Math-Net.Ru

Общероссийский математический портал

Б. Я. Рябко, А. А. Федотов, Оценка средней эффективности поисковых деревьев, построенных для произвольных наборов двоичных слов, *Дискрет. матем.*, 2002, том 14, выпуск 1, 142–150

DOI: <https://doi.org/10.4213/dm231>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.85.183.62

26 апреля 2023 г., 16:23:13

Оценка средней эффективности поисковых деревьев, построенных для произвольных наборов двоичных слов

© 2002 г. Б. Я. Рябко, А. А. Федотов

Рассматривается задача построения двоичного поискового дерева для произвольного множества двоичных слов, которая находит широкое применение в информатике, в биологии и минералогии при построении определительных таблиц, а также в других областях. Известно, что задача построения дерева минимальной стоимости NP-полна, поэтому возникает задача поиска простых алгоритмов построения деревьев, близких к оптимальным. В статье показано, что даже простейший алгоритм строит поисковые деревья, в среднем близкие к оптимальным, и доказано, что среднее количество проверяемых узлов в оптимальном дереве отличается от естественной нижней границы — двоичного логарифма числа слов — не более, чем на 1,04.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 98-01-00772.

1. Введение

Поисковое дерево — это форма записи алгоритма поиска в заданном множестве двоичных слов, представленном в виде бинарного дерева. Листьям этого дерева сопоставлены определяемые слова, а развилкам — номера проверяемых позиций. Алгоритм определения слова с помощью поискового дерева можно представить как движение по этому дереву от корня к одному из листьев. При этом направление движения в развилке выбирается в зависимости от соответствующего разряда определяемого двоичного слова.

На рисунке 1 приведены два дерева для поиска в множестве двоичных слов $\{000, 010, 110, 111\}$. Поясним, как происходит поиск по левому дереву. Сначала проверяется первый двоичный разряд рассматриваемого слова. Если он равен нулю, то мы идем по левой ветви дерева и проверяем второй двоичный разряд, в противном случае мы проверяем третий разряд слова. Таким образом, за две проверки мы всегда определяем заданное слово.

Двоичные поисковые деревья находят широкое применение на практике, например, в базах данных, а также при создании различных определительных таблиц. Поэтому задачи построения таких деревьев привлекали внимание многих исследователей [1, 2, 3]. Естественно возникает задача построения в том или ином смысле

Рис. 1. Пример поисковых деревьев с разным средним числом проверяемых букв

наилучшего дерева поиска для заданного множества двоичных слов. Критерием качества построенного дерева принято считать стоимость поиска, которая равна среднему числу проверяемых букв, требуемых для определения одного слова. Естественно искать дерево с минимальным средним числом проверяемых букв, которое называют оптимальным. Так, среднее число проверяемых букв для определения одного слова у левого дерева на рисунке 1 равно 2, в то время как у правого — 2,25. В теории информации известно, что это число у любого дерева не может быть меньше $\log_2 m$, где m — число определяемых слов. Таким образом, левое дерево является оптимальным.

Известно, что задача построения оптимального поискового дерева по заданному множеству слов NP-полна [4]. Неформально говоря, это означает, что данная задача практически неразрешима при достаточно большом числе слов. Поэтому возникает задача нахождения быстрых алгоритмов, строящих определительные деревья, в том или ином смысле близкие к оптимальным.

В статье показано, что даже простейший алгоритм построения поискового дерева, который случайно выбирает очередной разряд для проверки, дает, как правило, результат, близкий к оптимальному. Точнее, почти для каждого исходного набора слов этот простейший метод дает дерево, стоимость которого близка к минимальной. Показано также, что в среднем (по всем наборам начальных данных), стоимость поиска по дереву, построенному простейшим алгоритмом и, следовательно, стоимость поиска по оптимальному дереву, не превосходит $\log_2 m + 1,04$.

2. Постановка задачи и основной результат

Пусть дано множество из m двоичных слов длины n , $m \geq 1$, $n \geq 1$.

Определим стоимость поиска по дереву L , среднее число проверяемых букв, требуемых для определения одного слова, равенством

$$C(L) = \frac{1}{m} \sum_{i=1}^m L_i,$$

где L_i — число проверяемых букв, требуемое для определения i -го слова, а m — число слов.

Обозначим через $\mathcal{S}_{n,m}$ множество начальных данных, то есть множество наборов из m разных двоичных слов длины n . Легко видеть, что $\mathcal{S}_{n,m}$ не пусто лишь при $n \geq \log_2 m$.

Предположим теперь, что некоторый алгоритм F по набору слов $S \in \mathcal{S}_{n,m}$ строит дерево $F(S)$. Так как мы будем рассматривать также и рандомизированные алгоритмы, то будет удобно обозначить через $\bar{C}(F(S))$ математическое ожидание стоимости дерева $C(F(S))$ по мере, заданной рассматриваемым алгоритмом. Наконец, теперь мы можем определить величину $t_{n,m}(F)$, относительно которой справедлива основная оценка, доказанная в работе. Мы положим ее равной средней стоимости получаемого алгоритмом F дерева в предположении, что начальные данные выбираются из множества $\mathcal{S}_{n,m}$, то есть

$$t_{n,m}(F) = \frac{1}{|\mathcal{S}_{n,m}|} \sum_{S \in \mathcal{S}_{n,m}} \bar{C}(F(S)),$$

где $|A|$ — число элементов множества A . Здесь и далее мы предполагаем, что на множестве $\mathcal{S}_{n,m}$ задано равномерное распределение.

В дальнейшем мы будем рассматривать, по-видимому, простейший алгоритм построения поискового дерева, который будем называть случайным и обозначать через R . Его работу можно описать следующим образом. Сначала корневой развилке сопоставляется проверка случайно выбранного двоичного разряда. Затем для каждой из частей, на которые эта проверка делит все множество слов, тем же способом строится свое определительное дерево.

Основным результатом данной работы является следующая теорема.

Теорема 1. *Для средней стоимости построенного алгоритмом R дерева в предположении равномерного распределения на множестве начальных данных $\mathcal{S}_{n,m}$ справедлива оценка*

$$t_{n,m}(R) \leq \log_2 m + \frac{29}{28} - \frac{\log_2(2m)}{m}. \quad (1)$$

Из этой теоремы очевидным образом выводится следующий результат.

Следствие 1. *Пусть F_0 — оптимальный алгоритм, строящий для каждого набора данных оптимальное дерево. Тогда для средней стоимости построенного этим алгоритмом дерева в предположении равномерного распределения на множестве начальных данных $\mathcal{S}_{n,m}$ справедлива оценка*

$$t_{n,m}(F_0) \leq \log_2 m + \frac{29}{28} - \frac{\log_2(2m)}{m}.$$

Следующее следствие позволяет оценить стоимость дерева, построенного алгоритмом R , и, следовательно, оптимального дерева не только в среднем, но и почти для каждого набора данных.

Следствие 2. *Предположим, что на множестве начальных данных $\mathcal{S}_{n,m}$, $m \geq 2$, $n \geq \log_2 m$, задано равномерное распределение. Тогда для любого $\varepsilon > 0$ справедливо неравенство*

$$\mathbf{P}\{S \mid \bar{C}(R(S)) < (1 + \varepsilon) \log_2 m\} \geq 1 - \frac{29}{28\varepsilon \log_2 m}.$$

Доказательство. Это следствие немедленно получается из неравенства Маркова.

3. Доказательство

Приведем схему доказательства. Сначала мы показываем, что средняя стоимость дерева, построенного случайным алгоритмом, удовлетворяет некоторому рекуррентному соотношению. Затем вычисляем некоторые значения $t_{n,m}(\cdot)$, которые будем использовать в дальнейших доказательствах. Далее приводим две технические леммы и, наконец, доказываем с их использованием основную теорему 1. В заключение мы приводим доказательство громоздкого технического утверждения.

Итак, приведем краткое доказательство леммы, на основании которой мы затем получим все остальные утверждения о случайном алгоритме.

Лемма 1. *Предположим, что на множестве $\mathcal{S}_{n,m}$ задано равномерное распределение. Тогда для средней стоимости дерева, построенного случайным алгоритмом R , выполнено соотношение*

$$t_{n,m}(R) = 1 + \sum_{k=0}^m c_{n,m,k} \left(\frac{k}{m} t_{n-1,k}(R) + \frac{m-k}{m} t_{n-1,m-k}(R) \right), \quad (2)$$

где $c_{n,m,k}$ — вероятность того, что в множестве начальных данных $\mathcal{S}_{n,m}$ встретится набор, в котором ровно k слов начинается с единицы, то есть

$$c_{n,m,k} = \begin{cases} \binom{2^{n-1}}{k} \binom{2^{n-1}}{m-k} / \binom{2^n}{m} & \text{при } m - 2^{n-1} \leq k \leq 2^{n-1}, \\ 0 & \text{в противном случае.} \end{cases} \quad (3)$$

Здесь и далее мы полагаем по определению $t_{n,0} \equiv 0$.

Доказательство. Сначала покажем, как получается формула (3). Если обозначить через $\mathcal{S}_{n,m}^k$ множество начальных данных, состоящее из наборов m слов длины n , среди которых ровно k начинается с единицы, то по определению,

$$c_{n,m,k} = |\mathcal{S}_{n,m}^k| / |\mathcal{S}_{n,m}|.$$

Очевидно, что $|\mathcal{S}_{n,m}| = \binom{2^n}{m}$. Легко видеть также, что множество $\mathcal{S}_{n,m}^k$ при $m - 2^{n-1} \leq k \leq 2^{n-1}$ имеет $\binom{2^{n-1}}{k} \binom{2^{n-1}}{m-k}$ элементов, а в остальных случаях пусто. Таким образом, справедлива формула (3).

Рассмотрим работу случайного алгоритма R на входе S . Пусть в первой случайно выбранной позиции k единиц и $m-k$ нулей. Тогда в этом случае стоимость алгоритма равна

$$1 + \frac{k}{m} \bar{C}(R(T_1 S)) + \frac{m-k}{m} \bar{C}(R(T_0 S)),$$

где $T_0 S$ и $T_1 S$ — поднаборы набора S , состоящие из слов, содержащих все биты, кроме бита в выбранной позиции, и содержащие в выбранной позиции нуль и единицу соответственно. Усреднив это равенство по множеству $\mathcal{S}_{n,m}$, получим, что

$$t_{n,m}(R) = 1 + \sum_{k=0}^m \frac{|\mathcal{S}_{n,m}^k|}{|\mathcal{S}_{n,m}|} \left(\frac{k}{m} \sum_{S \in \mathcal{S}_{n,m}^k} \frac{\bar{C}(R(T_1 S))}{|\mathcal{S}_{n,m}^k|} + \frac{m-k}{m} \sum_{S \in \mathcal{S}_{n,m}^{m-k}} \frac{\bar{C}(R(T_1 S))}{|\mathcal{S}_{n,m}^{m-k}|} \right).$$

Для завершения доказательства леммы осталось заметить, что

$$\sum_{S \in \mathcal{S}_{n,m}^k} \frac{\bar{C}(R(T_1 S))}{|\mathcal{S}_{n,m}^k|} = t_{n-1,k}(R).$$

Лемма доказана.

В дальнейшем нам потребуются некоторые значения средней стоимости $t_{n,m}(\cdot)$.

Лемма 2. *Явно вычисляются следующие значения средней стоимости деревьев: $t_{n,1} = 0$, $t_{n,2} = 1$ и $t_{n,3} = 5/3$ при $n \geq 2$.*

Доказательство. При данных значениях n и m все деревья поиска выглядят одинаково и, следовательно, у них совпадает средняя стоимость поиска вне зависимости от алгоритма, используемого для их построения.

Для вычисления других значений $t_{n,m}(\cdot)$ требуется учитывать специфику алгоритма. В следующей теореме мы оценим $t_{n,4}(R)$, используя лемму 1.

Теорема 2. *Предположим, что на множестве начальных данных $\mathcal{S}_{n,4}$ задано равномерное вероятностное распределение. Тогда для средней стоимости деревьев, получаемых алгоритмом R , при $n \geq 2$ выполнено неравенство*

$$t_{n,4}(R) \leq \frac{16}{7}. \quad (4)$$

Доказательство. Докажем индукцией по n , что $t_{n,4} \leq 16/7$ при $n \geq 2$. База индукции при $n = 2$ очевидна: $t_{2,4} = 2 < 16/7$. Осталось показать, что при увеличении n неравенство (4) будет сохраняться.

Воспользовавшись симметрией, перепишем формулу (2) из леммы 1 в виде

$$t_{n,m}(R) = 1 + \sum_{k=0}^m c_{n,m,k} \frac{2k}{m} t_{n-1,k}(R).$$

Представим по этой формуле $t_{n,4}(R)$, подставив значения $t_{n-1,0}(R) = 0$ (по определению), $t_{n-1,1}(R) = 1$, $t_{n-1,2}(R) = 1$, $t_{n-1,3}(R) = 5/3$ (по лемме 2), а также верхнюю оценку $t_{n-1,4}(R)$, справедливую по предположению индукции. Отсюда видно, что неравенство (4) вытекает из неравенства

$$1 + \frac{2}{2} c_{n,4,2} + \frac{3}{2} c_{n,4,3} \frac{5}{3} + 2 c_{n,4,4} \frac{16}{7} \leq \frac{16}{7},$$

которое в свою очередь легко следует из неравенства $-27 \cdot 2^{n-1} + 42 \leq 0$, верного при $n \geq 3$. Теорема доказана.

Приведем теперь две технические леммы, которые мы будем использовать при доказательстве теоремы 1. Первая из них является оценкой некоторой громоздкой функции с помощью полинома второй степени.

Лемма 3. Пусть $m \geq 5$. Для функции

$$f(k) = \frac{k}{m} \log_2(2k) \left(1 - \frac{1}{k}\right) + \frac{m-k}{m} \log_2(2(m-k)) \left(1 - \frac{1}{m-k}\right), \quad (5)$$

определенной на отрезке $[1, m-1]$, справедливо неравенство

$$f(k) \leq A - Bk(m-k),$$

где

$$A = \log_2(2m) \left(1 - \frac{1}{m}\right), \quad B = \frac{4}{m^3} (m + \log_2 m - 1).$$

Доказательство этой леммы приводится в конце статьи из-за его громоздкости.

Докажем комбинаторное тождество, позволяющее производить взвешенное суммирование симметричных полиномов второй степени.

Лемма 4. Справедливо тождество

$$\sum_{k=0}^m c_{n,m,k} k(m-k) = \frac{2^n}{2^n - 1} \frac{m(m-1)}{4}, \quad (6)$$

где $c_{n,m,k}$ определяются равенствами (3).

Доказательство. Для доказательства данного утверждения надо приравнять коэффициенты при x^{m-2} в тождестве

$$(x-1)^{2^{n-1}-1} (x-1)^{2^{n-1}-1} = (x-1)^{2^n-2},$$

откуда путем несложных преобразований получается требуемое тождество. Лемма доказана.

Доказательство теоремы 1. Доказательство проведем индукцией по n . Взяв $t_{n,m}(\cdot)$ из леммы 2, прямым вычислением проверяем, что (1) справедливо при $m = 1, 2, 3$. Аналогично проверим неравенство (1) при $m = 4$, зная что по теореме 2, $t_{n,m}(R) \leq 16/7$ при $n \geq 2$.

Предыдущими рассуждениями нами доказана база индукции, так как при $n = 1, 2$ число определяемых объектов не превосходит четырех.

Проведем шаг индукции. Предположим, что при всех k таких, что $1 \leq k \leq \log_2(n-1)$, для $t_{n-1,k}(R)$ выполнено неравенство (1). Для доказательства шага индукции нам необходимо показать, что неравенство (1) выполнено для $t_{n,m}(R)$ при всех m , для которых $5 \leq m \leq \log_2 n$.

Докажем, что неравенство

$$\frac{k}{m} t_{n-1,k}(R) + \frac{m-k}{m} t_{n-1,m-k}(R) \leq A_1 - Bk(m-k), \quad (7)$$

где

$$A_1 = \frac{1}{28} + \log_2(2m) \left(1 - \frac{1}{m}\right), \quad B = \frac{4}{m^3} (m + \log_2 m - 1),$$

справедливо при всех k от 0 до m . Заметим, что $t_{n-1,0}(R) = 0$, $t_{n-1,m}(R) \leq A_1$. Таким образом, при $k = 0$ и при $k = m$ это неравенство доказывается очевидной проверкой.

При k от 1 до $m - 1$ неравенство (7) устанавливается заменой всех входящих в него $t_{n-1,k}(R)$ верхней оценкой (1) и последующим применением леммы 3.

Теперь мы можем заключить, что согласно уравнению (2) из леммы 1, для $t_{n,m}(R)$ справедлива оценка

$$t_{n,m}(R) \leq 1 + \sum_{k=0}^m c_{n,m,k} (A_1 - Bk(m-k)) = 1 + A_1 - B \frac{2^n}{2^n - 1} \frac{m(m-1)}{4},$$

где равенство устанавливается на основании леммы 4. Теперь для завершения шага индукции осталось показать, что

$$1 + A_1 - B \frac{2^n}{2^n - 1} \cdot \frac{m(m-1)}{4} \leq \log_2 m + \frac{1}{28} - \frac{\log_2(2m)}{m}. \quad (8)$$

Для этого перемножим очевидные неравенства

$$1 < 2^n / (2^n - 1)$$

и

$$m^2 < (m-1)(m-1 + \log_2 m)$$

при $m \geq 5$, после чего путем несложных преобразований получим требуемое утверждение (8). Теорема доказана.

Осталось привести доказательство леммы 3. Схема доказательства данной леммы аналогична доказательству известной в теории информации верхней оценке энтропии $H(p) \leq 4p(1-p)$ (см., например, [2]).

Доказательство леммы 3. Обозначим через $g(k)$ разность $A - Bk(m-k) - f(k)$. Для наглядности доказательства график этой функции при $m = 100$ приведен на рис. 2. Покажем, что при $m \geq 5$ эта функция неотрицательна на отрезке $[1, m-1]$. Для этого отметим следующие свойства функции g , доказательства которых приведем сразу после их формулировок.

Функция $g(k)$ симметрична относительно середины отрезка $[1, m-1]$, то есть $g(k) = g(m-k)$. Доказательство очевидно.

В середине отрезка $[1, m-1]$ функция $g(k) = 0$ обращается в нуль. Действительно,

$$\begin{aligned} g(m/2) &= A - \frac{4}{m^3} (m + \log_2 m - 1) \left(\frac{m}{2}\right)^2 - 2 \frac{m/2}{m} \log_2 m \left(1 - \frac{1}{m/2}\right) \\ &= A - \frac{m-1}{m} - \frac{1}{m} \log_2 m - \log_2 m \left(1 - \frac{2}{m}\right) \\ &= A - \left(1 - \frac{1}{m}\right) - \log_2 m \left(1 - \frac{1}{m}\right) = 0. \end{aligned}$$

В концах отрезка $[1, m-1]$ функция $g(k)$ положительна.

Известно, что справедлива оценка $\log_2 e > 7/5$. Но тогда при $m \geq 5$ выполнено неравенство $m(\log_2 e - 1) > 2$. Следовательно, справедливо неравенство

$$-2m + 2 + m(\log_2 e - 1) > -2m + 4, \text{ откуда } 2(m-2) > 3m - 2 - m \log_2 e.$$

Рис. 2. График $g(k)$ при $m = 100$

Учитывая, что $\log_2 m > 2$, получим, что $(m-2)\log_2 m > 3m-2-m\log_2 e$, откуда путем тождественных преобразований получается неравенство

$$m(m-2)\log_2 e + (m-2)^2\log_2 m + m^2 > 4(m-1)^2. \quad (9)$$

Теперь воспользуемся тем, что $\log_2 x / \log_2 e \leq x-1$ при $x \in (0, \infty)$. Полагая $x = 1 - 1/m$ находим, что

$$-\log_2(1 - 1/m) \geq \log_2 e/m.$$

Воспользовавшись этим неравенством, преобразуем (9) следующим образом:

$$\begin{aligned} -m^2(m-2)\log_2\left(1 - \frac{1}{m}\right) + (m-2)^2\log_2 m + m^2 &> 4(m-1)^2, \\ -f(1) + \frac{m-2}{m}\log_2(2m) + \frac{1}{m}\log_2 m + \frac{1}{m} &> (m-1)B, \end{aligned}$$

откуда следует, что

$$g(m-1) = g(1) = A - B(m-1) - f(1) > 0$$

при $m \geq 5$, что и требовалось доказать.

Вторая производная функции $g(k)$ строго неограниченно убывает от середины интервала $(0, m)$ к его краям. Для доказательства вычислим вторую производную функции $g(k)$:

$$g''(k) = 2B - \frac{\log_2 e}{m} \left(\frac{1}{k} + \frac{1}{k^2} + \frac{1}{m-k} + \frac{1}{(m-k)^2} \right).$$

Обозначим через d выражение $(k/m - 1/2)^2$. Тогда выражение в скобках, пропорциональное второй производной функции $f(k)$, можно переписать в виде

$$\frac{1}{k} + \frac{1}{k^2} + \frac{1}{m-k} + \frac{1}{(m-k)^2} = \frac{1}{m(1/4-d)} + \frac{d+1/2}{m^2(1/4-d)^2}.$$

Легко видеть, что оно строго неограниченно возрастает при изменении d от 0 до $1/4$, то есть при удалении точки k от центра отрезка. Но тогда $g''(k)$ при этом строго убывает.

В середине отрезка $[1, m-1]$ вторая производная $g''(m/2)$ положительна. Действительно,

$$g''(m/2) = 2B - \frac{\log_2 e}{m} \left(\frac{4}{m} + \frac{8}{m^2} \right) = \frac{4}{m^3} (2(m-1 + \log_2 m) - \log_2 e (1+m)) > 0,$$

так как при $m \geq 4$ выполнена цепочка неравенств

$$2(m-1 + \log_2 m) \geq 2(m-1+2) \geq \log_2 e (1+m).$$

Теперь нам осталось показать, что минимум функции $g(k)$ на отрезке $[1, m-1]$ достигается в одной из трех точек множества $\{1, m/2, m-1\}$. Так как в этих точках $g(k)$ неотрицательна, на отрезке $[1, m-1]$ она также неотрицательна, что и требуется доказать.

Для доказательства заметим, что так как вторая производная $g''(k)$ положительна в середине интервала $(0, m)$ и неограниченно убывает к его краям, найдется δ такое, что при $k = m/2 \pm \delta$, вторая производная функции $g(k)$ обратится в ноль. Более того, она положительна на интервале $(m/2 - \delta, m/2 + \delta)$ и отрицательна вне его замыкания.

Но тогда функция $g(k)$ выпукла вверх на отрезке $[m/2 - \delta, m/2 + \delta]$. Так как она симметрична относительно середины отрезка, минимум $g(k)$ достигается в этой середине. Если $m/2 - \delta \leq 1$, то все доказано, иначе $g(k)$ выпукла вниз на отрезках $[1, m/2 - \delta]$ и $[m/2 + \delta, m-1]$, и, следовательно, на этих отрезках достигает минимума в одном из концов. Итак, нами показано, что минимум функции $g(k)$ на отрезке $[1, m-1]$ достигается в одной из трех точек множества $\{1, m/2, m-1\}$, что завершает доказательство.

Авторы благодарят М. А. Алексеева за полезные обсуждения, а также рецензента за конструктивные замечания.

Список литературы

1. Кнут Д., *Искусство программирования на ЭВМ*, т. 3. Мир, Москва, 1976.
2. Альсведе Р., Вегенер И., *Задачи поиска*. Мир, Москва, 1982.
3. Кричевский Р. Е., *Сжатие и поиск информации*. Радио и связь, Москва, 1989.
4. Гэри М., Джонсон Д., *Вычислительные машины и труднорешаемые задачи*. Мир, Москва, 1982.

Статья поступила 21.09.1999.

Переработанный вариант поступил 31.08.2000.