. Algorithms and applications*. New York, CRC Зкуыы, 2014. 652 p.
2. Jain A. K., Murty M. N., Flynn P. J. Data clustering: a review. *ACM Computing Surveys*, 1999, vol. 31, no. 3, pp. 264–323. DOI: 10.1145/331499.331504.

3. Newman M. E. J. Detecting community structure in networks. *The European Physical Journal B – Condensed Matter and Complex Systems*, 2004, vol. 38, no. 2, pp. 321–330. DOI: 10.1140/epjb/e2004-00124-y.
4. Leskovec J., Rajaraman A., Ullman J. *Mining of massive datasets*. 2nd ed. Cambridge Univ. Press, 2014. 511 p.
5. Fortunato S. Community detection in graphs. *Physics Reports*, 2010, vol. 486, iss. 3, pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002.
6. *Information and analytical resource dedicated to machine learning, pattern recognition and intelligent data analysis*. Available at: http://www.machinelearning.ru (accessed 12 February, 2017) (in Russian).
7. Rosvall M., Axelsson D., Bergstrom C. T. The map equation. *The European Physical Journal – Special Topics*, 2009, vol. 178,iss. 1, pp. 13–23. DOI: 10.1140/epjst/e2010-01179-1.
8. Pons P., Latapy M. Computing communities in large networks using random walks. *Computer and Information Sciences – ISCIS 2005*, 2005, pp. 284–293. DOI: 10.1007/11569596_31.
9. Raghavan U. N., Albert R., Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 2007, vol. 76, iss. 3, pp. 036106. DOI: 10.1103/PhysRevE.76.036106.
10. Clauset A., Newman M. E. J., Moore C. Finding community structure in very large networks. *Phys. Rev. E*, 2004, vol. 70, iss. 6, pp. 066111. DOI: 10.1103/PhysRevE.70.066111.
11. Girvan M., Newman M. E. J. Community structure in social and biological networks. *Proc. National Academy of Sciences*, 2002, vol. 99, no. 12, pp. 7821–7826. DOI: 10.1073/pnas.122653799.
12. Blondel V. D., Guillaume J., Lambiotte R., Lefebvre E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008, vol. 2008, no. 10, pp. P10008. DOI: 10.1088/1742-5468/2008/10/P10008.
13. Waltman L., Eck N. J. A smart local moving algorithm for large–scale modularity–based community detection. *The European Physical Journal B*, 2013, vol. 86, no. 11, pp. 471. DOI: 10.1140/epjb/e2013-40829-0.
14. Romano S., Bailey J., Nguyen V., Verspoor K. Standardized mutual information for clustering comparisons : one step further in adjustment for chance. *Proc. 31st International Conference on Machine Learning*. Beijing, China, PMLR, 2014, vol. 32, no. 2, pp. 1143–1151. Available at: http://proceedings.mlr.press/v32/romano14.pdf (accessed 25 April, 2017).
15. Haykin S. *Neural Networks: A Comprehensive Foundation*. Singapore, Pearson Education (Singapore) Pte Ltd., 1998. 842 p. (Russ. ed. : Moscow, Publ. House Williams, 2006. 1104 p.)
16. Fortunato S., Barthelemy M. Resolution limit in community detection. *Proc. National Academy of Sciences*, 2007, no. 104, pp. 36–41. DOI: 10.1073/pnas.0605965104.
17. Traag V. A., Dooren P. V., Nesterov Y. Narrow scope for resolution-limit-free community detection. *Phys. Rev. E*, 2011, vol. 84, iss. 1, pp. 016114. DOI: 10.1103/PhysRevE.84.016114.
18. *Stanford Large Network Dataset Collection*. Available at: https://snap.stanford.edu/data (accessed 25 April, 2017).

Cite this article as:
Ionkin M. S., Ogneva M. V. Implementation, Efficiency Analysis and Quality Evaluation of Clustering Algorithms for Graph Models of Social Networks. *Izv. Saratov Univ. (N. S.), Ser. Math. Mech. Inform.*, 2017, vol. 17, iss. 4, pp. 441–451 (in Russian). DOI: 10.18500/1816-9791-2017-17-4-441-451.