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

1. Адаптивный рандомизированный жадный алгоритм

1.1. Применимость алгоритма SPSA

Для того, чтобы алгоритм SPSA был применим — необходимо иметь выпуклую функцию качества, которую необходимо минимизировать. В большинстве модулярность результатов работы рандомизированного жадного алгоритма на разных графах с разными значениями параметра k будет выглядеть следующим образом:

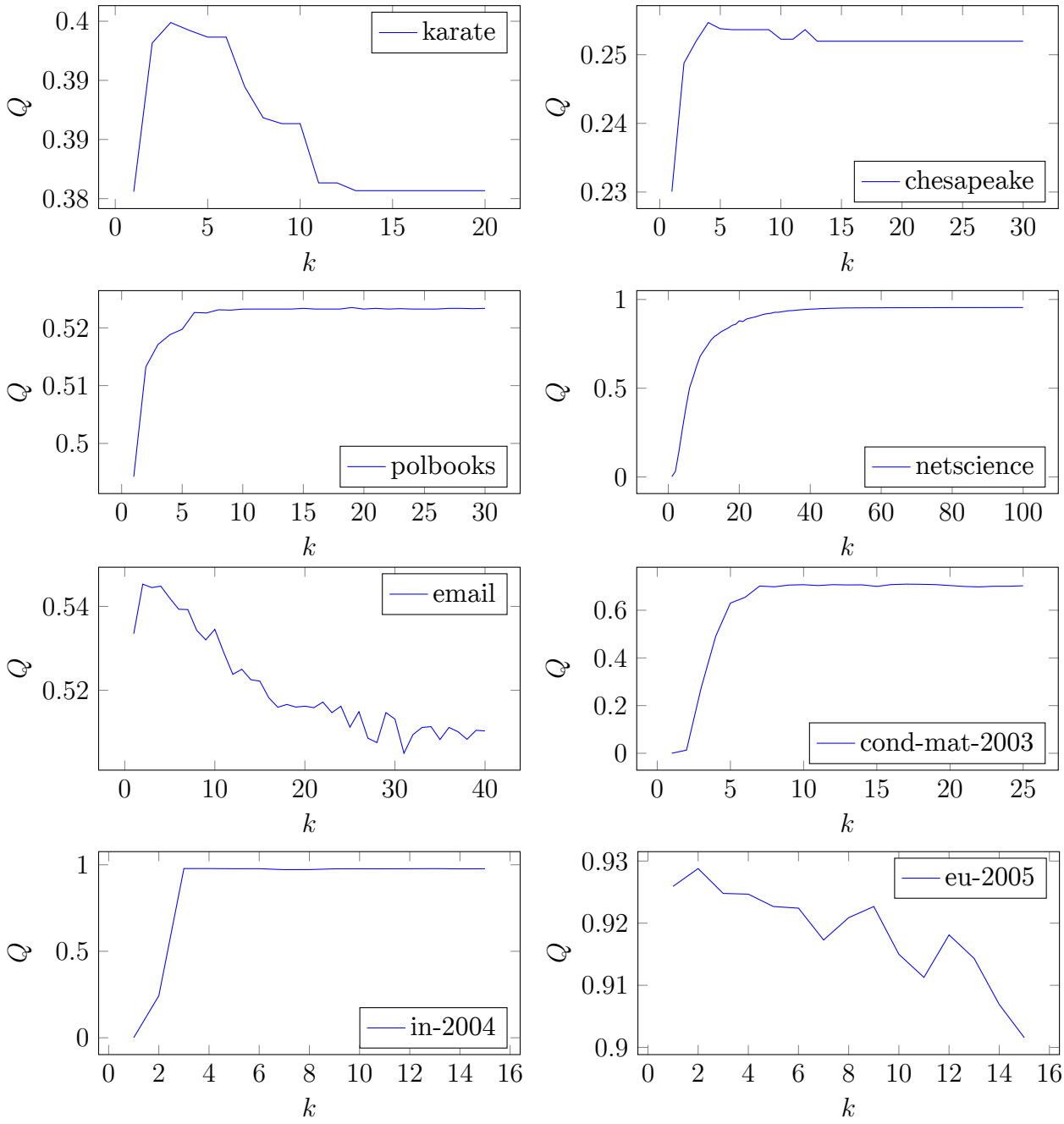

Рис. 1: Зависимость модулярности от k для рандомизированного случайного алгоритма на разных графах

Результаты показывают разделение поведения алгоритма на разных графах на два возможных случая: в первом алгоритм принимает наилучший результат при некотором небольшом, но разном k_{best} (например работа алгоритма на графе *karate*). Во втором результаты алгоритма постепенно возрастают, приближаясь к некоторой асимптоте (хорошим примером будет работа алгоритма на графе *netscience*. К первому случаю так же относится такое поведение, в котором алгоритм быстро (с ростом k) достигает своего лучшего значения, и затем его результаты очень несильно падают, и в дальнейшем держатся того же значения (такое выполняется, например, на графе *in-2004*).

Похожее поведение алгоритм показывает на автоматически сгенерированных графах, но в таких графах можно получить более отличающиеся значения k_{best} и предположить, что будет происходить с модулярностью при дальнейшем росте k.

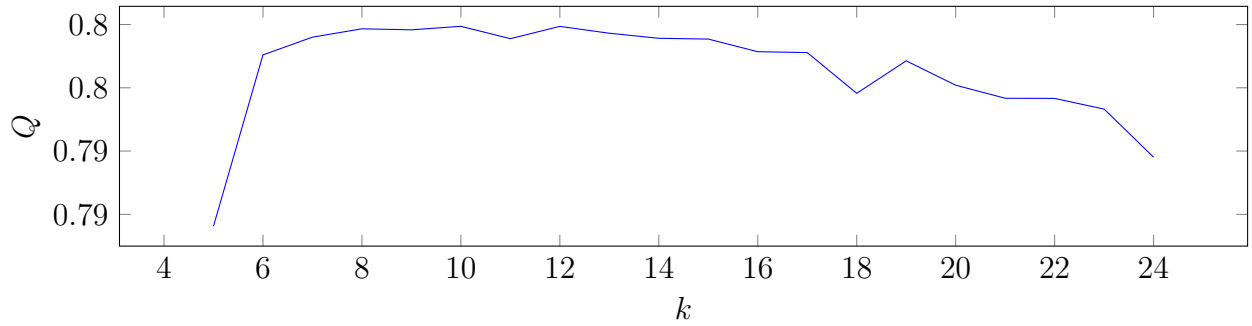

Рис. 2: Зависимость модулярности от k для рандомизированного случайного алгоритма на автоматически сгенерированном графе с параметрами $N = 40.000$, $K = 40$, $p_1 = 0.1$, $p_2 = 5 \cdot 10^{-4}$

1.2. Функция качества

Таким образом, имеет смысл использовать в функции качества не только модулярность, но и время. Подсчёт времени сам по себе занимает время и в разных случаях может давать сильно отличающиеся результаты. Теоретическая трудоёмкость алгоритма линейно зависит от параметра k, на реальных графах зависимость тоже близка к линейной:

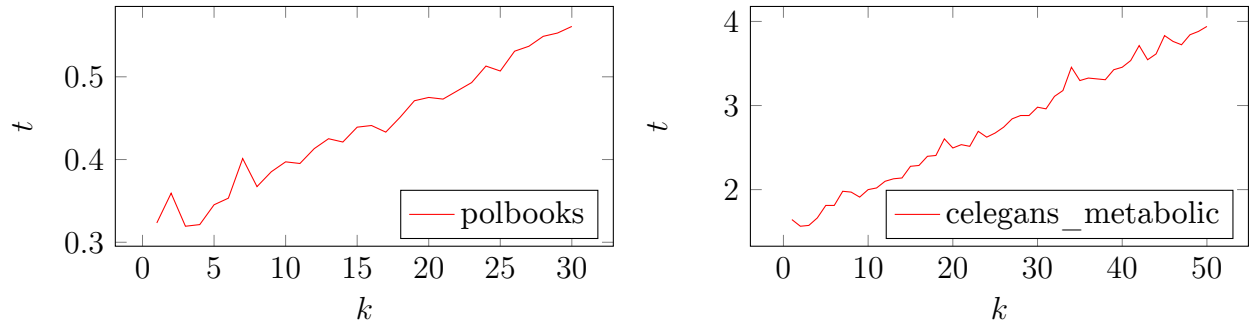

Рис. 3: Зависимость времени t в миллисекундах от k для рандомизированного случайного алгоритма на графах *polbooks* и *celegans_ metabolic*

Таким образом, вместо времени можно использовать значение k. Максимальное значение модулярности, которое может быть достигнуто на графе очень сильно отличается, поэтому нет смысла использовать абсолютное значение модулярности, но имеет смысл использовать относительные значения. При вычислении центрального значения (**??**) в алгоритме одновременно возмущаемой стохастической аппроксимации используется только разность функций качества. Таким образом, если использовать в функции качетсва логарифмы от модулярности — разность функций укажет, во сколько раз модулярность изменилась.

Так же функция качества должна принимать минимум, а не максимум, поэтому первой версией подобной функции может быть $f(Q) = -\alpha \ln Q,\ \alpha > 0$. Для того, чтобы принимать во внимание время работы, разумно добавить логарифм от k: $f(Q, k) = -\alpha(\ln Q - \beta \ln k),\ \alpha > 0, \beta \geq 0$. Коэффициент β в таком случае можно рассматривать в следующем виде $\beta = \frac{\ln \gamma}{\ln 2}$, где коэффициент β указывает, во сколько раз необходимо увеличиться модулярности для того, чтобы покрыть увеличение времени (то есть k) вдвое. Если время не имеет значение коэффициент γ принимает значение 1, а следовательно коэффициент β принимает значение 0. Коэффициент α же играет роль размера шага при выборе следующей центральной точки.

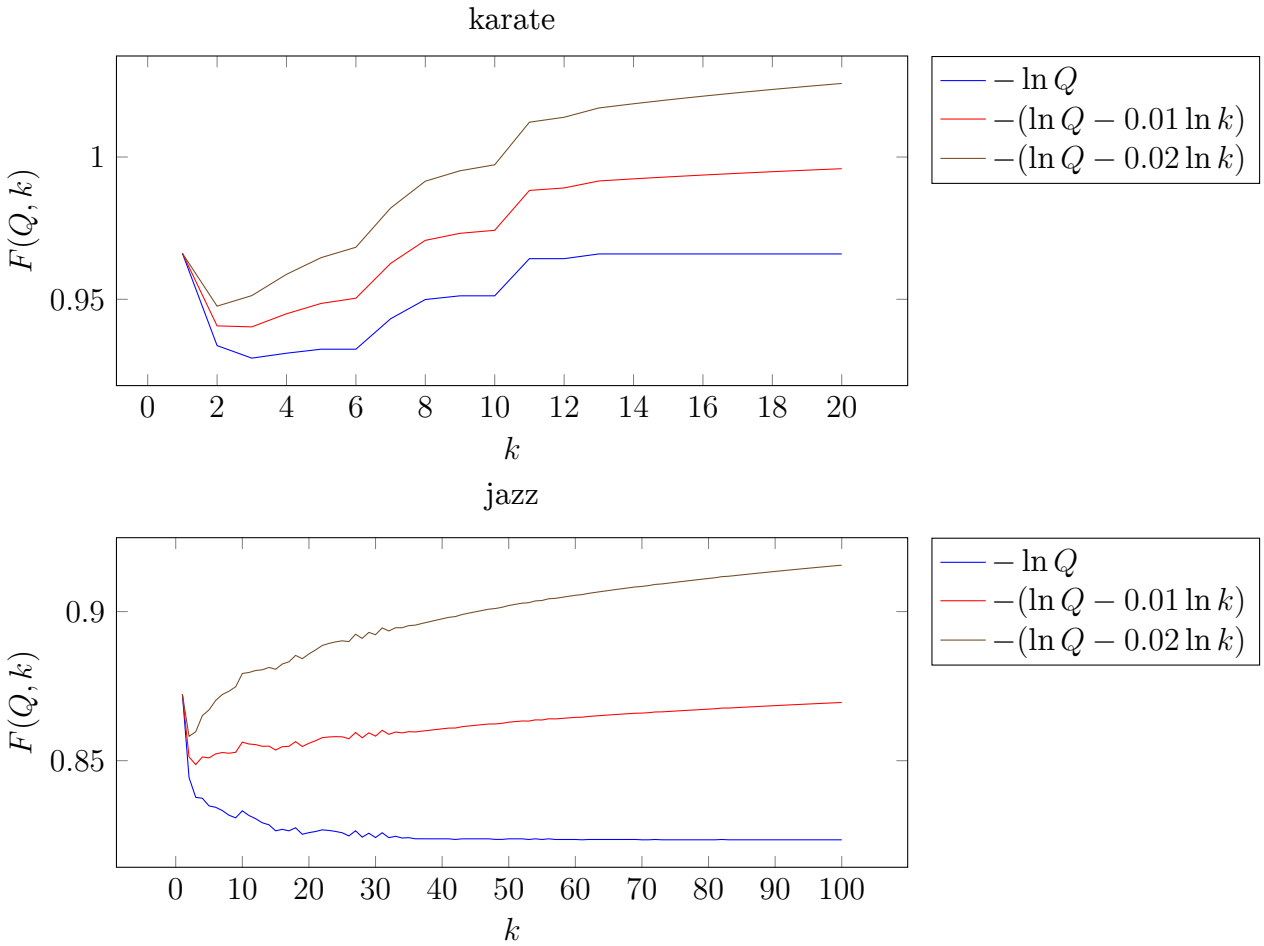

Рис. 4: Функции качества с разными коэффициентами β для графов *karate* и *jazz*

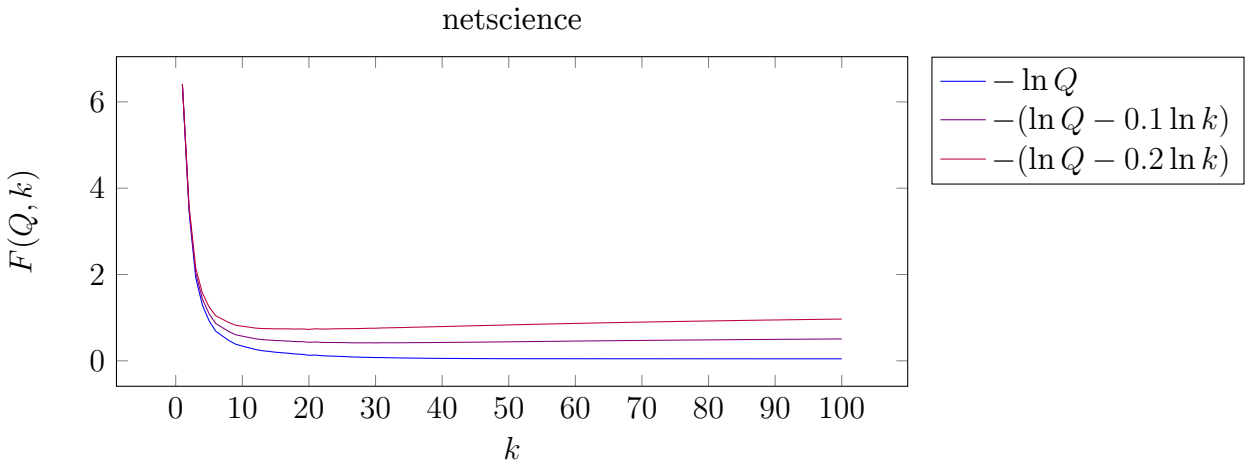

Рис. 5: Продолжение Рис. 4. Функции качества с разными коэффициентами β для графа *netscience*

Как видно из Рис. 5, иногда для того, чтобы функция качества имела минимум на небольших k, надо задавать довольно большое значение коэффициента β. Однако это логично, на данном графе если нас устраивает время работы — выгоднее всё время увеличивать значение k.

1.3. Адаптивный алгоритм

При использовании алгоритма SPSA для рандомизированного жадного алгоритма предлагается разбить действие алгоритма на периоды длиной в σ итераций. В течении каждого периода используется одно значение k. После каждого периода можно считать прирост модулярности, но вместо этого имеет смысл считать медиану прироста модулярности за σ итераций — так как алгоритм рандомизированный, время от времени будут появляться очень хорошие соединения сообществ, которые будут портить функцию качества, такой большой прирост может появиться даже при очень плохом k.

1. Выбор начальной центральной точки $\hat{k}_0 \in \mathbb{N}$, счётчик $n = 0$, выбор размера возбуждения $d \in \mathbb{N}$, коэффициентов функции качества $\alpha, \beta \in \mathbb{R},\ \alpha > 0, \beta \geq 0$, $\sigma \in \mathbb{N}$ — количество итераций в одном периоде

2. Увеличение счётчика $n \to n + 1$

3. Определение новых аргументов функции $k_n^- = \hat{k}_{n-1} - d$ и $k_n^+ = \hat{k}_{n-1} + d$

4. Выполнение σ итераций с параметром k_n^-, вычисление медианы прироста модулярности μ_n^-

5. Выполнение σ итераций с параметром k_n^+, вычисление медианы прироста модулярности μ_n^+

6. Вычисление функций качества $y^- = -\alpha(\ln \mu_n^- - \beta \ln k_n^-)$, $y^+ = -\alpha(\ln \mu_n^+ - \beta \ln k_n^+)$

7. Вычисление следующей центральной точки

$$\hat{k}_n = \hat{k}_{n-1} - \frac{y_n^+ - y_n^-}{k^+ - k^-} \tag{1}$$

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

Список литературы