Адаптивные рандомизированные алгоритмы выделения сообществ в графах

Т. П. Проданов
Санкт-Петербургский Государственный Университет
timofey.prodanov@gmail.com

Последние семнадцать лет развивается изучение сложных сетей и большое применение находит выделение тесно связанных групп узлов, или сообществ, в сложных сетях.
Эффективными оказались рандомизированные алгоритмы выделения сообществ, однако не существует набора параметров, при котором эти алгоритмы давали бы хороший результат на всех сложных сетях. Для решения этой проблемы в работе описывается применение алгоритма одновременно возмущаемой стохастической аппроксимации к рандомизированным алгоритмам для создания приспосабливающихся к входным данным адаптивных модификаций. Целью является создание алгоритмов, дающих хорошие значения на большем количестве сложных сетей.

Ключевые слова: выделение сообществ, сложные сети, стохастическая аппроксимация.

1. Введение

Исторически изучение сетей происходило в рамках теории графов, которая начала свое существование с решения Леонардом Эйлером задачи о кенигсбергских мостах [**?**]. В 1920-х взял свое начало анализ социальных сетей [**?**]. От изучения маленьких сетей внимание переходило к сетям из тысяч или миллионов узлов, развивались методы статистического анализа сетей. К примеру, теория массового обслуживания [**?**], рассматривающая в том числе сети запросов к телефонным станциям, использовала для описания потоков запросов распределение Пуассона.

Лишь последние двадцать лет развивается изучение *сложных сетей*, то есть сетей с неправильной, сложной структурой, в некоторых случаях рассматривают динамически меняющиеся сложные сети. Сложные сети с успехом были применены в таких разных областях, как эпидемиология [**?**], биоинформатика [**?**], поиск преступников [**?**], социология [**?**], изучение структуры и топологии интернета [**?**, **?**] и в многих других.

Типичной характеристикой узла сети является его *степень*, определяемая как количество ребер, выходящих из узла. В процессе изучения сложных сетей, построенных по реальным системам [?, ?, ?, ?, ?, ?], оказалось, что распределение степеней $P(s)$, определенное как доля узлов со степенью s среди всех узлов графа, сильно отличается от распределения Пуассона, которое ожидается для случайных графов. Также сети, построенные по реальным системам характеризуются короткими путями между любыми двумя узлами и большим количеством маленьких циклов [**?**]. Это показывает, что модели, предложенные теорией графов, не всегда будут хорошо работать для графов, построенных по указанным выше реальным системам.

Общее свойство для рассматриваемых в [?, ?, ?, ?, ?, ?] сетей является наличие *сообществ*. Под сообществами можно понимать такие группы узлов графа, что внутри каждой группы соединений между узлами много, а соединений между узлами разных групп мало. Например, тесно связанные группы узлов в социальных сетях представляют людей, принадлежащих социальным сообществам, плотно сплоченные группы узлов в интернете соответствуют страницам, посвященным распространенным темам [?]. Сообщества в сетях, описывающих взаимодействия между генами, связаны с функциональными модулями [**?**]. Способность находить и анализировать подобные группы узлов предоставляет большие возможности в изучении реальных систем, представленных с помощью сложных сетей. Поиск подобных групп узлов называют *выделением сообществ* в графах, или *кластеризацией*.

В [**?**] был предложен рандомизированный жадный алгоритм выделения сообществ на графах, а в [**?**] была представлена схема кластеризации основных групп графа. От выбора параметров этих алгоритмов выделения сообществ критически зависит качество их работы, и остается открытым вопрос об адаптивных версиях алгоритмов, которые были бы работоспособны на большем количестве задач. В выпускной квалификационной работе предлагаются новые версии алгоритмов, который этот вопрос в некоторой степени решают. Так же предложенные алгоритмы лучше решают проблему меняющегося во времени работы оптимального параметра.

Работа устроена следующим образом: в разделе **??** рассмотрена необходимая информация о графах и сложных сетях, существу-

ющие методы выделения сообществ и одновременно возмущаемая стохастическая аппроксимация. Затем в разделе **??** представлен адаптивный рандомизированный жадный алгоритм, его анализ и сравнение с рандомизированным жадным алгоритмом. В разделе **??** предложена адаптивная схема кластеризации основных групп графа, рассмотрены ее параметры и результаты сопоставлены с результатами неадаптивной схемы кластеризации основных групп графа.

2. Предварительные сведения

Определения и обозначения

Формально сложная сеть может быть представлена с помощью графа. В работе будут рассматриваться только невзвешенные неориентированные графы. Неориентированный невзвешенный граф $G = (\mathscr{N}, \mathscr{L})$ состоит из двух множеств — множества $\mathscr{N} \neq \emptyset$, элементы которого называются *узлами* или *вершинами* графа, и множества $\mathscr{L}$ неупорядоченных пар из множества $\mathscr{N}$, элементы которого называются *ребрами* или *связями*. Мощности множеств $\mathscr{N}$ и $\mathscr{L}$ равны N и L соответственно.

Узел обычно обозначают по его порядковому месту i в множестве $\mathscr{N}$, а ребро, соединяющее пару узлов i и j обозначают l_{ij}. Узлы, между которыми есть ребро называются *смежными*. Степенью узла назовают величину s_i, равную количеству ребер, выходящих узла i.

Пусть $G = (\mathscr{N}, \mathscr{L})$ — граф, *разбиением на сообщества* будем называть разбиение множества его вершин $P = \{C_1, \ldots, C_K\}$, то есть $\bigcup_{i=1}^{K} C_i = \mathscr{N}$ и $C_i \cap C_j = \emptyset \ \forall i \neq j \in 1..K$.

Множество сообществ $P = \{C_1, \ldots, C_{K_1}\}$ будем называеть разбиением на сообщества *на основе* разбиения $\widetilde{P} = \left\{\widetilde{C}_1, \ldots \widetilde{C}_{K_2}\right\}$, если $\forall i \in 1..K_2 \ \exists j \in 1..K_1 : \widetilde{C}_i \subset C_j$.

Нефорально, сообщество — это тесно сплоченное подмножество узлов графа $\mathscr{N}' \subset \mathscr{N}$. Два сообщества называются *смежными*, если существует ребро, направленное из вершины первого сообщества в вершину второго.

Модулярность

В 2004 году в [**?**] была введена целевая функция *модулярность*, оценивающая неслучайность разбиения графа на сообщества.

Допустим, имеется K сообществ, тогда *нормированная матрица смежности сообществ* $\mathbf{e}$ определяется как симметричная матрица размером $K \times K$, где элементы e_{ij} равны отношению количества ребер, которые идут из сообщества i в сообщество j, к полному количеству ребер в графе (ребра l_{mn} и l_{nm} считаются различными, где m, n — узлы). След этой матрицы $\mathrm{Tr}(\mathbf{e}) = \sum_{i \in 1..K} e_{ii}$ показывает отношение ребер в сети, которые соединяют узлы одного и того же сообщества, к полному количеству ребер в графе. Хорошее разбиение на сообщества должно иметь высокое значение следа.

Однако если поместить все вершины в одно сообщество — след примет максимальное возможное значение, притом, что такое разбиение не будет сообщать ничего полезного о графе. Поэтому определяется вектор $\mathbf{a}$ длины K с элементами $a_i = \sum_{j \in 1..K} e_{ij}$. Координата вектора a_i является *нормированной степенью сообщества* i и обозначает долю количества ребер, идущих к узлам, принадлежащим сообществу i, к полному количеству ребер в графе. Если в графе ребра проходят между вершинами независимо от сообществ — e_{ij} будет в среднем равно $a_i a_j$, поэтому модулярность можно определить следующим образом [?]:

$$Q(G, P) = \sum_{i \in 1..K} \left(e_{ii} - a_i^2 \right) = \mathrm{Tr}(\mathbf{e}) - \|\mathbf{e}^2\|, \tag{1}$$

где $\|\mathbf{x}\|$ является суммой элементов матрицы $\mathbf{x}$. Если сообщества распределены не лучше, чем в случайном разбиении — модулярность будет примерно равна 0. Максимальным возможным значением функции будет 1.

Теперь можно поставить задачу выделения сообществ следующим образом: *требуется найти* такое разбиение графа, на котором модулярность принимает максимальное значение. Можно заметить, что такая постановка задачи не использует какого-либо определения сообществ.

Такая задача все еще будет NP-сложной [**?**]. Однако преимущество модулярности состоит в том, что для того, чтобы посчитать,

какой выигрыш будет извлечен из объединения двух сообществ, необходимо произвести только одну операцию: $\Delta Q = 2(e_{ij} - a_i a_j)$, где i и j — потенциально объединяемые сообщества. Для того, чтобы объединить два сообщества необходимо сделать $O(\min\{n_i, n_j\})$ операций, где n_i и n_j обозначают количество смежных к i и j сообществ.

Рандомизированный жадный алгоритм (RG)

Эффективными оказались рандомизированные алгоритмы максимизации модулярности. В 2010 году Овельгенне и Гейер-Шульц в [?] был предложен рандомизированный жадный алгоритм (Randomized Greedy, *RG*), который на каждой итерации рассматривает k случайных сообществ и смежным к ним сообществ, а затем соединяет пару соседей, дающую наибольший выигрыш.

При отсутствии базового разбиения, на основе которого выделяются сообщества — перед первой итерацией граф разбивается на $K = N$ сообществ с одним узлом в каждом сообществом. Алгоритм останавливается в тот момент, когда не остается больше сообществ для объединения, однако применяются объединения сообществ до той итерации, когда объединение последний раз принесло глобальный выигрыш.

Далее в работе рандомизированный жадный алгоритм с параметром k будет обозначаться RG_k.

Схема кластеризации основных групп графа (CGGC)

В 2012 году Овельгенне и Гейер-Шульц выиграли конкурс 10th DIMACS Implementation Challenge в категории *кластеризация графа* со схемой кластеризации основных групп графа (Core Groups Graph Cluster, *CGGC*) [?]. Схема заключается в том, что сначала s *начальных алгоритмов* разбивают граф на сообщества. В тех вершинах, относительно которых начальные алгоритмы разошлись во мнении, выделяет по сообществам *финальный алгоритм*.

Формально это записывается следующим образом:

1. s начальных алгоритмов создают разбиения графа G на сообщества. S — множество *начальных разбиений*, то есть разбиений, полученных начальными алгоритмами
2. Создается *промежуточное разбиение* $\widetilde{P}$, равное максимальному перекрытию начальных разбиений из множества S
3. Финальным алгоритмом создается разбиение P графа G на основе промежуточного разбиения $\widetilde{P}$

Необходимо определить понятие *максимальное перекрытие*. Пусть существует множество разбиений $S = \{P_1, \ldots, P_s\}$, отображение $c_P(v)$ указывает, в каком сообществе находится узел v в разбиении P. Тогда у максимального перекрытия $\widetilde{P}$ множества S будут следующие свойства:

$$v, w \in \mathscr{N}, \forall i \in 1..s : \; c_{P_i}(v) = c_{P_i}(w) \Rightarrow c_{\widetilde{P}}(v) = c_{\widetilde{P}}(w)$$

$$v, w \in \mathscr{N}, \exists i \in 1..s : \; c_{P_i}(v) \neq c_{P_i}(w) \Rightarrow c_{\widetilde{P}}(v) \neq c_{\widetilde{P}}(w)$$

Существует итеративная версия схемы кластеризации основных групп графа, в которой начальные алгоритмы вновь выделяют сообщества на основе промежуточного разбиения до тех пор, пока это будет увеличивать модулярность промежуточного разбиения. Такой алгоритм далее обозначается как *CGGCi*. В качестве начальных и финального алгоритма можно использовать RG_k.

Одновременно возмущаемая стохастическая аппроксимация (SPSA)

Стохастическая аппроксимация была введена Роббинсом и Монро в 1951 году [**?**] и затем была использована для решения оптимизационный задач Кифером и Вольфовицем [**?**]. В [**?**] алгоритм стохастической аппроксимации был расширен до многомерного случая. В m-мерном пространстве обычная KW-процедура, основанная на конечно-разностной аппроксимации градиента, использовала $2m$ измерений на каждой итерации (по два измерения на каждую координату градиента). Последовательно Граничин [**?**], Поляк и Цыбаков [**?**] и Спалл [**?**] предложили алгоритм *одновременно возмущаемой стохастической аппроксимации* (*SPSA*), который на

каждой итерации использует всего два измерения. Алгоритм *SPSA* имеет такую же скорость сходимости, несмотря на то, что в многомерном случае (даже при $m \to \infty$) в нем используется заметно меньше измерений [**?**].

Алгоритмы стохастической аппроксимации показали свою эффективность в решении задач минимизации стационарных функционалов. В [**?**] для функционалов, меняющихся со временем были применены метод Ньютона и градиентный метод, но они применимы только в случае дважды дифференцируемых функционалов и в случае известных ограничений на Гессиан функционала. Так же оба метода требуют возможности вычисления градиента в произвольной точке.

Общую схему одновременно возмущаемой стохастической аппроксимации можно представить следующим образом:

1. Выбор начального приближения $\hat{\theta}_0 \in \mathbb{R}^m$, счетчик $n \leftarrow 0$, выбор параметров алгоритма $d \in \mathbb{R} \setminus \{0\}$, $\{\alpha_n\} \subset \mathbb{R}^m$
2. Увеличение счетчика $n \leftarrow n + 1$
3. Выбор вектора возмущения $\Delta_n \in \mathbb{R}^m$, чьи координаты независимо генерируются по распределению Бернулли, дающему ± 1 с вероятностью $\frac{1}{2}$ для каждого значения
4. Определение новых аргументов функционала $\theta_n^- \leftarrow \hat{\theta}_{n-1} - d\Delta_n$ и $\theta_n^+ \leftarrow \hat{\theta}_{n-1} + d\Delta_n$
5. Вычисление значений функционала $y_n^- \leftarrow f(\theta_n^-),\ y_n^+ \leftarrow f(\theta_n^+)$
6. Вычисление следующей оценки

$$\hat{\theta}_n \leftarrow \hat{\theta}_{n-1} - \alpha_n \Delta_n \frac{y_n^+ - y_n^-}{2d} \tag{2}$$

7. Далее происходит либо остановка алгоритма, либо переход на второй пункт

В [**?**, **?**, **?**] рассматривается метод стохастической аппроксимации с постоянным размером шага, в таком случае вместо последовательности $\{\alpha_n\}$ используется единственный параметр $\alpha \in \mathbb{R}^m$, и

следующая оценка вычисляется по следующей формуле вместо (2):

$$\hat{\theta}_n \leftarrow \hat{\theta}_{n-1} - \alpha\Delta\frac{y_n^+ - y_n^-}{2d} \tag{3}$$

Это позволяет эффективно решать проблемы очень плохого начального приближения $\hat{\theta}_0$ и дрейфующей оптимальной точки, когда аргумент, при котором функционал принимает лучшие значения, меняется со временем.

Постановка задачи адаптации параметров алгоритма

Рандомизированный жадный алгоритм имеет один параметр k, в то время как на результаты схемы кластеризации основных групп графа влияют параметр s и параметры начальных и финального алгоритма.

Часто алгоритмы на разных входных данных имеют разные оптимальные параметры, то есть нет одного набора параметров, решающих каждую задачу наилучшим образом. При одних и тех же параметрах некоторые графы будут разбивать хорошо, в то время как другие — критически плохо. Алгоритм *SPSA* показал хорошие результаты в создании адаптивных модификаций алгоритмов, то есть модификаций, подстраивающих параметры под входные данные.

В работе рассматривается применение алгоритма *SPSA* к алгоритмам *RG* и *CGGC* для создания алгоритмов, хорошо выделяющих сообщества на большем количестве графов, чем при основных наборах параметров этих рандомизированных алгоритмов.

Оценка качества

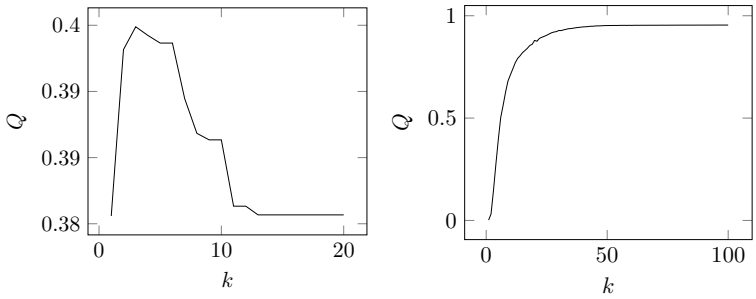
Для оценки качества сообщества используется целевая функция модулярность (1), а в качестве тестовых данных — графы, используемые для оценки алгоритмов на конкурсе 10th DIMACS Implementation Challenge, которые можно найти по адресу `http://www.cc.gatech.edu/dimacs10/archive/clustering.shtml`.

3. Адаптивный рандомизированный жадный алгоритм (ARG)

Применимость алгоритма SPSA

Применимость *SPSA* обоснована теоретически для выпуклой усредненной функции качества.

В зависимости от графа модулярность результатов работы RG_k либо принимает максимум при небольшом k, как на рисунке 1a, либо постепенно увеличивается при росте k, как на рисунке 1b.



(a) Зависимость модулярности от k на графе *karate*

(b) Зависимость модулярности от k на графе *netscience*

Рис. 1: Зависимость модулярности от параметра k на двух графах