САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

Кафедра Информатики

Проданов Тимофей Петрович

Адаптивный рандомизированный алгоритм выделения сообществ в графах

Бакалаврская работа

Допущена к защите.
Зав. кафедрой:

Научный руководитель:
д. ф.-м. н., профессор О.Н. Граничин

Рецензент:
В.А. Ерофеева

Санкт-Петербург
2015

SAINT-PETERSBURG STATE UNIVERSITY
Mathematics & Mechanics Faculty

Department of Computer Science

Timofey Prodanov

Adaptive randomised algorithm for community detection in graphs

Bachelor's Thesis

Admitted for defence.
Head of the chair:

Scientific supervisor:
Professor Oleg Granichin

Reviewer:
Victoria Erofeeva

Saint-Petersburg
2015

Оглавление

Введение

1. Предварительные сведения

1.1. Выделение сообществ в графах

Исторически, изучение сетей происходило в рамках теории графов, которая начала своё существование с решения Леонардом Эйлером задачи о кёнигсбергских мостах. В 1920-х взял своё начало анализ социальных сетей и лишь последние двадцать лет развивается изучение *сложных сетей*, то есть сетей с неправильной, сложной структурой, в некоторых случаях рассматривают динамически меняющейся во времени сложные сети. От изучения маленьких сетей внимание переходит к сетям из тысяч или миллионов узлов.

В процессе изучения сложных систем, построенных по реальным системам, оказалось, что распределение степеней $P(s)$, определённое 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[1]. Это показывает, что модели, предложенные теорией графов, часто будут оказываться далеко от реальных потребностей.

Современное изучение сложных сетей привнесло значительный вклад в понимание реальных систем. Сложные сети с успехом были применены в таких разных областях, как изучение структуры и топологии интернета [2, 3], эпидемиологии [4], биоинформатике [5], поиске преступников [6], социологии [7] и многих других.

Свойством, присутствующим почти у любой сети, является структура сообществ, разделение узлов сети на разные группы узлов так, чтобы внутри каждой группы соединений между узлами много, а соединений между узлами разных групп мало. Способность находить и анализировать подобные группы предоставляет большие возможности в изучении реальных систем, представленных с помощью сложных сетей. Плотно связанные группы узлов в социальных сетях представляют людей, принадлежащих социальным сообществам, плотно сплочённые группы узлов в интернете соответствуют страницам, посвящённым распространённым темам, а сообщества в генетических сетях связаны с функциональными модулями [1]. Таким образом, выделение сообществ в сети является мощным инструментом для понимания функциональности сети.

1.2. Определения и обозначения

Формально, сложная система может быть представлена с помощью графа. В этой 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 мно-

жества $\mathscr{L}$ неупорядоченных пар из множества $\mathscr{N}$, элементы которого называются *рёбрами* или *связями*. Мощности множеств $\mathscr{N}$ и $\mathscr{L}$ равны N и L соответственно.

Подграфом называется граф $G' = (\mathscr{N}', \mathscr{L}')$, где $\mathscr{N}' \subset \mathscr{N}$ и $\mathscr{L}' \subset \mathscr{L}$.

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

Прогулка из узла i в узел j — это последовательность узлов, начинающаяся с узла i и заканчивающаяся узлом j. Путь — это прогулка, в которой каждый узел встречается единожды. Геодезический путь — это кратчайший путь, а количество узлов в нём на один больше геодезического расстояния.

До того, как мы определили понятие *сообщество*, определим *разбиение* на сообщества. Пусть $G = (\mathscr{N}, \mathscr{L})$ — граф, тогда разбиением на сообщества будет называться разбиение множества его вершин $P = \{C_1, \ldots, C_K\}$, то есть $\bigcup_{i=1}^{K} C_i = \mathscr{N}$ и $C_i \cap C_j = \emptyset \ \forall i \neq j \in 1..K$.

Сообщество — это такой подграф, чьи узлы плотно связаны, однако структурная сплочённость узлов можно определить по разному. Одно из определений вводит понятие *клик*. Клик — это максимальный такой подграф, состоящий из трёх и более вершин, каждая из которых связана с каждой другой вершиной из клика. n-клик — это максимальный подграф, в котором самое большое геодезическое расстояние между любыми двумя вершинами не превосходит n. Другое определение гласит, что подграф G' является сообществом, если сумма всех степеней внутри G' больше суммы всех степеней, направленных в остальную часть графа [8]. Сообщества называются смежными, если существует ребро, направленное из вершины первого сообщества в вершину второго.

1.3. Модулярность

Однако подобными определениями сообществ пользоваться неудобно и их проверка достаточно долгая. В 2004 году была представлена *модулярность* — целевая функция, оценивающая неслучайность разбиения графа на сообщества [9]. Допустим, у нас K сообществ, определим тогда симметричную матрицу $\mathbf{e}$ размером $K \times K$. Пусть e_{ij} — отношение количества рёбер, которые идут из сообщества i в сообщество j, к полному количеству рёбер в графе (рёбра l_{mn} и l_{nm} считаются различными, m, n — узлы). След такой матрицы $\mathrm{Tr}\mathbf{e} = \sum_{i \in 1..K} e_{ii}$ показывает отношение рёбер в сети, которые соединяют узлы одного и того же сообщества, и 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 далее определяется вектор $\mathbf{a}$ длины K, элементы которой $a_i = \sum_{j \in 1..K} e_{ij}$, которая обозначает долю количества рёбер, идущих к узлам, принадлежащим сообществу i, к полному количеству рёбер в графе. Если в графе рёбра 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 количество рёбер внутри сообществ не будет отличаться от случайного взятого количества — модулярность будет примерно равна 0. Максимальным возможным значением функции будет 1, но на практике модулярности графов лежат между 0.3 и 0.7.

Было предложено несколько вариаций модулярности [10, 11]. Так, эквивалентным приведённому выше определению будет

$$Q(G, P) = \frac{1}{2L} \sum_{x,y \in 1..N} \left(w_{xy} - \frac{s_x s_y}{2L}\right) \ \delta(c_P(x), c_P(y)), \tag{2}$$

где L — мощность $\mathscr{L}$, w_{xy} — вес ребра между вершинами x и y, s_x и s_y — степени вершин x и y соответственно, δ — символ Кронекера, а отображение $c_P(\cdot)$ указывает, в каком сообществе разбиения лежит узел графа.

Теперь можно поставить задачу выделения сообществ следующим образом: требуется найти такое разбиение графа, что модулярность примет максимальное значение. Можно заметить, что такая постановка не использует какого-либо определения сообществ, и получившиеся разбиение не проверяется на дополнительные свойства, кроме подсчёта модулярности. Однако такая задача всё ещё будет NP-сложной [12].

Преимущество модулярности состоит в том, что для того, чтобы посчитать, какой выигрыш мы извлечем из объединения двух сообществ, необходимо произвести только одну операцию. В рамках определения (1) такой выигрыш будет равен $\Delta Q = 2(e_{ij} - a_i a_j)$, 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. Не умоляя общности, $n_j \leq n_i$, тогда необходимо обновить столбец i-ый столбец и i-ую строку матрицы $\mathbf{e}$, а так же i-ый элемент вектора $\mathbf{a}$: $e_{ik} = e_{ki} = e_{ki} + e_{kj}$, где k — смежное к j сообщество, и $a_i = a_i + a_j$. При этом сообщество j следует удалить из дальнейшего рассмотрения.

Имея матрицу $\mathbf{e}$ и вектор $\mathbf{a}$ не очень важно, как устроен граф и сообщества, что позволяет искать сообщества, основываясь на некотором начальном разбиении, для которого построены $\mathbf{e}$ и $\mathbf{a}$.

1.4. Рандомизированный жадный алгоритм

Ньюман в 2004 году предложил алгоритм, максимизирующий модулярность [13]. Алгоритм начинается с разбиения графа на N сообществ из одной вершины, а затем на каждой итерации просматривает все пары сообществ и соединяет ту пару, которая даст наибольший выигрыш модулярности. Такой алгоритм достаточно долго работает и страдает от несбалансированного объединения сообществ — сообщества растут с разной скоростью, большие кластеры соединяются со своими небольшими соседями независимо от того, выгодно это глобально или нет [14].

Поэтому был предложен рандомизированный жадный алгоритм (RG) [15], который на каждой итерации рассматривал k случайных сообществ и смежных к ним сообществ, а затем так же соединял пару, дающую наибольший выигрыш. Трудоёмкость такого алгоритма примерно равна $O(L \ln N)$. И первый алгоритм, и его рандомизированная вариация соединяют сообщества, записывая только номера соединений, до тех пор, пока не останется только одно сообщество, а затем создают разбиение из списка соединений до того момента, когда достигалась максимальная модулярность (так как в результате лучшего соединения модулярность может уменьшиться).

Можно отметить, что таким алгоритмом можно кластеризовать не только граф, но и граф с некоторым начальным разбиением, в котором можно сообщества разбивать дальше, но нельзя их соединять. При этом только немного поменяется начальный этап инициализации матрицы $\mathbf{e}$ и вектора $\mathbf{a}$ (смотри Алгоритм 1).

1.5. Ансамблевая стратегия

Овельгённе и Гейер-Шульц в 2012 году выиграли 10th DIMACS Implementation Challenge с ансамблевой стратегией выделения сообществ (ES). Ансамблевая стратегия заключается в том, что сначала s начальных алгоритмов разбивают граф на сообщества, и считается, что те вершины, в которых начальные алгоритмы сошлись во мнении определены по сообществам правильно, а те, которые остались, распределяет по сообществам финальный алгоритм [16].

Формализовать это можно следующим образом:

1. Создать множество S из s разбиений G с помощью начальных алгоритмов
2. Создать разбиение $\hat{P}$, равное максимальному перекрытию разбиений из множества S
3. Финальным алгоритмом создать разбиение $\widetilde{P}$ графа G на основе разбиения $\hat{P}$

Необходимо определить понятие *максимальное перекрытие*. Пусть у нас есть множество $S = \{P_1, \ldots, P_s\}$, $c_P(v)$ указывает, в каком сообществе находится узел v с

Входные данные: Невзвешенный неориентированный граф $G = (\mathscr{N}, \mathscr{L})$, параметр k

Выходные данные: Разбиение на сообщества P

for $i \in 1..N$ **do**
  for $j \in 1..N$ **do**
    if i и j смежные **then**
      $e[i, j] = 1/(2 * L)$;
    else
      $e[i, j] = 0$;
    end
  end
  $a[i] = \sum_j e[i, j]$;
end
$global\Delta Q \leftarrow 0$;
$max\_global\Delta Q \leftarrow -\infty$;

for $i \in 1..N$ **do**
  $max\Delta Q \leftarrow -\infty$;
  for $j \in 1..k$ **do**
    $c1 \leftarrow$ случайное сообщество;
    forall сообщества $c2$, смежные с $c1$ **do**
      $\Delta Q \leftarrow 2 * (e[i, j] - a[i] * a[j])$;
      if $\Delta Q > max\Delta Q$ **then**
        $max\Delta Q \leftarrow \Delta Q$;
        $next\_join \leftarrow (c1, c2)$;
      end
    end
  end
  $joins\_list.push(next\_join)$;
  $global\Delta Q \leftarrow global\Delta Q + max\Delta Q$;
  if $global\Delta Q > max\_global\Delta Q$ **then**
    $max\_global\Delta Q \leftarrow global\Delta Q$;
    $best\_step \leftarrow i$;
  end
  $(c1, c2) \leftarrow next\_join$;
  if количество соседей($c2$) $>$ количество соседей($c1$) **then**
    поменять местами $c1$ и $c2$;
  end
  forall соседи $c3$ сообщества $c2$, где $c3 \neq c1, c2$ **do**
    $e[c3, c1] \leftarrow e[c3, c1] + e[c3, c2]$;
    $e[c1, c3] \leftarrow e[c3, c1]$;
  end
  $e[c1, c1] \leftarrow e[c1, c1] + e[c2, c2] + e[c1, c2] + e[c2, c1]$;
  $a[c1] \leftarrow a[c1] + a[c2]$;
end

$P \leftarrow$ создать разбиение из $joins\_list[1..best\_step]$;

Алгоритм 1: Рандомизированный жадный алгоритм

разбиении P. Тогда у максимального перекрытия $\hat{P}$ множества S будут следующие свойства:

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

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

Ансамблевую стратегию можно итерировать, заставляя начальные алгоритмы разбивать максимальное перекрытие и получившееся максимальное перекрытие до тех пор, пока это будет увеличивать модулярность. В таком случае схема будет выглядеть следующим образом:

1. Инициализировать $\hat{P}$ разбиением из сообществ из одного узла
2. Создать множество S из s разбиений графа G на основе разбиения $\hat{P}$ с помощью начальных алгоритмов
3. Записать в $\hat{P}$ максимальное перекрытие множества S
4. Если P_{best} не существует или оно хуже, чем $\hat{P}$, то присвоить $P_{best} \leftarrow \hat{P}$ и вернуться на второй шаг
5. Финальным алгоритмом создать разбиение $\widetilde{P}$ графа G на основе разбиения P_{best}

1.6. Одновременно возмущаемая стохастическая аппроксимация

Стохастические аппроксимация была введена Роббинсом и Монро в 1951 году [17] и затем была использована для решения оптимизационный задач Кифером и Вольфовицем (KW) [18]. В [19] 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). Спалл предложил алгоритм *одновременно возмущаемой стохастической аппроксимации* (SPSA) [20], который на каждой итерации использует всего два измерения. Он показал, что SPSA алгоритм имеет такую же скорость сходимости, несмотря на то, что в многомерном случае (даже при $m \to \infty$), несмотря на то, что в нём используется заметно меньше измерений [21].

Стохастическая аппроксимация первоначально использовалась как инструмент для статистических вычислений и в дальнейшем разрбатывалась в рамках отдельной ветки теории управления. На сегодняшний день стохастическая аппроксимация имеет большое разнообразие применений в таких областях, как адаптивная обработка сигналов, адаптивное выделение ресурсов, адаптивное управление.

Алгоритмы стохастической аппроксимации показали свою эффективность в решении задач минимизации стационарных функционалов. В [22] 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 начальной центральной точки $\theta_0 \in \mathbb{R}^m$, счётчик $n = 0$, выбор параметров алгоритма $d \in \mathbb{R} \setminus \{0\}$, $\{\alpha_n\} \subset \mathbb{R}^m$
2. Увеличение счётчика $n \to n + 1$
3. Выбор вектора возмущения $\Delta_n \in \mathbb{R}^m$, чьи координаты независимо генерируются и в среднем дают ноль. Часто для генерации компонент вектора используют распределение Бернулли, дающее ± 1 с вероятностью $\frac{1}{2}$ для каждого значения
4. Определение новых аргументов функции $\theta_n^- = \hat{\theta}_{n-1} - d\Delta_n$ и $\theta_n^+ = \hat{\theta}_{n-1} + d\Delta_n$
5. Вычисление значений функционала $y_n^- = f(\theta_n^-), y_n^+ = f(\theta_n^+)$
6. Вычисление следующей центральной точки
$$\hat{\theta}_n = \hat{\theta}_{n-1} - \alpha_n \frac{y_n^+ - y_n^-}{|\theta_n^+ - \theta_n^-|}$$
7. Далее происходит либо остановка алгоритма, либо переход на второй пункт

В [23] был представлен метод стохастической аппроксимации с константным размером шага, в таком случае вместо последовательности $\{\alpha_n\}$ используется единственный параметр $\alpha \in \mathbb{R}^m$, и следующая центральная точка вычисляется по следующей формуле: $\hat{\theta}_n = \hat{\theta}_{n-1} - \alpha \frac{y_n^+ - y_n^-}{|\theta_n^+ - \theta_n^-|}$

Список литературы

[1] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.

[2] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999.

[3] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1):309–320, 2000.

[4] Cristopher Moore and Mark EJ Newman. Epidemics and percolation in small-world networks. *Physical Review E*, 61(5):5678, 2000.

[5] Jing Zhao, Hong Yu, Jianhua Luo, ZW Cao, and Yixue Li. Complex networks theory for analyzing metabolic networks. *Chinese Science Bulletin*, 51(13):1529–1537, 2006.

[6] Wang Hong, Wang Zhao-wen, Li Jian-bo, and Qiu-hong Wei. Criminal behavior analysis based on complex networks theory. In *IT in Medicine & Education, 2009. ITIME'09. IEEE International Symposium on*, volume 1, pages 951–955. IEEE, 2009.

[7] John Scott. *Social network analysis*. Sage, 2012.

[8] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[9] Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, Feb 2004.

[10] Stefanie Muff, Francesco Rao, and Amedeo Caflisch. Local modularity measure for network clusterizations. *arXiv preprint cond-mat/0503252*, 2005.

[11] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[12] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2):172–188, 2008.

[13] Mark E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, Jun 2004.

[14] Michael Ovelgönne and Andreas Geyer-Schulz. A comparison of agglomerative hierarchical algorithms for modularity clustering. In *Challenges at the Interface of Data Analysis, Computer Science, and Optimization*, pages 225–232. Springer, 2012.

[15] Michael Ovelgönne and Andreas Geyer-Schulz. Cluster cores and modularity maximization. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1204–1213. IEEE, 2010.

[16] Michael Ovelgönne and Andreas Geyer-Schulz. An ensemble learning strategy for graph clustering. *Graph Partitioning and Graph Clustering*, 588:187, 2012.

[17] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[18] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[19] Julius R Blum. Multidimensional stochastic approximation methods. *The Annals of Mathematical Statistics*, pages 737–744, 1954.

[20] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[21] James C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

[22] Boris T. Polyak. *Introduction to optimization*. Optimization Software New York, 1987.

[23] Oleg Granichin and Natalia Amelina. Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances. *IEEE Transactions on Automatic Control*, 60(5), 2015.