УДК 004.021

БЕНЧМАРКИНГ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ ГРАФОВ ДЛЯ ЗАДАЧ ПРИНЯТИЯ РЕШЕНИЙ

Фирсов М.И., Целых А.А.

ФГАОУ ВО «Южный федеральный университет», Ростов-на-Дону, e-mail: tselykh@sfedu.ru

В данной статье методом бенчмаркинга на эталонных графовых наборах данных сравниваются распространенные алгоритмы кластеризации графов с использованием таких метрик качества, как F-мера, модулярность, нормализованный и относительный разрезы графа. Содержится описание проведенного эксперимента, в котором шестью алгоритмами были обработаны 18 эталонных графов, разделенных на подгруппы на основе средней степени графа. Отдельно рассматривались графы, для которых известно истинное разбиение на кластеры. Представлены несколько подходов к решению отдельно взятой задачи кластеризации с учетом специфики задач принятия решений, которые часто характеризуются неоднозначностью результатов кластеризации. Рассматриваемые алгоритмы реализованы в библиотеках Lightcluster и Igraph на языке программирования Python. По результатам эксперимента выявлены наиболее результативные и универсальные алгоритмы, сделаны содержательные выводы.

Ключевые слова: кластеризация графов, бенчмаркинг, сравнение алгоритмов, принятие решений

BENCHMARKING GRAPH CLUSTERING ALGORITHMS FOR DECISION SUPPORT PROBLEMS

Firsov M.I., Tselykh A.A.

Southern Federal Unuiversity, Rostov-on-Don, e-mail: tselykh@sfedu.ru

In this paper, we use benchmarking method on reference graph datasets in order to evaluate common graph clustering algorithms using such quality measures as F-measure, modularity, normalized cut, and ratio cut. We describe an experiment where we applied six algorithms to process 18 reference graphs divided into subgroups based on the average degree of the graph. We individually consider graphs with a known true partitioning into clusters. We consider several approaches to a certain clustering problem taking into account the specifics of decision support tasks often described as ambiguous. Algorithms under estimation are implemented in Lightcluster and Igraph libraries for Python. Summarizing the results of experiment we identify the most effective and versatile algorithms and provide a meaningful conclusion.

Keywords: graph clustering, benchmarking, algorithm evaluation, decision making

Большой практический и академический интерес к методам принятия управленческих решений активно способствовал становлению и развитию теории и практики кластерного анализа на графах [1]. Широкий и постоянно растущий спектр прикладных задач сформировал устойчивый запрос на алгоритмы для взвешенных, ориентированных, знаковых и нечетких графов, а также графов причинно-следственных связей, отличительными характеристиками которых являются большая размерность и наличие циклов. В данной работе предпринята попытка методом бенчмаркинга – эталонного сопоставления на известных наборах данных [2] – выявить алгоритмы, эффективные с учетом современной динамики теоретико-графовых моделей данных

Для понятия «кластер», «когезионная подгруппа» или «сообщество» существует ряд различных и иногда противоречивых определений. Объединяет их понятие группы тесно связанных между собой узлов, которая, в свою очередь, слабо связана с остальной частью сети. Любая функция качества для назначения узлов в сообщества следует простому принципу: объединить все, что связано, и не разделять то, что связано не явно.

Для решения этой задачи ряд алгоритмов требует указания на входе параметров, таких как количество кластеров и разного рода пороговых значений. Другие алгоритмы не используют входных параметров либо вычисляют их самостоятельно.

Ряд существующих методов позволяет формировать пересекающиеся кластеры, которые допускают вхождение одной вершины сразу в несколько сообществ. Отдельные алгоритмы работают с понятием «шума» – точками, которые не принадлежат ни одному из кластеров. Многие алгоритмы кластеризации допускают существование аутлаеров – кластеров с единственной вершиной.

Растущий интерес к парадигме больших данных подчеркивает значение вычислительной сложности и времени работы алгоритмов.

В нашем эксперименте мы постарались охватить все типы алгоритмов, перечисленные выше.

Материалы и методы исследования

Для эксперимента были взяты 18 графовых наборов данных с различным соотношением числа вершин и дуг, а также различным распределением весов на дугах.

Для части графов известно истинное разбиение на кластеры, для другой части – нет. Графы с известным разбиением на кластеры были выделены в отдельную группу. Оставшиеся графы были разбиты на две равные группы: в одну попали графы, средняя степень которых не превышает двух, в другую – графы с более плотными связями между вершинами. Характеристики графов представлены в таблице.

После формирования эталонных графовых наборов данных были определены алгоритмы кластеризации для эксперимента.

Часть алгоритмов, а именно SCAN, Spectral, LPA и Walktrap, реализована в библиотеке Lightcluster. Алгоритмы Infomap и Spinglass реализованы в библиотеке Igraph. Обе библиотеки написаны для языка программирования Python.

Все алгоритмы, за исключением алгоритма LPA, используют в своей работе различные входные параметры.

Алгоритм Infomap использует механизм случайного блуждания [3]. Проблема изоляции сообществ в графе понимается как задача кодирования пути, через который пройдет путник, с минимизацией длины присвоенного кода. У каждого сообщества есть свой собственный уникальный двоичный код. В сообществе у каждой вершины есть свой собственный уникальный внутренний код, у вершин из различных сообществ может быть тот же самый код. Кроме того, есть дополнительный код завершения, который не соответствует главным кодам в этом сообществе. Кодирование пути происходит следующим образом: когда вершина попадает в другое сообщество, его код и внутренний код вершины фиксируются. Когда вершина переходит к другому сообществу, пишется код завершения для этого сообщества и код для нового, то есть кодирование происходит на сообществах и вершинах.

В основе алгоритма LPA или LabelPropagation лежит следующий принцип [4]: узел принадлежит сообществу, которому принадлежит большинство соседей. В начале расчета каждый кластер состоит из одной вершины. Далее метки кластеров пересчитываются итеративно. На каждом шаге вершины случайным образом перетасовываются, и вершины обновляют свои идентификаторы согласно меткам соседних вершин. Процесс завершается, когда каждая вершина оказывается с той же самой меткой, что и наибольшее число ее соседей. Можно видеть, что при использовании параметров по умолчанию первоначальное распределение вершин происходит случайным образом, что может приводить к разным результатам при нескольких расчетах.

Для алгоритма SCAN требуется задать количество смежных с вершиной вершин (μ), а также радиус соседства (ε). На основе этих двух параметров вычисляется плотность кластера [5]. Данный алгоритм позволяет выделить пересекающиеся кластеры, а также шум.

Алгоритм Spectral основан на нахождении собственных значений матриц, связанных с графом, и собственных векторов лапласиана графа [6]. Далее применяются известные алгоритмы кластеризации, например, k-средних.

Алгоритм Walktrap также использует метод случайного блуждания на графах. Идея состоит в том, что при случайном блуждании по графу вероятность остаться в одном кластере значительно выше, чем выйти за пределы этого кластера, поскольку между кластерами проходит меньше ребер, чем их насчитывается в кластере [7].

Описание графовых наборов данных для бенчмаркинга

№ п/п	Набор данных	Число вершин	Число ребер	Средняя степень графа
1.1	Dolphins	62	159	5,13
1.2	Football	115	613	10,66
1.3	Sustainable Banking	167	187	1,12
1.4	Zachary Karate Club NEW	34	156	4,59
2.1	Performance 15	15	19	1,27
2.2	Benefits 14	14	19	1,36
2.3	Наркоугрозы безопасности мегаполиса	25	35	1,40
2.4	Политика президента Малайзии	24	34	1,42
2.5	Deployment 19	19	28	1,47
2.6	Политика президента Южной Кореи	31	47	1,52
2.7	Training 17	17	27	1,59
3.1	Crime	7	15	2,14
3.2	Социально-экономическое развитие Кронштадта	37	86	2,32
3.3	Uni_reresearch	9	23	2,56
3.4	Качество системы нематериальной мотивации	46	252	5,48
3.5	Качество корпоративного управления	75	411	5,48
3.6	Качество корпоративной культуры	72	531	7,38
3.7	celegansneural	297	2345	7,90

Алгоритм Spinglass используется понятие спинового стекла из физики, когда при намагничивании частицы получают отрицательный или положительный заряд или спин. Как известно, при нагревании ферромагнитные сплавы теряют свое свойство, и заряды частиц в сплаве уравновешивают общее состояние магнитного поля. При понижении температуры ферромагнетик вновь обретает свое свойство, при этом частицы группируются позарядно, положительно заряженные – с положительно заряженными, а отрицательно заряженные – с отрицательными. Данный метод работает с ориентированными графами и учитывает веса дуг [8].

Для оценки эффективности и сравнения алгоритмов между собой нами использовались следующие метрики качества.

Для оценки кластеризации графов с известным разбиением можно использовать F-меру Ван Ризенбергена (F1 score, F score), которая нашла широкое применение в теории информационного поиска [9]. Эта характеристика, которая позволяет дать оценку одновременно по точности (precision) и полноте (recall). В контексте задачи кластеризации точность – это число вершин, попавших в кластер, соответствующий истинному кластеру, деленное на общее число вершин в истинном кластере, в то время как полнота – это число вершин, попавших в кластер, соответствующий истинному кластеру, деленное на общее число вершин в кластере. F-мера может быть интерпретирована как взвешенное среднее этих двух метрик, которое изменяется в диапазоне от единицы до нуля, где 1 – это наилучший результат.

Сложнее оценить качество кластеризации в случае, когда истинное разбиение неизвестно.

Модулярность – относительно новая, но популярная метрика была предложена М. Гирваном и М. Ньюманом [10]. Значение метрики лежит на интервале между 0,5 и 1, рассчитывается по формуле

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

где A – матрица смежности, A_{ij} (i, j) – элемент матрицы, d_i – степень i-й вершины, Ci – кластер, в котором находится вершина, m – количество дуг в графе. $\delta(C_i, C_j) = 1$, если $Ci = C_j$, в противном случае 0.

Преимущество модулярности состоит в том, что достаточно легко рассчитать прирост модулярности в случае перемещения некоторого количества вершин между кластерами, не пересчитывая заново метрику.

Удалив все ребра между полученными кластерами, мы получим два независимых подграфа или так называемый разрез графа [2]:

$$cut\left(\hat{C}_i,\ldots,\hat{C}_{N_{\hat{C}}}\right) = \frac{1}{2}\sum_{k=1}^{N_{\hat{C}}}\sum_{i\in\hat{C}_k, j\notin\hat{C}_k}\omega_{ij}.$$

Нормализовав группы вершин на объем занимаемый ими в графе, мы получим нормализованный разрез графа (Normalized Cut):

$$NormalizedCut\left(\hat{C}_i,\ldots,\hat{C}_{N_{\hat{C}}}\right) = \sum_{i=1}^{N_{\hat{C}}}\frac{cut\left(\hat{C}_i, V\setminus\hat{C}_i\right)}{vol\left(\hat{C}_i\right)}.$$

Идея относительного разреза (Ratio Cut) графа состоит в том, чтобы заменить значение $vol\left(\hat{C}_i\right)$ каждого блока его размером – количеством вершин графа $\left|\hat{C}_i\right|$. В этом случае мы можем описать относительный разрез следующей формулой:

$$RatioCut\left(\hat{C}_i,\ldots,\hat{C}_{N_{\hat{C}}}\right) = \sum_{i=1}^{N_{\hat{C}}}\frac{cut\left(\hat{C}_i, V\setminus\hat{C}_i\right)}{\left|\hat{C}_i\right|}.$$

Заметим, что указанные метрики в базовой постановке неприменимы для оценки качества пересекающихся кластеров, требуется их модификация.

Результаты исследования и их обсуждение

При анализе результатов на основе модулярности (рис. 1) можно выделить графы, хорошо поддающиеся кластеризации, и графы с плохой или неудовлетворительной кластеризацией. Графы *Uni reresearch*, *Crime*, *«Качество системы нематериальной мотивации»*, *«Качество корпоративной культуры»*, *«Качество корпоративного управления»* и *Performance* продемонстрировали плохую тенденцию разбиения на подграфы, что может говорить о некоторой структурной особенности этих графов, затрудняющей их кластеризацию. Все вышеперечисленные графы находятся во второй подгруппе, то есть в той группе графов, средняя степень которых выше двух. В то же время графы *dolphins*, *«Политика президента Южной Кореи»*, *Deployment*, *Sustainable Banking* и *football* поддаются более качественному разбиению.

Из всех алгоритмов, представленных в статье, наилучшие результаты показали алгоритмы *Spinglass* и *Walktrap*. При этом метод *Spinglass* демонстрирует лучший показатель по метрике *Modularity* относительно метода *Walktrap*.

Метод *Spectral* в целом показал хорошие результаты, за исключением графа *Sustainable Banking*. Кроме того, следует отметить нестабильность этого метода относительно других алгоритмов. Большую нестабильность можно наблюдать только у алгоритма *SCAN*, единственного из алгоритмов, который работает с пересекающимися кластерами.

В силу особенностей метрик Ratio Cut и Normalized Cut верхняя граница метрик не определена, что затрудняет визуальное восприятие графиков при оценке качества разбиения на кластеры.

На графах, для которых было известно истинное разбиение (рис. 2), алгоритм Spinglass не потерял своих позиций, достаточно точно распределив вершины по кластерам и даже добившись идеального разбиения в графе *Zachary Karate Club*, что можно объяснить небольшим количеством идеальных кластеров на выходе.

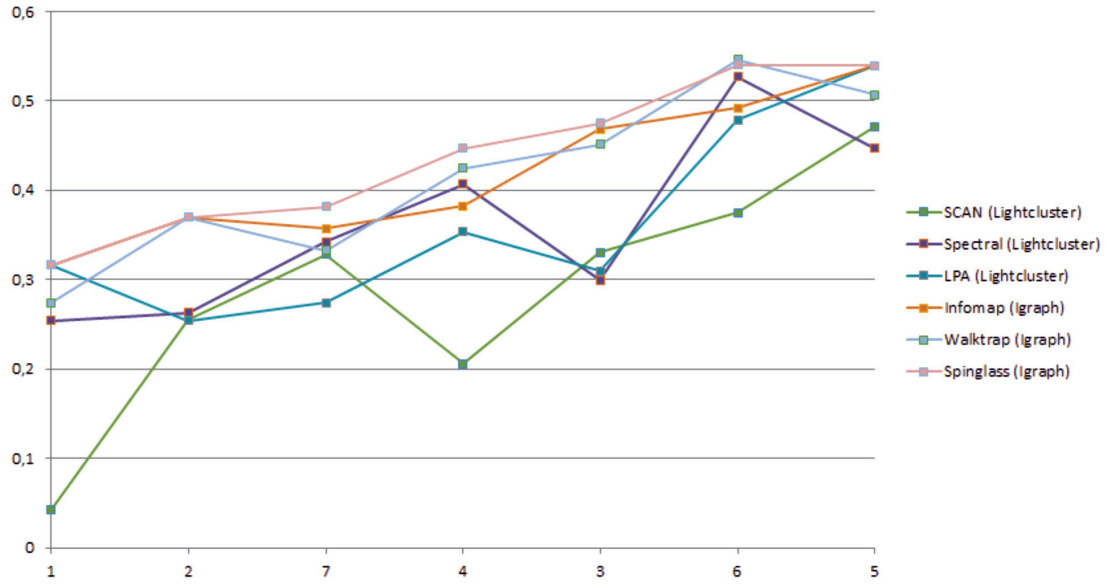

Рис. 1. Значения метрики модулярности для графов со степенью больше 2

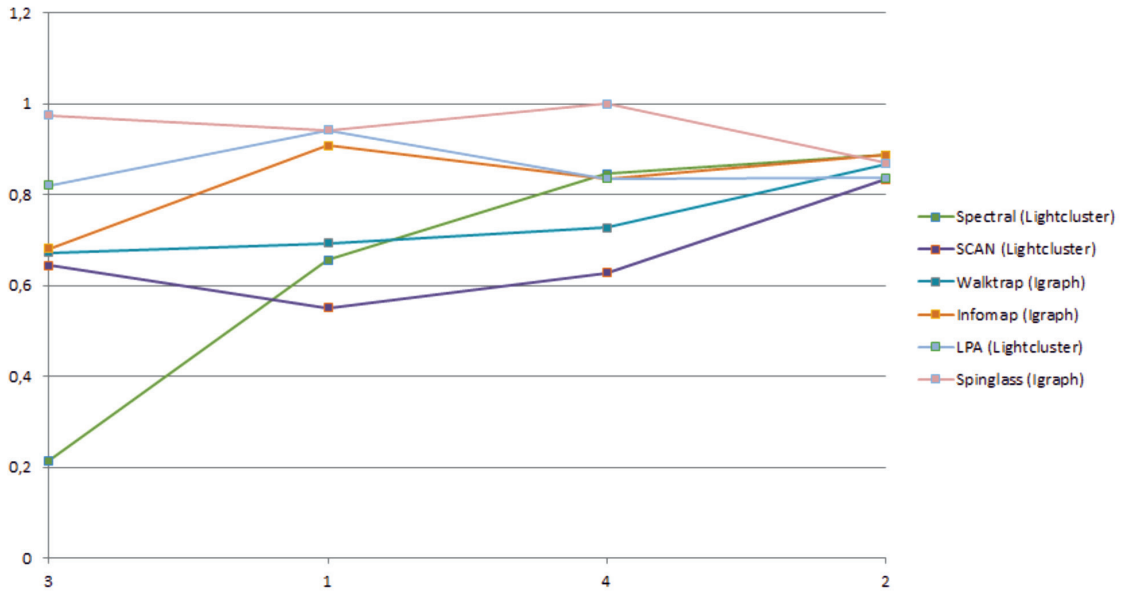

Рис. 2. Значения метрики F-мера для графов с известным разбиением на кластеры

В то же время метод Walktrap не попал в первую тройку. Это можно объяснить двумя причинами. Во-первых, мы не задавали в данном случае алгоритму явных значений входных параметров, позволив ему вести расчет с параметрами, которые он вычислял самостоятельно. Вторая причина может заключаться в том, что данный алгоритм вычисляет результат, подбирая оптимальное значение метрики модулярности, в то время как в графах, моделирующих реальную практическую ситуацию, значение метрики модулярности может отличаться от идеального.

Метод Spectral для графов с известным разбиением показал худший результат, особенно на модели *Sustainable Banking*.

Особенно нужно выделить алгоритм SCAN, еще одного аутсайдера. Эталонные графы были выбраны нами с расчетом

того, что не имеют вершин, принадлежащих одновременно нескольким кластерам. Метод SCAN, ориентированный на выявление пересекающихся кластеров, выделил много именно таких пересекающихся кластеров.

Заключение

В этой статье мы рассмотрели основные типы алгоритмов кластеризации графов. Оценка эффективности и сравнение алгоритмов проводились на основе таких метрик качества, как F-мера, модулярность, нормализованный и относительный разрезы графа. В ходе эксперимента шестью алгоритмами была произведена обработка 18 эталонных графов, разделенных на подгруппы на основе средней степени графа, отражающей сложность отношений между вершинами.

На основе результатов эксперимента может быть сделано заключение о том, что среди представленных алгоритмов кластеризации нет универсального алгоритма, который бы значительно опередил остальные на всех наборах данных. Лидерами бенчмаркинга являются алгоритмы Spinglass и Walktrap. Алгоритм SCAN несколько выбивается из общего ряда алгоритмов и стоит особняком в данном эксперименте. Можно также выделить алгоритм Spectral, получивший высокие оценки на большинстве графов.

При анализе значений метрик качества мы отметили, что при одинаковом разбиении на кластеры получались разные оценки качества кластеризации.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-01-00243.

Список литературы

1. Миркин Б.Г. Методы кластер-анализа для поддержки принятия решений: обзор / Б.Г. Миркин. – М.: Изд. дом Национального исследовательского университета «Высшая школа экономики», 2011. – 88 с.

2. Силин И., Панов М. Обзор и экспериментальное сравнение алгоритмов кластеризации графов. // Информационные технологии и системы 2015: Сборник трудов 39-й междисциплинарной школы-конференции ИППИ РАН (Сочи, 07–11 сентября 2015 г.) – М.: Изд-во Института проблем передачи информации им. А.А. Харкевича РАН, 2015. – С. 1042–1059.

3. Rosvall M., Axelsson D., Bergstrom C.T. The map equation // The European Physical Journal-Special Topics. – 2009. – No. 178. – P. 13–23.

4. Raghavan U.N., Albert R., Kumara S. Near linear time algorithm to detect community structures in large-scale networks. URL: https://arxiv.org/abs/0709.2938 (дата обращения: 05.11.2017).

5. Ester M., Kriegel H.-P., Sander J., Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD'96 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. – 1996. – P. 226–231.

6. Luxburg U.V. A Tutorial on Spectral Clustering // Statistics and Computing. – 2007. – No. 17(4). – P. 395–416.

7. Latapy P., Pons P. Computing Communities in Large Networks // Journal of Graph Algorithms and Applications. – 2006. – Vol. 10, No. 2. – P. 191–218.

8. Reichardt J., Bornholdt S. Statistical Mechanics of Community Detection. URL: https://arxiv.org/pdf/cond-mat/0603718v1.pdf html (дата обращения: 05.11.2017).

9. Powers D.M.W. Evaluation: from Precision, Recall and F-measure to ROC // Journal of Machine Learning Technologies. – Vol. 2. – P. 37–63.

10. Girvan M., Newman M.E.J. Community structure in social and biological networks // Proc Natl Acad Sci USA. – 2002. – Vol. 99, No. 12. – P. 7821–7826.