Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Фаляхов Искандер Рамилевич

Предобработка графов в задачах кластеризации с высокой плотностью связей

010990 — Интеллектуальный анализ данных

Выпускная квалификационная работа магистра

Научный руководитель:
д.т.н. Матвеев Иван Алексеевич

Москва
2020

Содержание

1 Введение

Анализ и поиск паттернов в графовой структуре актуален в различных прикладных областях. Это связано с тем, что большое количество видов взаимосвязей объектов в реальном мире можно формализовать с помощью графов: социальные сети, соединения белков, схемы дорог и т.д. Актуальны задачи обучения без учителя (кластеризация одна из них), ведь процесс разметки данных часто является очень дорогим и времязатратным, из-за чего размечается очень малая часть данных. Следует заметить, что в задачах кластеризации графов часто важен не только сам алгоритм кластеризации, но методы предобработки графа до его применения, так как в случаях высокой плотности связи узлов и высокой сложности структуры данных применение алгоритма без предварительной работы с графом редко приводит к удовлетворительным результатам. В этой работе будет проведён анализ методов предобработки графов и показано, в каких случаях их применение приведет к улучшению качества кластеризации.

Мотивацией проведения этой работы послужила прикладная задача определения рабочих групп сотрудников компании Qiwi. Для этого использовался набор данных с их деловыми встречами. Эти данные преобразовывались в граф встреч, где узлы — сотрудники компании, рёбра — их встречи друг с другом.

Во время решения этой задачи, была обнаружена проблема — многие методы кластеризации графов выдавали плохие по качеству и по стабильности результаты на данных данной природы. Было решено провести анализ методов предобработки графов, чтобы увидеть, какие из них могут улучшить результаты работы алгоритмов кластеризации.

2 Обзорно-постановочный раздел работы

2.1 Основные понятия и определения

В работе рассматривается взвешенный граф $G = (V, E, W)$, где V — непустое множество узлов, E — множество неупорядоченных пар узлов (рёбра), W — множество весов, соответствующие весам рёбер.

Степень узла (degree) — количество узлов, с которыми у этого узла есть общее ребро.

Разбиение узлов на кластеры $C = \{c_1, c_2, \ldots\}$, где c_i — кластер-подмножество, $c_i \subseteq V$, $\cup c_i = V$ и $c_i \cap c_j = \varnothing$ при $i \neq j$.

Модулярность графа — метрика оптимизации, вычисляемая по формуле:

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w), \tag{1}$$

где A_{vw} — вес рёбер смежных узлами v и w одновременно, $m = |E|$, k_v — сумма весов рёбер смежных узлу v, c_v — кластер узла v. Метрика принимает значения на отрезке $[-\frac{1}{2}, 1]$. Чем больше значение, тем сильнее связаны узлы внутри одного кластера, чем узлы из разных кластеров. Это широко используемая метрика оптимизации для алгоритмов кластеризации графов [1, 2].

Мера Жаккара — мера сходства двух множеств, вычисляемая по формуле:

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}, \tag{2}$$

где S_1, S_2 — множества объектов.

Степень связанности узлов:

$$S(v, w) = \frac{|N(v) \cap N(w)|}{|N(v) \cup N(w)|}, \tag{3}$$

где v, w — узлы графа, $N(v)$ — множество соседей узлов v. По сути, это мера Жаккара по множествам соседей узлов, чью степень связности нужно найти.

Коэффициент ассортативности степеней узлов (degree assortativity) — описывающее свойство графа. Описывающее свойство графа — некая характеристика, показывающее свойство структуры графа (например, максимальная степень узлов в графе, количество узов в графе). Вычисляется как коэффициент корреляции Пирсона. Показывает линейную зависимость степени узлов от степени узлов их соседей по рёбрам, чем оно выше, тем чаще узлы с большим количеством связей имеют общее ребро с узлами с большим количеством связей, а узлы с малым количеством связей имеют общее ребро с узлами с малым количеством связей. Пусть $x^m = \{x_1, x_2, ..., x_m\}, y^m = \{y_1, y_2, ..., y_m\}$, — это степени узлов с левого и правого края рёбер соотвественно, m — количество рёбер в графе. То есть x_i и y_i — степени

вершин в графе, у которых есть общая вершина. Тогда, коэффицент ассортативности вычисляется по формуле:

$$A(x,y) = \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x})^2 \sum_{i=1}^{m}(y_i - \overline{y})^2}}, \tag{4}$$

где $\overline{x}$ и $\overline{y}$ — выборочные средние.

2.2 Обзор современного состояния проблемы

При рассмотрении задач кластеризации графов основными аспектами являются:

- Метрика оптимизации;
- Алгоритм оптимизации;
- Методы предобработки.

В моей работе внимание сфокусировано на методах предобработки.

Родоначальником теории графов считается Леонард Эйлер в 1736 году сформулировавший задачу о семи кёнигсбергских мостах. Термин «граф» был введён спустя 150 лет Джеймсом Джозефом Сильвестром.

Известными задачами теории графов можно назвать: задачу коммивояжёра, проблему 4 красок, нахождение минимального остовного дерева, задачу о мостах, упомянутую выше.

Машинное обучение также используется в решении задач на графах, в том числе в задаче кластеризации графов — разбиения его узлов на кластеры (группы). Чаще всего эту задачу решают алгоритмами, оптимизирующими модулярность разбиения. Сейчас набирают популярность решения, использующие нейронные сети для представления структурной информации о графе в табличном виде, т.е. каждый узел представляется в виде вектора с помощью погружения графа в линейное пространство признаков (embedding).

Можно заметить, что в статьях, посвященных задаче кластеризации графов, довольно часто вопрос предобработки графа опускается. Особенно часто это бывает в работах, описывающих применение глубоких сетей для представления графа в табличном виде [3, 4, 5, 6, 7].

В довольно масштабной статье [8], в которой рассказывается вся основная теория задачи кластеризации графов, проведён анализ большого числа алгоритмов с использованием метода препроцессинга L-SPAR. Можно указать недостаток, состоящий в том, что там используются алгоритмы, которые для решения задач кластеризации графов с большими наборами данных считаются уже далеко не самыми лучшими. Здесь произведена работа с алгоритмами, которые не затронуты в этой статье.

В статье [9] говорится, что основной метрикой для оптимизации используется модулярность, но они применили другую метрику The Weakly Connected Component (WCC), которая которая максимизирует среднее количество треугольников в кластерах. Но по графикам их метрик видно, что Лувенский алгоритм, оптимизирующий метрику модулярности, имеет такое же качество.

В следующей статье [10] представлена одна из версий алгоритма L-SPAR, использующая параметр разреженности, который использовался и в моей работе. Также там используется мера Жаккара для оценки схожести соседей узлов, которая использовалась в моей работе для оценки стабильности кластеризации. Одним из недостатков является то, что в статье явно не указывается сам алгоритм кластеризации Также хотелось бы узнать, как меняется стабильность разбиений при разных методах предобработки. Более того, в моей работе указаны некоторые недостатки их версии алгоритма L-SPAR.

В [11] снова применяется мера Жаккара, но в использовании как алгоритма кластеризации. Не указываются методы предобработки.

В [12] первый раз вводится понятие ассортативности и поднимается вопрос важности этой характеристики графа. Ассортативность используется в этой работе при анализе результатов вычислительных экспериментов на разных наборах данных.

Одна из основных проблем статей связанных с данной темой то, что во многих случаях не проводится анализ методов предобработки.

2.3 Постановка задачи

В этой работе применены разные алгоритмы предобработки графа в различных комбинациях вместе Лувенским алгоритмом (его работа будет описана позднее) кластеризации. Задачей работы является проведение анализа этих методов на разных при-

мерах данных, сверить их качество, найти случаи, в которых какие-то методы и их комбинации работают лучше других и объяснить почему.

3 Алгоритмы кластеризации

Алгоритмом кластеризации графов выбран Лувенский алгоритм, разработанный в Католическом университете г. Лёвен.

Алгоритм описывается в [13]. В начале работы алгоритма каждый узел образует отдельный кластер из одного узла. Итерация алгоритма состоит из двух фаз.

На первой фазе для каждого узла происходит попытка найти кластер, перемещение в который даст максимальный положительный прирост метрики модулярности (1). Переместить вершину можно только по смежным рёбрам, то есть только в те кластера, которым принадлежат узлы-соседи. Просмотр узлов графа продолжается до тех пор, пока алгоритм производит перемещения узлов в новые кластера.

На второй фазе граф агрегируется: каждый кластер становится вершиннами нового графа. Рёбра между этими узлами-кластерами образуются следующим образом: если между узлами, входящими в эти кластера, были рёбра, то между узлами-кластерами ребро то же есть. Алгоритм останавливается, когда граф перестает изменяться.

Асимптотика сложности — $O(n \log(n))$, где n — количество узлов в графе.

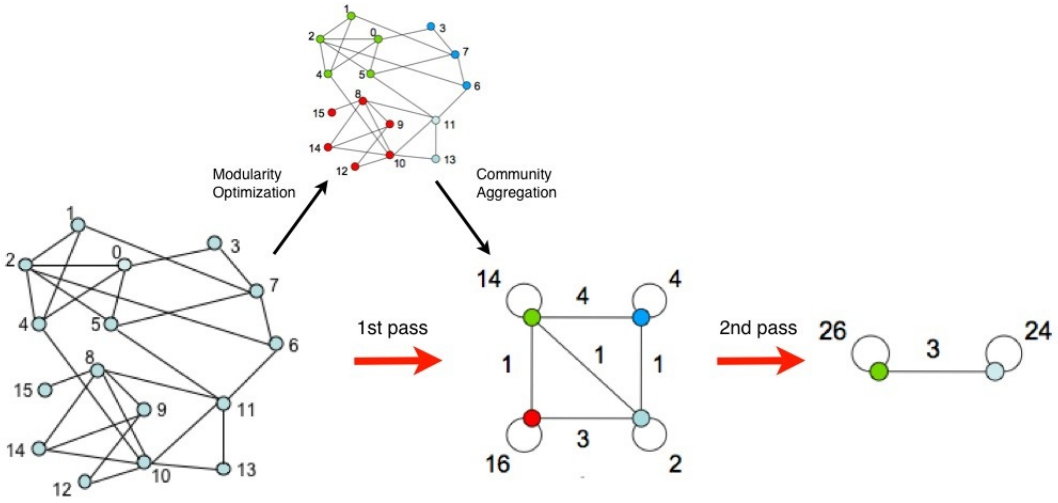

Рис. 1: Схема работы Лувенского алгоритма

Data: граф — $G = (V, E)$

Result: разбиение $C = \{c_1, c_2, ..., c_r\}$

// Каждый узел изначально составляет отдельный кластер

for *i = 1, ... , |V|* **do**

    $c_i = \{v_i\}$;

    $C = C \cup c_i$;

end

$modularity_{max} = Q(G, C)$;

$moves = 1$;

while $moves > 0$ **do**

    $moves = 0$;

    for *i = 1, ..., |V|* **do**

        for v' ***in*** $v_i.neighbors$ **do**

            c' — кластер, которому принадлежит v';

            C' — копия разбиения C, но где кластера c_i и c' объединены;

            if $Q(G, C') > modularity_{max}$ **then**

                $C = C'$;

                $moves = moves + 1$;

                $modularity_{max} = Q(G, C')$;

            end

        end

    end

    if *moves > 0* **then**

        $G' = (V', E')$ — агрегированный граф, где $V' = C$. Рёбра образуются следующим образом — если между любым узлом из кластера c_i имеется ребро в кластер c_j, то будет ребро из v'_i в v'_j;

        $G = G'$

    end

end

return C

Algorithm 1: Лувенский алгоритм

4 Методы предобработки

4.1 Удаление мостовых вершин с помощью метрики центральности

Самым первым методом предобработки графа была использована метрика центральности (Betweenness Centrality) [14], вычисляемая по формуле:

$$B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{5}$$

где s, v, t — узлы графа; σ_{st} — количество кратчайших путей из узла s в узел t; $\sigma_{st}(v)$ — количество кратчайших путей из узла s в узел t, проходящих через узел v;

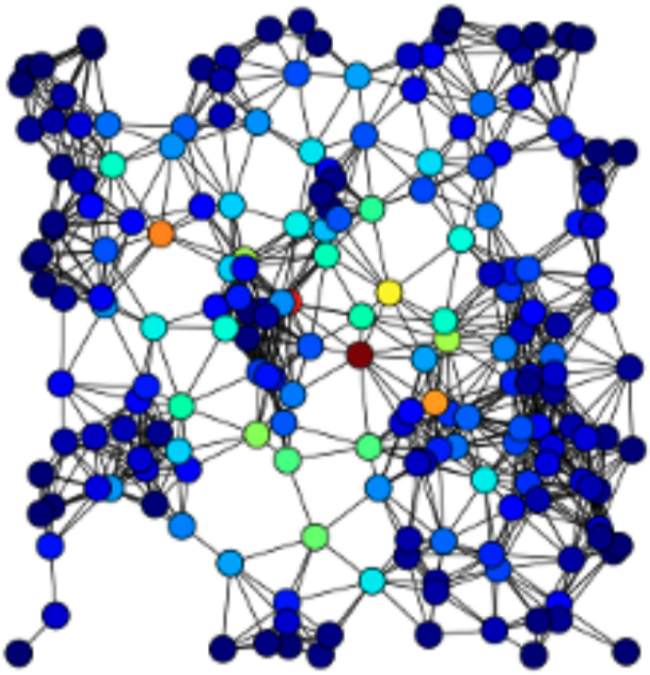

Рис. 2: Пример графа, с вычисленными значениями Betweenness Centrality. Чем теплее цвет, тем больше значение

Эта метрика позволяет находить узлы, являющиеся связующим звеном между большим количествои узлов. Такие узлы будем в дальнейшем называть мостовыми. Я предполагаю, что перед кластеризацией лучше очищать граф от мостовых вершин, потому что они будут связывать кластера друг с другом, ухудшая качество кластеризации, ведь алгоритм может начать считать эти кластера за один. Метод

предобработки заключается в таких шагах: вычисляем метрику для всех узлов; находим узлы, у которых значение метрики выше, чем среднее по всему графу; удаляем эти вершины. Кластера для непосредственно мостовых вершин определялись уже в постобработке методом голосования: с каким кластером у узла больше общих рёбер, к такому кластеру и относить эту вершину.

Асимптотика сложности — $O(mn)$, где m — количество рёбер в графе, n — количество узлов в графе.

4.2 Метод L-SPAR с параметром разреженности

Local Sparsification (L-SPAR) — метод, находящий рёбра в графе, которые соединяют плохосвязные вершины и убирает их. Основная идея состоит в том, чтобы сохранить основную структуру графа, при этом убрав шумные связи, которые могут помешать алгоритму кластеризации построить более качественное разбиение.

Метод проходит по всем вершинам графа, вычисляет степень связности (3) каждого узла с его соседями, сортирует соседей по степени связности и оставляет рёбра только с некоторым количеством хорошо связанных соседей.

В этой версии метода требуется подбор параметра разреженности, задающий количество хорошо связанных соседей, с которыми нужно оставить ребра. Параметр разреженности — $\omega \in [0, 1]$, количество узлов, рёбра с которыми нужно оставить — $N(v)^{\omega}$, где v — узел, $N(v)$ — количество соседей узла v.

Асимптотика сложности — $O(n)$, где n — количество узлов в графе.

Data: граф — $G = (V, E)$, параметр разреженности w

Result: предобработанный граф — G_{sparse}

G_{sparse} — пустой граф;

for v *in* E **do**

  d_v — степень узла v;

  E_v — рёбра, смежные узлу v;

  for $e = (v, k)$ *in* E_v **do**

    $e.sim = Sim(v, k)$ — схожесть узлов, посчитанная по формуле (3);

  end

  Отсортировать все рёбра в E_v по $e.sim$ в порядке невозрастания;

  Добавить первые d_v^w в граф G_{sparse};

end

return G_{sparse}

Algorithm 2: L-SPAR с параметром разреженности

4.3 Усредненный метод L-SPAR

Данная версия метода отличается от версии с параметром разреженности способом определения количества хорошо связанных соседей, с которыми нужно оставить ребро. В этой версии метода рёбра оставляют со всеми соседями, чья степень связности больше средней степени связанности узла с его соседями.

Данная версия имеет преимущества перед предыдущей в виду отсутствия параметра разреженности, который требует подбора под конкретный граф.

Асимптотика сложности — $O(n)$, где n — количество узлов в графе.

5 Метрики качества

В этом разделе описаны метрики качества кластеризации, которые были использованы в анализе методов.

Все метрики качества кроме последней требуют того, чтобы были известны реальные разбиения узлов графа на классы. В дальнейшем в формулах через $Y = \{Y_1, Y_2, ...\}$ будет обозначаться список реальных классов, которым принадлежат узлы.

5.1 Чистота

Чистота (purity) — разновидность точности (accuracy) для задачи кластеризации. Вычисляется по формуле:

$$PU(C, Y) = \frac{1}{N} \sum_{i=1}^{|C|} \max_j |c_i \cap Y_j| , \tag{6}$$

где c_i — i-ый кластер, Y_j — j-ый класс, N - количество узлов.

Суть метрики заключается в том, что каждому кластеру даётся метка класса, членов которого в нем находится больше всего. После обозначений этих меток, вычисляется доля узлов с правильными метками класса.

5.2 Нормированная взаимная информация

Нормированная взаимная информация (Normalized Mutual Information, NMI) — метрика, основанная на понятии энтропии. Вычисляется по формуле:

$$NMI(C, Y) = \frac{I(C, Y)}{[H(C) + H(Y)]/2}, \tag{7}$$

$$I(C, Y) = \sum_i^{|C|} \sum_j \frac{c_i \cap Y_j}{N} \frac{c_i \cap Y_j}{|c_i||Y_j|}, \tag{8}$$

$$H(X) = -\sum_L \frac{|X_L|}{N} \log \left(\frac{|X_L|}{N} \right), \tag{9}$$

$H(X)$ — энтропия множества.

5.3 Исправленный индекс Рэнда

Исправленный индекс Рэнда (Adjusted Rand Index, ARI) — усовершенственная версия Индекса Рэнда (Rand Index). Индекс Рэнда вычисляется по формуле:

$$RI = \frac{a+d}{a+b+c+d}, \tag{10}$$

где a — количество пар, находящихся в одном классе и в одном кластере; d — количество пар, находящихся в разных классах и в разных кластерах; c — количество пар, находящихся в одном классе, но в разных кластерах; b — количество пар, находящихся в разных классах, но в одном кластере. Метрика может принимать значение от 0 до 1.

Проблема этой метрики состоит в том, что при сильном увеличении количества кластеров, эта метрика может увеличивать свое значение. Исправленный индекс Рэнда нивелирует этот недостаток. Представим таблицу:

Y / C	c_1	c_2	...	c_s	Суммы по горизонтали
Y_1	n_{11}	n_{12}	...	n_{1s}	a_1
Y_2	n_{21}	n_{22}	...	n_{2s}	a_2
...					
Y_r	n_{r1}	n_{r2}	...	n_{rs}	a_r
Суммы по вертикали	b_1	b_2	...	b_s	n

Где Y_i — i-ый класс (разметка); c_j — j-ый кластер; n_{ij} — количество узлов из i-ого класса, попавшие в j-ый кластер; a_i — количество узлов из i-ого класса; a_j — количество узлов из j-ого кластера; n — количество узлов всего в графе.

Для описанной выше таблицы смежности, ARI вычисляется по формуле:

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)}, \tag{11}$$

где $RI = \sum\limits_{ij} C^2_{n_{ij}}$;
$E(RI) = [\sum\limits_{i} C^2_{a_i} \sum\limits_{j} C^2_{b_j}]/C^2_n$ — ожидаемое зачение RI;
$max(RI) = \frac{1}{2}[\sum\limits_{i} C^2_{a_i} + \sum\limits_{j} C^2_{b_j}]$ — максимальное значение RI;
$C_(n)^k$ — биномиальный коэффициент из n по k,

5.4 Стабильность разбиения

Данная метрика является вторичной, потому что напрямую не показывает качество самого разбиения. Однако, она является важной, в случае, когда алгоритм кластеризации является частично случайным. Лувенский алгоритм является частично случайным, так как в разных случаях он может начать объединять узлы в сообщества в разном порядке, из-за чего результат разбиений может немного отличаться от запуска к запуску.

Вычисляется стабильность следующим алгоритмом:

1. Строится 2 разбиения;
2. Для этих двух разбиений вычисляется мера Жаккара (2);
3. Предыдущие два шага повторяются 10 раз;
4. Получившиеся меры Жаккара усредняются.

На 3-ем шаге количество повторений равно 10. Предполагается, что оптимальное значение числа повторений зависит от количества узлов и рёбер, но вычислить это теоретическое значение не получилось. Однако выявлено, что 10 раз достаточно для наборов данных, используемых в вычислительных экспериментах.

Низкое значение этой метрики будет показывать, что найденная структура данных является случайной, так как при повторных запусках алгоритм не может выявить ее снова.

6 Данные

6.1 Qiwi

Основным набором данных для проведения экспериментов является граф о деловых встречах сотрудников компании Qiwi, так как эта работа изначально возникла из задачи определения сообществ сотрудников внутри компании по рабочим группам. Вершинами графа являются сотрудники компании, а рёбрами — встречи сотрудников между собой. Граф является частично размеченным, примерно для 20% процентов узлов известны их реальные кластера.

Изначально данные были очищены от встреч с количеством сотрудников больше 20 человек, потому что такие встречи обычно не говорят о тесной связи сотрудников друг с другом (например, большие общие встречи департаментов) и могут негативно влиять на результаты моделирования.

Количество узлов: 1440, количество рёбер: 16972, максимальная степень вершина: 147, средняя степень вершины: 24, максимальное количество треугольников: 1689, среднее количество треугольников: 153, ассортативность: 0.19

Далее вкратце описаны другие данные, на которых были проведены дополнительные эксперименты.

6.2 Cora

Данные [15] посвящёны перекрёстным ссылкам в научных статьях. Рёбрами являются ссылки между статьями. Данные размечены, каждая статья отнесена к одному из 7 классов.

Количество узлов: 2708, количество рёбер: 5278, максимальная степень узлов: 168, средняя стпень узлов: 4, максимальное количество треугольников: 160, среднее количество треугольников: 2, ассортативность: -0.06

6.3 Citeseer

Набор данных [16] тоже посвящён научным работам. Узлы — научные статьи, рёбра — ссылки с между ними. Статьи распределены по 6 классам.

Количество узлов: 3264, количество рёбер: 4536, максимальная степень узлов:

99, средняя степень узлов: 3, максимальное количество треугольников: 85, среднее количество треугольников: 2, ассортативность: 0.05

6.4 COLLAB

Набор данных [17] представляет из себя эго-сети сотрудничества исследователей в трёх областях: физика высоких энергий, физика твёрдого тела, астрофизика. Самый большой набор данных, на котором здесь проводились вычислительные эксперименты.

Количество узлов: 372474, количество рёбер: 12288916, максимальная степень узлов: 491, средняя степень узлов: 66, максимальное количество треугольников: 39633, среднее количество треугольников: 5016, ассортативность: 0.92

Далее в работе описаны методы и алгоритмы, используемые в этой работе.

7 Вычислительные эксперименты

В этом разделе приведены вычислительные эксперименты на данных, описанных в соответствующем разделе, с разными комбинациями методов предобработки. Комбинации, с которыми были проведены эксперименты:

- Отсутствие методов предобработки ($LOUV$);
- Удаление мостовых вершин (BD);
- Удаление рёбер алгоритмом L-SPAR с оптимальным параметром разреженности (LSP);
- Удаление рёбер усреднённым алгоритмом L-SPAR ($AverLSP$);
- Удаление мостовых вершин, затем удаление рёбер алгоритмом L-SPAR с оптимальным параметром разреженности ($BD + LSP$);
- Удаление мостовых вершин, затем удаление рёбер усреднённым алгоритмом L-SPAR ($BD + AverLSP$);
- Удаление рёбер алгоритмом L-SPAR с оптимальным параметром разреженности, затем удаление мостовых вершин ($LSP + BD$);

- Удаление рёбер усреднённым алгоритмом L-SPAR с оптимальным параметром разреженности, затем удаление мостовых вершин ($AverLSP + BD$).

Результаты вычислительных экспериментов:

- В наборах данных Qiwi и Cora хорошие результаты показали алгоритмы с последовательным применением удаления мостовых вершин и алгоритмом L-SPAR, с параметром разреженности и усредненный. Следует заметить, что применение алгоритма L-SPAR с подобранным параметром разреженности показывает результаты хуже, либо не сильно превышающий по качеству усредненный алгоритм L-SPAR.

- В наборе данных Citeseer результаты менее однозначные: хороших результатов добился алгоритм с последовательным применением удаления мостовых вершин и алгоритмом L-SPAR с параметром разряженности и алгоритм без применения методов предобработки.

- В наборе данных COLLAB наилучших результатов добился алгоритм без применения каких-либо методов предобработки.

Таблица 1: Метрики качества на наборе данных Qiwi

Algo	PU	NMI	ARI	STAB
$LOUV$	0.64	0.38	0.25	0.6
BD	0.69	0.41	0.29	0.69
LSP	0.75	0.63	0.5	0.75
$AverLSP$	0.74	0.63	0.51	0.76
$BD + LSP$	**0.88**	**0.79**	**0.65**	**0.89**
$BD + AverLSP$	0.87	**0.79**	**0.65**	**0.89**
$LSP + BD$	0.81	0.76	0.59	**0.89**
$AverLSP + BD$	0.81	0.76	0.59	**0.89**

Таблица 2: Метрики качества на наборе данных Cora

Algo	PU	NMI	ARI	STAB
$LOUV$	0.69	0.4	0.19	0.65
BD	0.73	0.41	0.21	0.7
LSP	0.74	0.4	0.2	0.7
$AverLSP$	0.74	0.39	0.2	0.7
$BD + LSP$	**0.78**	0.46	**0.31**	**0.91**
$BD + AverLSP$	**0.78**	**0.47**	0.3	**0.91**
$LSP + BD$	0.75	0.42	0.24	**0.91**
$AverLSP + BD$	0.75	0.41	0.24	**0.9**

Таблица 3: Метрики качества на наборе данных Citeseer

Algo	PU	NMI	ARI	STAB
$LOUV$	**0.74**	0.37	**0.1**	**0.95**
BD	0.73	0.39	0.08	**0.95**
LSP	**0.74**	0.38	0.07	**0.95**
$AverLSP$	**0.74**	0.37	0.07	**0.95**
$BD + LSP$	**0.74**	**0.42**	**0.1**	**0.95**
$BD + AverLSP$	**0.74**	0.41	0.09	**0.95**
$LSP + BD$	**0.74**	0.41	0.08	**0.95**
$AverLSP + BD$	0.73	0.4	0.08	**0.95**

Таблица 4: Метрики качества на наборе данных COLLAB

Algo	PU	NMI	ARI	STAB
$LOUV$	**1**	**1**	**1**	**1**
BD	**1**	**1**	**1**	**1**
LSP	0.91	0.87	0.75	**1**
$AverLSP$	0.91	0.87	0.75	**1**
$BD + LSP$	0.85	0.74	0.64	**1**
$BD + AverLSP$	0.84	0.74	0.59	**1**
$LSP + BD$	0.83	0.68	0.5	**1**
$AverLSP + BD$	0.83	0.68	0.5	**1**

8 Заключение

Проведён анализ различных методов предобработки графов: метод L-SPAR с параметром разреженности, усредненный алгоритм L-SPAR. Так же был предложен и проанализирован метод предобработки путём удаления из графа вершин, имеющих большие значение метрики центральности чем в среднем по графу. Указанные методы предобработки были проанализированы в различных комбинациях друг с другом.

Из полученных результатов можно вывести некоторые закономерности:

- На всех наборах данных применение алгоритма L-SPAR с подобранным параметром разреженности показывает результаты хуже, либо не сильно превышающий по качеству усредненный алгоритм L-SPAR. Учитывая то, что при применении алгоритма на неизвестных данных, параметр разреженности придется подбирать заново, можно сказать, что использование усредненного алгоритма L-SPAR может быть хорошей альтернативой, в виду одинакового порядка метрик и отсутсвия надобности подбора гиперпараметров, вследствие чего его можно применять к новому набору данных без дополнительного анализа.

- На наборах данных с большим значением среднего количества треугольников (T_{mean}) и малым значением ассортативности (assort.) лучше всего себя показывают алгоритмы с последовательным применением удаления мостовых вершин и алгоритмом L-SPAR, с параметром разряженности и усреднённый.

- На наборах данных с таким же малым значением ассортативности, но при этом малым средним количеством треугольников, нет чёткой картины. В таких случаях отсутвие предобработки может показать себя лучше.

- На наборах данных с значением ассортативности близким к единице методы предобработки могут быть излишни: алгоритм без применения методов предобработки показал себя максимально эффективным.

Список литературы

[1] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, D. Wagner. On Modularity Clustering // IEEE Transactions on Knowledge and Data Engineering. 2008. 172-188

[2] Martynov N., Khandarova O., and Khandarov F. Graph Clustering Based on Modularity Variation Estimations // The Bulletin of Irkutsk State University. Series Mathematics. 2018. 25. 63-78.

[3] Xiaotong Zhang, Han Liu, Qimai Li, Xiao-Ming Wu. Attributed Graph Clustering via Adaptive Graph Convolution // Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. 4327-4333. URL: https://www.ijcai.org/Proceedings/2019/601

[4] Zou Dongmian, Lerman Gilad. Encoding robust representation for graph generation // International Joint Conference on Neural Networks (IJCNN). 2019. 1-9. URL: https://ieeexplore.ieee.org/document/8851705

[5] Shaosheng Cao, Wei Lu, Qiongkai Xu. Deep Neural Networks for Learning Graph Representations // AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. 2016. 1145–1152. URL: https://www.semanticscholar.org/paper/Deep-Neural-Networks-for-Learning-Graph-Cao-Lu/1a37f07606d60df365d74752857e8ce909f700b3

[6] Carl Yang, Mengxiong Liu, Zongyi Wang, Liyuan Liu, Jiawei Han. Graph Clustering with Dynamic Embedding // arXiv:1712.08249. 2017. URL: https://arxiv.org/abs/1712.08249

[7] Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, Weixiong Zhang. Modularity based community detection with deep learning // IJCAI'16: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. 2016. 2252–2258. URL: https://dl.acm.org/doi/10.5555/3060832.3060936

[8] Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, Weixiong Zhang. Modularity based community detection with deep learning // IJCAI'16: Proceedings

of the Twenty-Fifth International Joint Conference on Artificial Intelligence. 2016. 2252–2258. URL: https://dl.acm.org/doi/10.5555/3060832.3060936

[9] Ovelgönne M. Scalable Algorithms for Community Detection in Very Large Graphs. // ArXiv. 2011. URL: https://pdfs.semanticscholar.org/cf4b/34e9e03d467a0fe75648cce25b49affd3dcd.pdf?_ga=2.65053038.1741635911.1591878252-682179575.1591878252

[10] Arnau Prat-Pérez, David Dominguez-Sal, and Josep-Lluis Larriba-Pey. High quality, scalable and parallel community detection for large real graphs // 23rd international conference on World wide web (WWW '14). 2014. 225–236. URL: https://dl.acm.org/doi/10.1145/2566486.2568010

[11] Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering // 2011 ACM SIGMOD International Conference on Management of data (SIGMOD '11). 2011. 721–732. URL: https://dl.acm.org/doi/10.1145/1989323.1989399

[12] Israa Hadi, Firas Sabar Miften. A Graph Clustering Algorithm Based on Adaptive Neighbors Connectivity // e-ISSN: 2289-8131 Vol. 9 No. 2-11. 2018. 18-23. URL: https://www.researchgate.net/publication/326235250_A_Graph_Clustering_Algorithm_Based_on_Adaptive_Neighbors_Connectivity

[13] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre. Fast unfolding of communities in large networks // Journal of Statistical Mechanics: Theory and Experiment. 2008. URL: https://iopscience.iop.org/article/10.1088/1742-5468/2008/10/P10008

[14] Ulrik Brandes. A faster algorithm for betweenness centrality // The Journal of Mathematical Sociology. 2001. 25:2. 163-177.

[15] M. E. J. Newman. Assortative mixing in networks // Physical Review Letters 89. 2002. 20870. URL: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.89.208701

[16] Ryan A. Rossi, Nesreen K. Ahmed. Cora Dataset, The Network Data Repository with Interactive Graph Analytics and Visualization // URL: http://networkrepository.com/cora.php

[17] Ryan A. Rossi, Nesreen K. Ahmed. Citeseer Dataset, The Network Data Repository with Interactive Graph Analytics and Visualization // URL: http://networkrepository.com/citeseer.php

[18] Ryan A. Rossi, Nesreen K. Ahmed. COLLAB Dataset, The Network Data Repository with Interactive Graph Analytics and Visualization // URL: http://networkrepository.com/COLLAB.php