Конспект к лекции 10. (Санкт-Петербург, 16 апреля 2017 г.)[1]

22 Алгоритм проверки связности графа с использованием логарифмической памятью

Мы уже видели, что зигзаг-произведение позволяет «собирать» из маленьких экспандеров сколь угодно большие экспандеры с ограниченной степенью и достаточно малым вторым собственным числом. Теперь мы рассмотрим ещё одно замечательное одно применение этих операций. Мы докажем теорему о дерандомизации одного из самых знаменитых вероятностных алгоритмов — алгоритма проверки существования пути в неориентированном графе (задача USTCON) с логарифмической памятью.

Задача USTCON: Задан неориентированный граф $G = (V, E)$, в котором выделены две вершины $v, w \in V$. Требуется выяснить, есть ли в графе путь из вершины v в вершину w.

Теорема 22.1 *Задача* USTCON *может быть решена вероятностным алгоритмом с логарифмической памятью.*

Вероятностный алгоритм для решения задачи UPATH устроен очень просто: нужно сделать $N = \mathrm{poly}(|V|)$ (выбор полинома мы уточним чуть позже) шагов случайного блуждания по графу, начав с вершины v. Если за N шагов нам удастся побывать в вершине w, мы точно знаем, что в графе есть путь из v в w. В противном случае мы полагаем, что такого пути нет.

В каждый момент работы алгоритма нам требуется помнить номер текущего шага блуждания (от 1 до N) и номер вершины, в которой мы в данным момент находимся. Для хранения этой информации достаточно памяти размера $O(\log |V|)$.

Ясно, что если пути из v в w нет, то алгоритм выдаст правильный ответ. Остаётся оценить вероятность другой ошибки: путь из v в w существует, но за N шагов блуждания мы его не обнаружим. Без ограничения общности можно считать, что граф регулярен и недвудолен (мы всегда можем добиться этого, добавив в граф некоторое количество петель). Далее покажем, что при случайном блуждании по связному однородному и недвудольному графу распределение вероятностей на вершинах быстро приближается к однородному. Ключевое свойство графа:

Лемма 22.1 *В d-регулярном однородном и недвудольном графе с n вершинами щель между первым и вторым по абсолютной величине собственными числами не может быть меньше $1/\mathrm{poly}(n)$, т.е.*

$$\lambda/d \geqslant 1 - \Theta(1/n^c)$$

[1] Последние исправления: 15.05.2017.

для некоторой константы c (не зависящей ни от n, ни от d).

Доказательство леммы: Прежде всего, без ограничения общности можно считать, что граф связен (если это не так, мы перейдём рассмотрению одной связной компоненты).

Далее, если у графа есть отрицательные собственные числа, мы перейдём от исходного графа G к его квадрату G^2. При возведении в квадрат все собственные числа также возведутся в квадрат и станут положительными. При этом щель между абсолютной величиной первого и второго (по модулю) собственного числа матрицы графа измениться не очень сильно — зазор между d и γd превратится в зазор между d^2 и $(\gamma d)^2$. Важно отметить, что поскольку исходный граф G не был двудольным, в его квадрате максимальное собственное число имеет кратность 1 (связный недвудольный граф при возведении в квадрат остаётся связным).

Таким образом, остаётся доказать лемму для связного графа, у которого все собственные значения положительны. Обозначим $\mathbf{f} = (f_1, \dots, f_n)$ собственный вектор, соответствующий второму собственному числу G^2 (он ортогонален первому собственному вектору $(1, 1, \dots, 1)$, т.е., $\sum f_i = 0$).

Всегда можно считать, что норма $\mathbf{f}$ равна единице. Тогда найдётся координата i такая, что $|f_i| \geqslant 1/\sqrt{n}$. Предположим для определённости, что f_i положительно. Поскольку сумма всех координат $\bar{f}$ равна нулю, то найдётся и координата j, для которой $f_j \leqslant 0$.

Рассмотрим в графе кратчайший путь из i-ой вершины в j-ую:

$$f_i - \ldots - f_j.$$

В этом пути найдётся хотя бы одно ребро $f_l - f_m$, для которого

$$|f_l - f_m| \geqslant |f_i - f_j|/n \geqslant \frac{1}{n\sqrt{n}}$$

Итак, мы нашли в графе такую пару вершин, соединённых ребром, что разница $|f_l - f_m|$ не меньше $1/n^{1.5}$.

Теперь просуммируем $(f_l - f_m)^2$ по всем рёбрам (l, m) графа. При этом каждое ребро мы считаем по одному разу:

$$\sum_{\{l,m\}\in E} (f_l - f_m)^2 = \sum_{\{l,m\}\in E} (f_l^2 + f_m^2 - 2f_l f_m) = d\sum_{s=1}^{n} f_s^2 - \mathbf{f}M\mathbf{f}^{\perp} = d - \lambda$$

(здесь M, как обычно, обозначает матрицу графа, d — его степень). Нетрудно заметить, что данное равенство верно независимо от того, есть ли в графе петли. Данная сумма снизу ограничена $(f_s - f_l)^2 \geqslant \frac{1}{n^3}$. Следовательно, разность $d - \lambda$ ограничена снизу $\Theta(1/n^3)$. Лемма доказана.

С помощью этой леммы мы докажем корректность работы нашего алгоритма. Обозначим $\bar{p}(i)$ распределение вероятностей на вершинах после i шагов случайного блуждания по графу (распределение $\bar{p}(0)$ сосредоточено в единственной вершине s). Пусть обозначим равномерное распределение

$\bar{u} = (\frac{1}{n}, \dots \frac{1}{n})$ на вершинах компоненты связности s, и разложим $\bar{p}(i)$ в сумму $\bar{u}$ и некоторого вектора из его ортогонального дополнения:

$$\bar{p}(i) = \bar{u} + \bar{q}(i),$$

где сумма координат вектора $\bar{q}(i)$ равна нулю. Если $\hat{M}$ — нормализованная матрица графа, то $\bar{q}(i+1) = \hat{M}\bar{q}(i)$. На подпространстве векторов с нулевой суммой норма линейного оператора $\hat{M}$ равна (нормализованному) второму собственному числу графа; по лемме это число не может быть больше $1 - \Theta(1/n^c)$, где n есть число вершин в компоненте связности вершины t. Следовательно, на каждом шаге норма $\bar{q}(i)$ уменьшается по крайней мере в $(1 - \Theta(1/n^c))$ раз, и через $\mathrm{poly}(n)$ шагов распределение $\bar{p}(i)$ станет очень близко к равномерному (на компоненте связности графа). Таким образом, если s и t принадлежат одной компоненте связности, то вероятность попасть через $\mathrm{poly}(n)$ шагов в вершину t будет близка к $1/n$. Если же увеличить число шагов ещё в полином раз, то вероятность хотя бы раз побывать в t станет близка к единице. Теорема доказана.

Далее мы покажем, как дерандомизовать алгоритм случайного блуждание на графе без значительного увеличения используемой памяти. Более формально, мы докажем следующую теорему.

Теорема 22.2 *Задача* USTCON *может быть решена* детерминированным *алгоритмом с логарифмической памятью.*

Прежде чем доказывать теорему, заметим, что мы уже умеем решать на логарифмической памяти задачу USTCON для $(n, d, 0.99)$-экспандеров. В самом деле, мы знаем, что диаметр такого экспандера равен $O(\log n)$. Мы можем перебрать все пути длины $C \log n$ с началом в вершине v и проверить, ведёт ли хотя бы один из них в w; такая проверка очевидно требует лишь логарифмической памяти (и полиномиального времени).

Чтобы решить задачу для произвольного графа G, мы превратим его в экспандер с помощью и зигзаг-произведения

Доказательство теоремы: Мы предполагаем, что нам задан (в виде оракула) неориентированный граф G с n вершинами, без петель и параллельных рёбер. Далее мы построим на основе G несколько «воображаемых» графов, последний из которых будет экспандером с достаточно сильным свойством расширения. Мы сможем моделировать блуждание по каждому из этих воображаемых графов с помощью исходного оракула и дополнительной памяти размера $O(\log n)$.

Воображаемый граф G': заменим в исходном графе каждую вершину v_i степени $d_i > 3$ на цикл длины d_i; рёбра, входившие ранее в данную вершину мы по одному присоединим к вершинам этого цикла. Таким образом, в графе G' степень каждой вершины не превосходит 3. Обозначим через n' число вершин в G' (это число не превосходит $\mathrm{poly}(n)$).

Воображаемый граф G'': Добавим к каждой из вершин G_1 нужное число петель так, чтобы получился D-регулярный граф для некоторого $D = d^4$;

целое число d мы выберем так, чтобы существовал спектральный экспандер H с параметрами $(D = d^4, d, < 0.01)$.

Воображаемые графы G_i : $G_0 = G''$; каждый следующий граф G_{i+1} определяется рекурсивно:

$$G_{i+1} = (G_i ⓩ H)^2.$$

При этом каждый граф G_i будет экспандером с параметрами

$$(n' \cdot D^i, d^4, < 1 - \varepsilon_i)$$

для некоторого ε_i.

Нас интересует значение параметров ε_i (насколько хорошими экспандерами будет построенные графы). Оказывается, что на каждом шаге значение ε будет увеличиваться почти вдвое. В самом деле, произведение $G_i ⓩ H$ будет спектральным экспандером с параметрами $(nD, d^2, 1 - (0.99)^2\varepsilon_i)$ (доказанное ранее свойство зигзаг-произведения). Затем мы берём вторую степень этого графа, и все собственные числа возводятся в квадрат. Для малых x имеем $(1 - x)^2 \approx 1 - 2x$; таком образом, если ε_i достаточно мало, то

$$\varepsilon_{i+1} \approx 2 \cdot 0.99^2 \varepsilon_i \approx 2\varepsilon_i.$$

Применяя Лемму 22.1 к графу G_0, заключаем, что $\varepsilon_0 \geqslant \Omega(1/(n')^c)$. Далее, для каждого следующего G_i значение ε_i становится почти в два раза больше. Следовательно, для $k = \Theta(\log n)$ граф G_i оказывается экспандером, у которого нормализованное второе собственное число по крайней мере не превосходит 0.99.

Вершины G_i получаются как тензорное произведение вершины графа G'' и i копий вершин графа H. Зигзаг-произведение устроено так, что вопрос о существовании пути из v в e в исходном графе G эквивалентен вопросу о существовании пути в G_i из вершин, у которых первая тензорная компонента равна v, в вершины, у которых первая тензорная компонента рана w. Поскольку для $k = \Theta(\log n)$ у графа G_k нормализованное второе собственное число не превосходит 0.99, мы можем проверить данное свойство, перебрав все пути логарифмической длины.

Остаётся заметить, что моделирование блуждания по графу G_k моделируется на логарифмической памяти. В самом деле, для хранения номера вершины G_k нам нужно хранить набор из $(k + 1)$ компонент; самая первое содержит некоторый номер вершины $G_0 = G''$, а каждая следующая — номер одной из вершин H. Ребро в каждом графе G_{i+1} есть путь длины 2 в графе $(G_i ⓩ H)$. Остаётся понять, как организовать рекурсивный вызов для моделирования одного шага по ребру $(G_i ⓩ H)$. Мы предлагаем читателю убедиться, что организация данной рекурсивной процедуры требует лишь $O(1)$ ячеек памяти на каждую компоненту $i = 1, \ldots, k = \Theta(\log n)$. Таким образом, моделирование одного шага блуждания на «воображаемом» графе G_k требует памяти $O(\log n)$.

Упражнение 22.1 *Опишите подробнее рекурсивный алгоритм, моделирующий один шаг блуждания на графе G_k с использованием памяти $O(\log n)$.*

Упражнение 22.2 *Покажите, что для решения следующих задач существуют детерминированные алогоритмы, работающие на логарифмической памяти:*

(а) проверка двудольности графа,

(б) проверка выполнимости булевой формулы вида

$$(x_{i_1} \oplus x_{j_1})\&(x_{i_2} \oplus x_{j_2})\& \ldots \&(x_{i_k} \oplus x_{j_k}),$$

(в) проверка того, что у двух заданных графов одинаковой число компонент связности,

(г) проверка того, что в заданном графе имеется цикл, проходящий через заданное ребро.

23 Надёжные схемы из функциональных элементов

В этом параграфе мы обсуждаем задачу построения надёжных схем из функциональных элементов. Мы считаем, что зафиксирован некоторый конечный *полный базис* булевых функций B, и каждой внутренней вершине схемы сопоставляется некоторая функция $g \in B$, причём арность g совпадает с входной степенью вершины (строго говоря, нужно ещё зафиксировать соответствие между входящими рёбрами и аргументами g). Входным вершинам схемы (вершинам с входной степенью 0) сопоставляются $x_1, \ldots, x_n$ (аргументы функции, которую должна вычислять схема.

Пусть задана схема из N функциональных элементов, вычисляющая некоторую функцию $f : \mathbb{B}^n \to \mathbb{B}$. Рассмотрим работу данной схемы *с ошибками*. Будем предполагать, что каждый из функциональных элементов независимо от других элементов (и от входов схемы) с некоторой вероятностью ε «портится», становится «неисправным». Будем называть данное распределение сбоев *ε-случайным*. При этом мы не предполагаем, что испорченные функциональные элементы *всегда* возвращают неверное значение (отрицание правильного результата вычислений для заданных аргументов). Мы считаем поведение испорченного элемента непредсказуемым — он может возвращать и правильные, и неправильные значения. Можно полагать, что все неисправные элементы схемы переходят во власть злонамеренного противника, который по своему произволу определяет их выходы. При этом выходы на остальных (исправных) функциональных элементах определяются по обычным правилам.

Определение 23.1 *Схема из функциональных элементов (ε, δ)-надёжно вычисляет функцию f, если для любого набора входных значений, при ε-случайном выборе элементов, в которых возникает неисправность, с вероятностью не менее $(1 - \delta)$ схема выдаёт правильное значение функции, как бы ни действовал противник.*

Теорема 23.1 *Для произвольного полного базиса булевых функций B, для всех достаточно малых ε найдётся $\delta = O(\varepsilon)$ такое, что всякая булева функция может быть вычислена (ε, δ)-надёжной схемой в данном базисе.*

Доказательство: Прежде всего заметим, что если теорема верна для одного полного базиса, то она обязана выполняться и для любого другого базиса, быть может с другими ε и δ (поскольку элементы одного базиса можно моделировать блоками, составленными из элементов другого базиса). Без ограничения общности мы можем считать, что наш базис состоит из всех булевых функций трёх аргументов. Мы покажем, что любую обычную булеву схему можно переделать в (ε, δ)-надёжную. Доказательство проведём индукцией по глубине формулы.

Итак, пусть выход (обычной) булевой схемы вычисляется применением функионального элемента b к тройке значений f_1, f_2, f_3. Каждое из значений f_1, f_2, f_3 в свою очередь вычисляются некоторыми подсхемами (быть может, пересекающимися). Глубины этих подсхем заведомо меньше, чем глубина всей схемы; поэтому мы можем считать, что для f_1, f_2, f_3 уже имеются (ε, δ)-надёжные схемы T_1, T_2, T_3. Если к выходам схем T_1, T_2, T_3 применить операцию b, то вероятность получить неверный ответ не превосходит $(3\delta + \varepsilon)$ (итоговый результат может оказаться неверным, если хотя бы одно из значений f_i вычислено неправильно или если неисправность возникла в самом элементе b). Назвоём построенную схему R. Чтобы уменьшить вероятность ошибки, мы изготовим три копии схемы R и применим к выходам этих трёх схем функцию большинства. Вероятность того, что и после этого мы получим ошиочный ответ, не превосходит

$$3(3\delta + \varepsilon)^2 + \varepsilon$$

(ошибка должна случиться хотя бы в двух из трех независимых копий схемы C либо в итоговом вычислении большинства). Для малых ε и подходящего $\delta = O(\varepsilon)$ получаем

$$3(3\delta + \varepsilon)^2 + \varepsilon \leqslant \delta$$

и теорема доказана.

Отметим, что приведённая конструкция может экспоненциально увеличить размер схемы, хотя её глубина увеличивается лишь в константу раз.

Упражнение 23.1 *Докажите, что* функцию голосования

$$majority(x_1, \ldots, x_k) = \begin{cases} 1, & \textit{если более половины } x_i \textit{ равны } 1, \\ 0, & \textit{иначе} \end{cases}$$

можно вычислить схемой глубины $O(\log k)$, состоящей из $\mathrm{poly}(k)$ *функциональных элементов.*

(б)Докажите, функцию голосования $majority(x_1, \ldots, x_k)$ можно вычислить (ε, δ)-надёжной схемой размера $\mathrm{poly}(n)$*.*

Далее мы докажем более сильное утверждение о надёжных вычислениях.

Теорема 23.2 *Для произвольного полного базиса булевых функций B, для всех достаточно малых ε найдётся $\delta = O(\varepsilon)$ такое, что всякая булева схема C из N элементов может быть переделана (за время $\mathrm{poly}(N)$) в (ε, δ)-надёжную схему размера $O(N \log N)$.*

Доказательство: Прежде чем доказывать теорему, введём определение:

Определение 23.2 *Двудольный граф называется (k, d, α, β)-компрессором, если*

1. *в левой и правой долях графа содержится по k вершин;*
2. *степень каждой вершины равна d;*
3. *пусть A — произвольное множество вершин левой доли графа, и $|A| \leqslant \alpha k$; обозначим B множество таких вершин правой доли графа, у которых большинство соседей принадлежат A; тогда размер B не превосходит βk.*

Лемма 23.1 (о компрессоре) *Если $4\alpha(\gamma^2 + \alpha) < \beta < 1/2$, то матрица смежности спектрального (k, d, γ)-экспандера задаёт (k, d, α, β)-компрессор (двудольный граф с $2 \times k$ вершинами также задаётся матрицей $k \times k$).*

Отложим доказательство леммы и покажем, как она помогает доказать теорему. Зафиксируем некоторый параметр k (в последствии мы выберем $k = O(\log N)$). Долее, построим (k, d, α, β)-компрессор такой, что

$$\beta + Const \cdot \varepsilon < \alpha$$

(константа $Const$ не зависит от k и определяется соотношением числа d и базиса, над которым мы строим схему; подробнее значение $Const$ мы обсудим ниже).

Мы преобразовываем заданную нам схему C в эквивалентную ей (ε, δ)-надёжную схему C'. Для этого мы заменим каждый функциональный элемент на некоторый блок из $O(k)$ элементов (устройство такого блока мы сейчас опишем). Если в схеме C выход элемента номер i подавался на вход элементу номер j, то в новой схеме C' от блока номер i к блоку номер j будет идти 'кабель' из k проводов. В идеальной ситуации (когда нет ошибок) сигналы во всех проводах этого кабеля будут одинаковы; более того, это будет ровно тот сигнал, который проходил по соответствующему проводу в исходной схеме (при тех же значениях входов схемы).

Теперь опишем устройство блока, соответствующего одному из элементов схемы C. Мы объясним конструкцию на простейшем примере: пусть в C присутствовал функциональный элемент конъюнкция; наша задача — построить надёжный блок, успешно моделирующий этот функциональный элемент при *умеренном* количестве ошибок. В этот блок будут входить $2k$ сигналов (два кабеля по k проводов). Мы сводим соответствующие провода из этих кабелей (первый с первым, второй со вторым, и т.д.) и для каждой пары вычисляем конъюнкцию. Получаем k результирующих сигналов. Затем пропускаем эти сигналы через *корректор*: это схема с k входами и k выходами; каждый выход вычисляется как *большинство* среди некоторых d входов; а правило, по которому каждому из выходов сопоставляются d входов, есть (k, d, α, β)-компрессор. Отметим, что блок реализуется схемой глубины $O(1)$ и состоит из $O(k)$ функциональных элементов (константы зависят от выбора базиса).

С помощью оценки вероятности больших уклонений (неравенство Чернова) нетрудно показать, что если $k = \Omega(\log N)$, то с большой вероятностью ни в одном из N описанных блоков не случится больше $Const \cdot \varepsilon k$ (число $Const$ определяется глубиной схемы-корректора, т.е. зависит от выбора базиса). В таком случае, если каждый из входных кабелей несет не более αk «неправильных» сигналов (т.е. сигналов, отличных от значения в соответствующем проводе исходной схемы C), то и среди k выходных сигналов не более αk ошибочных. Действительно, перед применением *корректора* неправильные сигналы обоих входов складываются – их может стать 2α. Затем мы пропускаем сигналы через компрессор, и доля ошибок уменьшается до β. Наконец, нужно учесть ещё $O(\varepsilon k)$ новых ошибок, которые могли случиться в самом блоке. Всего на выходе имеем долю ошибок $\beta + O(\varepsilon) < \alpha$.

Чтобы закончить конструкцию, нам нужно вычленить из k-жильного кабеля на выходе последнего блока *один* сингнал с ответом. Для этого нам нужно вычислить *большинство* среди значений этих k сигналов. Это можно делать разными способами; например, можно применить «экспоненциальную» конструкцию надёжной схемы (при вычислении функции большинства среди $O(\log N)$ значений данный метод даст схему размера $\mathrm{poly}(\log N)$, см. упражнение выше).

Доказательство леммы о компрессоре: Пусть $\mathbf{e}_1, \ldots, \mathbf{e}_k$ — ортонормированный собственный базис матрицы M заданного (k, d, γ)-экспандера, а $\lambda_1, \ldots, \lambda_k$ – соответствующие собственные числа. Мы будем считать, что собственные числа упорядочены по убывания абсолютной величины. При этом

$$\mathbf{e}_1 = \frac{1}{\sqrt{k}}(1, 1, \ldots, 1),$$

а $\lambda_1 = d$ (по условию леммы остальные собственные числа по модулю не превосходят γk). Пусть A – некоторое множество вершин графа, и $|A| \leqslant \alpha k$. Обозначим $\mathbf{f} = (f_1, \ldots, f_k)$ характеристический вектор этого множества ($f_i = 1$ если и только если i-ая вершина графа принадлежит A). Ясно, что $||\mathbf{f}||^2 \leqslant \alpha k$. Оценим норму вектора $M\mathbf{f}$.

$$||M\mathbf{f}||^2 = (M\mathbf{f}, M\mathbf{f}) = (\mathbf{f}, M^2\mathbf{f}) = \sum_{i=1}^{k} \lambda_i^2 (\mathbf{f}, \mathbf{e}_i)^2 = \alpha^2 d^2 k + \sum_{i=2}^{k} \lambda_i^2 (\mathbf{f}, \mathbf{e}_i)^2$$

Поскольку все собственные числа кроме первого по модулю не превосходят

γk, получаем

$$||M\mathbf{f}||^2 \leqslant \alpha^2 d^2 k + (\gamma d)^2 \sum_{i=2}^{k} (\mathbf{f}, e_i)^2 \leqslant \alpha^2 d^2 k + (\gamma d)^2 ||\mathbf{f}||^2 \leqslant (\alpha^2 d^2 + \alpha\gamma^2 d^2)k.$$

Далее, для выбранного A мы рассмотрим множество B, которое состоит из всех вершин графа, у которых не менее $d/2$ соседей лежат в A. Это значит, что B состоит из таких вершин $i = 1, \ldots, n$, что в i-ой координате вектора $\mathbf{g} = (M\mathbf{f})$ стоит число не менее $d/2$. Получаем

$$|B| \leqslant \sum_{i=1}^{k} \left(\frac{f_i'}{d/2} \right)^2 \leqslant \frac{4}{d^2} ||M\mathbf{f}||^2 \leqslant 4(\alpha^2 + \alpha\gamma^2)k \leqslant \beta k,$$

и лемма доказана.