Конспект к лекции 7. (Санкт-Петербург, 15 апреля 2017 г.)

17 Рекурсивные конструкции спектральных экспандеров

В этом параграфе мы построим последовательность явно заданных графов одной и той же степени с растущим числом вершин, имеющих малые собственные числа. Основная идея: возводя матрицу в квадрат, мы не меняем число вершин графа и уменьшаем (возводим в квадрат) нормализованное (т.е. делённое на степень графа) второе собственное число. Зато мы увеличиваем (тоже возводим в квадрат) степень вершины. Но это можно скомпенсировать зигзаг-умножением на фиксированный граф H.

Опишем конструкцию более подробно. Начнём с того, что зафиксируем спектральный экспандер H с параметрами $(d^4, d, 1/10)$ для некоторого d (как мы видели, для всех достаточно больших чётных d такой граф существует, и один граф с такими параметрами можно найти перебором). Затем построим последовательность графов $G_0, G_1, \ldots$ следующим образом

- $G_0 = H^2$. Параметры этого экспандера: $(d^4, d^2, 1/100)$.
- G_{n+1} есть зигзаг-произведение G_n^2 и H. По индукции доказывается, что экспандер G_n имеет параметры $(d^{4n+4}, d^2, \leqslant 1/2)$. В самом деле, после возведения G_n в квадрат мы получаем экспандер с параметрами $(d^{4n+4}, d^4, \leqslant 1/4)$. А умножение на H даёт новый граф с числом вершин d^{4n+8}, степенью каждой вершины d^2 и нормализованным вторым собственным числом

$$1 - \alpha\beta^2 \leqslant 1 - (3/4) \cdot (0{,}99)^2 < \frac{1}{2}$$

(см. оценку из параграфа 16), что завершает доказательство.

Мы получили конструкцию экспандера, конторая является эффективной в «слабом» смысле — такие графы можно строить за время полиномиально зависящее от числа вершин. В приложениях нам могут понадобиться экспандеры эффективные в более сильном смысле — графы, для которых по номеру вершины можно эффективно найти список номеров её соседей.

Чтобы более точно определить, что такое эффективная конструкция графа $G = (V, E)$, мы произвольным образом зафиксируем для каждой вершины графа нумерацию инцидентных ей рёбер. *Функцией вращения* называют отображение

$$N \ : \ \langle x, i\rangle \to y,$$

которое сопоставляет вершине графа $x \in V$ и номеру i (не превосходящему степени вершины x) вершину $y \in V$, которая является i-ым соседом x (выйдя из x по i-ому ребру, мы попадём в вершину $y = N(x, i)$).

Если число вершин в графе не превосходит 2^n, а степень каждой вершины равна 2^d, то каждую вершину можно задавать n-битным индексом, а номер выходящего из вершины ребра, соответственно, d-битным индексом. Таким образом, функцию вращения можно понимать как отображение

$$N \ : \ \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n.$$

Сильная эффективность означает, что время вычисления $N(x,i)$ полиномиально зависит от длины аргументов, т.е., от от логарифма числа вершин и от логарифма степени графа. (Для сравнения: в «слабо эффективной» конструкции графа функция вращения вычислима за время полиномиально зависящее от самого числа вершин графа, а не от его логарифма).

Как происходит вычисление функции N в построенной нами последовательности графов? Номер ребра представляет собой пару чисел, каждое от 1 до d. Вершина графа G_{n+1} представляет собой пару: одна вершина G_n^2 (=вершина G_n) и одна вершина H. Движение по ребру: сначала идём по ребру H, попадаем в какую-то вершину H (в диапазоне $1 \dots d^4$), воспринимаем её как пару чисел в диапазоне $1 \dots d^2$, проходим по двум рёбрам графа G_n и затем делаем ещё один проход по ребру H. Таким образом, вычисление N для графа G_{n+1} использует два вызова аналогичного вычисления для G_n, что приводит к экспоненциальной оценке по n. Таким обрзом, полиномиальной вычислимости функции N не получается.

Однако можно модифицировать конструкцию, используя не предыдущий граф G_n, а граф с половинным индексом. Для начала выберем граф H с параметрами $(d^8, d, 1/10)$, а затем построим последовательность графов

$$\begin{aligned}
&G_0 \colon (1, d^2, \leqslant 1/2)\\
&G_1 \colon (d^8, d^2, \leqslant 1/2)\\
&G_2 \colon (d^{16}, d^2, \leqslant 1/2)\\
&\quad \dots\\
&G_n \colon (d^{8n}, d^2, \leqslant 1/2)\\
&\quad \dots
\end{aligned}$$

Начальные графы G_0 и G_1 построить легко (G_0 — это граф, состоящий из единственной вершины и d^2 петель, граф G_1 можно получить из H размножением рёбер в d раз, что не меняет собственных чисел). Затем можно воспользоваться рекуррентной формулой

$$G_n = (G_{\lfloor (n-1)/2 \rfloor} \otimes G_{\lceil (n-1)/2 \rceil})^2 \textcircled{z} H,$$

где $\otimes$ обозначает тензорное произведение, а $\textcircled{z}$ — зигзаг-произведение. Индукцией по n можно показать, что графы G_n являются спектральными экспандерами с параметрами $(d^{8n}, d^2, \leqslant 1/2)$.

Преимущество новой конструкции в том, что при вычислении функции вращения два рекурсивных вызова относятся к половинным значениям n;

глубина рекурсии теперь стала логарифмической по n, а значит число всех рекурсивных вызовов ограничено $\mathrm{poly}(n)$. Таким образом, общее время вычисления полиномиально зависит от n.

18 Случайное блуждание на экспандерах (2-ая часть)

Ранее (см. параграф 13) мы доказали следующую оценку для спектра индуцированного подграфа спектрального экспандера:

Теорема 18.1 *Пусть граф G является спектральным (n, d, γ)-экспандером и A — некоторое множество вершин графа, состоящее из αn вершин (для некоторого $\alpha > 0$). Тогда все собственные числа индуцированного подграфа*[1] *на вершинах A не превосходят $(\gamma + \alpha(1-\gamma))d$.*

Эта теорема помогла нам в изучении блуждания по экспандеру, мы получили следующее утверждение:

Утверждение 18.1 *Пусть граф G является спектральным (n, d, γ)-экспандером и A — некоторое множество вершин графа, состоящее из αn вершин. Рассмотрим случайное блуждание по графу*

$$x_0 - x_1 - \ldots - x_t,$$

где вершина x_0 выбирается случайно (по равномерному распределению), а затем на каждом шаге $i = 1, \ldots, t$ следующая вершина x_i выбирается случайно (также равномерно) среди всех соседей x_{i-1}. Тогда

$$\mathrm{Prob}[x_i \in A \text{ для всех } i] \leqslant (\alpha + \gamma(1-\alpha))^t.$$

Доказательство данного утверждения (с минимальными измененениями) можно использовать для получения более общих результатов.

Обобщение 1: Рассмотрим случайное блуждание по экспандеру, состоящее из $k = 2t$ шагов,

$$x_0 - x_1 - \ldots - x_{2t}.$$

Как и раньше, вершина x_0 выбирается случайно (по равномерному распределению), а затем на каждом шаге $i = 1, \ldots, 2t$ следующая вершина x_i выбирается случайно (также равномерно) среди всех соседей x_{i-1}. Будем интересоваться вероятностью того, что все вершины с чётными номерами, т.е., $x_0, x_2, x_4, \ldots, x_{2t}$, попали в множество A. Вероятность этого события оценивается следующим образом:

$$\mathrm{Prob}[x_i \in A \text{ для всех чётных } i] \leqslant (\alpha + \gamma^2(1-\alpha))^t \leqslant (\alpha + \gamma(1-\alpha))^t.$$

[1] В индуцированном подграфе G_S множество вершин совпадает с A (некоторым подмножеством вершин исходного графа G), а в качестве рёбер берутся все рёбра графа G, оба конца которых принадлежат A.

В самом деле, нужно применить утверждение 18.1 не к исходному графу G, а к его 2-ой степени (к графу на n вершинах, рёбрами которого являются пути длины 2 в G).

Обобщение 2: Снова рассмотрим случайное блуждание по экспандеру, состоящее из t шагов,

$$x_0 - x_1 - \ldots - x_t.$$

На этот раз оценим вероятностью того, что в множество A попали все вершины с «контролируемыми» номерами $x_{i_1}, x_{i_1+i_2}, x_{i_1+i_2+i_3}, \ldots, x_{i_1+i_2+\ldots+i_r}$. Вероятность этого события не превосходит

$$(\alpha + \gamma^{i_1}(1-\alpha)) \cdot (\alpha + \gamma^{i_1}(1-\alpha)) \cdot \ldots \cdot (\alpha + \gamma^{i_r}(1-\alpha)) \leqslant (\alpha + \gamma(1-\alpha))^r.$$

В самом деле, в рассуждение из замечания 1 легко переносится на случай, когда расстояние между «контролируемыми» номерами шагов x_j варьируются. Таким образом, мы умеем доказывать следующий результат:

Утверждение 18.2 *Пусть граф G является спектральным (n, d, γ)-экспандером и A — некоторое множество вершин графа, состоящее из αn вершин. Рассмотрим случайное блуждание по графу*

$$x_0 - x_1 - \ldots - x_t,$$

где вершина x_0 выбирается случайно и равномерно, а затем на каждом шаге $i = 1, \ldots, t$ следующая вершина x_i выбирается случайно равномерно среди всех соседей x_{i-1}.

Пусть $I \subset \{0, \ldots, t\}$ некоторое подмножество номеров шагов. Тогда

$$\mathrm{Prob}[x_i \in A \text{ для всех } i \in I] \leqslant (\alpha + \gamma - \alpha\gamma)^{|I|-1}.$$

Мы будем применять это утверждение при $\alpha < 1/4$ и достаточно малых γ.

19 Блуждание на экспандере как генератор псевдослучайных битов.

Предположим, что для решения некоторой задачи имеется полиномиальный вероятностный алгоритм, который на любом входе x с вероятностью не менее $1 - \delta$ возвращает правильный ответ. Параметр δ (предположим, что $\delta < 1/4$) называется вероятностью ошибки алгоритма. Чтобы уменьшить вероятность ошибки алгоритма, можно последовательно выполнить имеющийся алгоритм t раз на независимых значениях датчика случайных битов, а затем из полученных t результатов выбрать наиболее часто случающийся. У нового алгоритма вероятностью ошибки не будет превосходить c^t для некоторого $c < 1$. Таким образом, сделав число итераций t достаточно большим, можно сделать вероятность ошибки меньше любого наперёд заданного числа. Можно даже сделать вероятность ошибки экспоненциально убывающей (с ростом длины входа), если взять число итераций $t = t(n)$

сравнимым с длиной входа. При этом время работы алгоритма будет оставаться полиномиальным. Очевидным недостатком этого подхода является рост числа используемых случайных битов — их число умножается на t.

Мы покажем, что существует альтернатива простому повторению исходного алгоритма на независимых наборах случайных битов. Данный подход позволит значительно уменьшить вероятность ошибки и при этом незначительно увеличить расход случайные биты. Для этого мы будем генерировать «псевдослучайные» биты с помощью экспандеров. Набор псевдослучайных битов можно будет вырастить из короткого «зерна» — небольшого набора настоящих случайных битов. При этом полученные псевдослучайные биты, как мы увидим, можно использовать для независимого запуска многих копий вероятностного алгоритма, (почти) как если бы они были по-настоящему случайными независимыми.

В этом параграфе мы рассматриваем алгоритмы с двусторонней ошибкой — считаем, что вероятностный алгоритм может выдавать как положительные, так и отрицательные ложные ответы. (При этом мы полагаем, что для любого входа вероятность ошибки ограничена некоторым $\delta < 1/4$.)

Обозначим через k количество случайных битов, оторое требовалось исходному вероятностному алгоритму (при работе со входами длины n). Рассмотрим спектральный $(2^k, d, \gamma)$-экспандер. Определим следующий случайный процесс: выберем случайно (по равномерному распределению) исходную вершину графа x_0, а затем сделаем t шагов случайного блуждания по графу,

$$x_0 - x_1 - \ldots - x_t,$$

на каждом шаге выбирая случайного соседа x_{i+1} предыдущей вершины x_i. Затем запустим $t+1$ копию старого алгоритма, используя индексы вершин $x_0, x_1, \ldots, x_t$ как наборы случайных битов. Среди полученных ответов выберем самый часто встречающийся и объявим его результатом работы нового алгоритма.

Утверждение 18.2 позволяет оценить вероятность ошибки нового алгоритма. Она не превосходит

$$\sum_{I \subset \{0,\ldots,t\},\ |I|>t/2} (\delta+\gamma-\delta\gamma)^{|I|-1} \leqslant 2^{t+1}(\delta+\gamma-\delta\gamma)^{(t-1)/2} = O\left(\left(\sqrt{4(\delta+\gamma-\delta\gamma)}\right)^t\right),$$

т.е., для $\delta < 1/4$ и достаточно малых γ вероятность ошибки будет экспоненциально убывать с ростом t. При этом случайное блуждание длины t в d-регулярном графе задается $k + t \log d$ случайными битами (случайная первая вершина и t переходов от текущей вершины к случайному соседу). Это значительно меньше, чем $t \cdot k$ битов, нужных для выбора t по-настоящему независимых наборов по k битов.

Упражнение 19.1 *Предположим, что исходный вероятностный алгоритм ошибался с вероятностью не более δ в ответах «да» и никогда не ошибался в ответах «нет»* (*вероятностный алгоритм с* односторонней *ошибкой*).

Докажите, что метод последовательного повторения исходного алгоритма на «псевдо-независимых» наборах случайных битов, соответствующих случайному блужданию на графе

$$x_0 - x_1 - \ldots - x_t,$$

(*аналогичный описанной выше конструкции для алгоритмов с двусторонней ошибкой*) *уменьшает итоговую вероятность ошибки до* $\delta \cdot (\delta + \gamma - \delta\gamma)^t$.

Упражнение 19.2 *Рассмотрим следующий алгоритм, тестирующий число* n *на простоту:*

1. *если* $n = 2$, *то объявляем* n *простым*
2. *если* n *чётно и не равно* 2, *то объявляем* n *составным*
3. *для нечётных* $n > 2$ *повторяем следующую процедуру для* $i = 1, \ldots, t$
   - 3.1. *выбираем случайное целое* y_i *от* 2 *до* $n - 1$;
   - 3.2. *если НОД*$(y_i, n) \neq 1$ *хотя бы для одного* i, *то объявляем* n *составным*
   - 3.3. *вычисляем* $z_i = y_i^{\frac{n-1}{2}} \mod n$
4. *если все значения* z_i *принадлежат* $\{1, n-1\}$, *причём оба значения* 1 *и* $n-1$ *встречаются хотя бы по одному разу, то объявляем* n *простым; иначе объявляем* n *составным*

(а) Покажите, что данный алгоритм выполняется за время, полиномиально зависящее от величны параметра t *и от длины двоичной записи входа* n.

(б) Докажите, что вероятность ошибки данного алгоритма не превосходит $O(c^{-t})$ *для некоторого* $c < 1$.