Семинар 8.
Вероятностные классы

Составил Р. Делла Пиетра

20.4.20

1 Определения

1.1 Вероятностная машина Тьюринга

Первое определение: есть ячейка, в которой появляется случайный бит, равновероятно 0 или 1. МТ может считывать этот бит и как-либо менять своё поведение в зависимости от этого бита, и после каждого чтения бит меняется на новый случайный.
Второе определение: есть дополнительная лента со случайными битами, которые МТ может считывать только слева направо.
Третье определение: есть две функции переходов δ_1 и δ_2, и в каждый момент времени равновероятно выбирается одна из них.
Эти три определения эквивалентны, и далее под ВМТ будет иметься в виду такая МТ.

1.2 «Базовые» классы

Аналогично $DTIME$, вводится такие классы:

$$BPTIME(t(n)) = \{L | \exists \text{ ВМТ, «разрешающая» } L \text{ с двусторонней ошибкой за } O(t(n)) \text{ тактов}\}$$

$$RTIME(t(n)) = \{L | \exists \text{ ВМТ, «разрешающая» } L \text{ с односторонней ошибкой за } O(t(n)) \text{ тактов}\}$$

$$ZPTIME(t(n)) = \{L | \exists \text{ ВМТ, «разрешающая» } L \text{ без ошибок за } O(t(n)) \text{ тактов в среднем}\}$$

Под разрешением с двусторонней ошибкой имеется в виду это:

$$x \in L \implies \mathbb{P}\{M(x) = 1\} \geqslant \frac{2}{3}$$

$$x \notin L \implies \mathbb{P}\{M(x) = 0\} \geqslant \frac{2}{3}$$

Под разрешением с односторонней ошибкой имеется в виду это:

$$x \in L \implies \mathbb{P}\{M(x) = 1\} \geqslant \frac{1}{2}$$

$$x \notin L \implies \mathbb{P}\{M(x) = 0\} = 1$$

$O(t(n))$ тактов в среднем значит, что матожидание количества тактов равно $O(t(n))$

1.3 Классические вероятностные классы

$$\mathcal{BPP} = \bigcup_{c=1}^{\infty} BPTIME(n^c)$$

$$\mathcal{RP} = \bigcup_{c=1}^{\infty} RTIME(n^c)$$

$$\mathcal{ZPP} = \bigcup_{c=1}^{\infty} ZTIME(n^c)$$

2 Лемма об уменьшении ошибки

Теорема Чернова: если есть некоторое множество независимых одинаково распределённых случайных величин (i.i.d.) с конечным матожиданием μ, выполняется такое неравенство:

$$\mathbb{P}\left\{\left|\frac{\sum x_i}{n} - \mu\right| \geqslant \varepsilon\right\} \leqslant 2e^{-\varepsilon^2 n}$$

Если взять x_i = результат i-го запуска ВМТ на входе x, теорема превращается в оценку на вероятность ошибки, и можно оценить, сколько запусков надо сделать, что достичь любой наперёд заданный допуск.

3 Иерархия

Несколько тривиальных утверждений: $\mathcal{P} \subseteq \mathcal{RP}$, потому что полиномиально разрешимые языки разрешаются без ошибок с обоих сторон, $\mathcal{RP} \subseteq \mathcal{BPP}$, что легко доказывается с помощью леммы об уменьшении вероятности ошибки.
Открытый вопрос — как соотносятся $\mathcal{P}$, $\mathcal{NP}$ и $\mathcal{BPP}$. Обычно предполагается, что $\mathcal{BPP} = \mathcal{NP}$, но это всё же открытый вопрос, и стоит относиться так же, как к равенству $\mathcal{NP}$ и **EXP**.

3.1 $\mathcal{BPP} \subseteq \mathcal{P}/poly$

Воспользуемся определением $\mathcal{P}/poly$ через подсказку полиномиальной длины: $\exists h(n)$ полиномиальной длины, с помощью которой можно за полиномиальное время разрешить все $y : |x| = n$. $h(x)$ не обязана быть вычислимой.
Покажем, что такая подсказка существует, с помощью оценок: пусть для всех входов длины n ВМТ использует не более $k(n)$ случайных битов. При этом она работает за полиномиальное время, поэтому, очевидно, $k(n)$ тоже полином.
С помощью леммы об уменьшении ошибки добьёмся такой точности: $\mathbb{P}\{M(x) \neq \chi_L(x)\} \leqslant \frac{1}{2^{n^2}}$.
Всего возможных строчек $2^{k(n)}$, из них «плохих» для некоторого фиксированного $x_0 : |x_0| = n$ будет $2^{k(n)-n^2}$. Всего таких слов 2^n, поэтому всего плохих строк $2^{k(n)+n-n^2} < 2^{k(n)}$. Это значит, что среди $2^{k(n)}$ случайных строк есть строки, с которыми ВМТ на всех входах не ошибается. Один из таких входов и будет подсказкой, а разрешающая МТ будет моделировать ВМТ с предвыбранными случайными битами, что можно сделать детерминированно.

3.2 $\mathcal{BPP} \subseteq \mathcal{P}/poly$

Воспользуемся определением $\mathcal{P}/poly$ через подсказку полиномиальной длины: $\exists h(n)$ полиномиальной длины, с помощью которой можно за полиномиальное время разрешить все $y : |x| = n$. $h(x)$ не обязана быть вычислимой.
Покажем, что такая подсказка существует, с помощью оценок: пусть для всех входов длины n ВМТ использует не более $k(n)$ случайных битов. При этом она работает за полиномиальное время, поэтому, очевидно, $k(n)$ тоже полином.
С помощью леммы об уменьшении ошибки добьёмся такой точности: $\mathbb{P}\{M(x) \neq \chi_L(x)\} \leqslant \frac{1}{2^{n^2}}$.
Всего возможных строчек $2^{k(n)}$, из них «плохих» для некоторого фиксированного $x_0 : |x_0| = n$ будет $2^{k(n)-n^2}$. Всего таких слов 2^n, поэтому всего плохих строк $2^{k(n)+n-n^2} < 2^{k(n)}$. Это значит, что среди $2^{k(n)}$ случайных строк есть строки, с которыми ВМТ на всех входах не ошибается. Один из таких входов и будет подсказкой, а разрешающая МТ будет моделировать ВМТ с предвыбранными случайными битами, что можно сделать детерминированно.

4 Задачи

4.1 Задача сравнения двух чисел (файлов)

Пусть есть два больших числа (или файла) X и Y, считаем что в каждом из них $n-1$ бит. Сколько бит достаточно сравнить, чтобы с вероятностью не менее $\frac{3}{4}$ сказать, что числа равны?
Оказывается, при больших n логарифма бит достаточно.
Случайно выберем некоторое простое число p, лежащее на отрезке $[n, 2n]$. Оно точно найдётся по постулату Бертрана. Далее будем сравнивать остатки от деления X и Y на p (U и V) соответственно.
$|p| \approx \log n \implies |U|, |V| \approx \log n$. Найдём вероятность ошибки, то есть $\mathbb{P}\{U = V, X \neq Y\} = \mathbb{P}\{X \neq Y, X \equiv_p Y\}$.
У числа $|X-Y|$ существует как минимум один делитель на отрезке $[n, 2n]$ (число p). Обозначим за $p_1, \dots, p_m$ все делители из этого отрезка.

$$2^n \geqslant |X - Y| \geqslant p_1 p_2 \dots p_m \geqslant n^m \implies m \leqslant c\frac{n}{\ln n}$$

Зная, что количество простых чисел в натуральном ряду растёт как $\frac{n}{\ln n}$ для достаточно больших n, осталось оценить вероятность неблагоприятного исхода.

4.2 Выполнение $\geqslant \frac{7}{8}$ дизъюнктов в РОВНО3КНФ

Пусть есть некоторая формула в РОВНО3КНФ φ. Также добавим ограничение: в каждом дизъюнкте литералы должны быть разные.
Введём случайные величины $\xi(\mathbf{A})$ = количество выполненных дизъюнктов на наборе $\mathbf{A}$, $\xi_i(\mathbf{A})$ = выполнен ли дизъюнкт i на таком наборе. Очевидно, $\xi = \sum\limits^k \xi_i$.

$\mathbb{E}\xi_i = \frac{7}{8}$, т.к. для трёх разных переменных есть только один набор из восьми, для которого дизъюнкт не выполняется. Либо в дизъюнкте есть переменная и её отрицание, тогда матожидание просто 1.
$\mathbb{E}\xi = \sum \xi_i \geqslant \frac{7}{8}k$. Это значит, что точно существует набор, удовлетворяющий не менее $\frac{7}{8}$ дизъюнктов. Если это не так, матожидание не может быть больше $\frac{7}{8}$: $\mathbb{E}\xi = \sum\limits_y y\mathbb{P}\{\xi = y\} < \frac{7}{8}\sum\limits_y \mathbb{P}\{\xi = y\} = \frac{7}{8}$.

Как найти этот набор? Сделаем подобие двоичного поиска. Начинаем с первой переменной:

$$\mathbb{E}\xi = \mathbb{E}\{\xi|x_1 = 0\}\mathbb{P}\{x_1 = 0\} + \mathbb{E}\{\xi|x_1 = 1\}\mathbb{P}\{x_1 = 1\} = \frac{\mathbb{E}\{\xi|x_1 = 0\} + \mathbb{E}\{\xi|x_1 = 1\}}{2}$$

Опять же, одно из слагаемых должно быть не меньше, чем $\frac{7}{8} \implies$ выбираем x_1 по этому слагаемому. В итоге мы выберем весь набор, и матожидание ξ при фиксированном наборе, то есть просто количество выполненных дизъюнктов, будет не меньше $\frac{7}{8}$.

4.3 Вероятностный РОВНО2КНФ

Построим итерационный алгоритм: начинаем с набора из нулей, на каждом шаге если формула не выполнена, выбираем невыполненный дизъюнкт, выбираем в нём любую переменную и меняем её значение.
Теперь проанализируем это как вероятностный алгоритм. Пусть формула выполнима, то есть есть некоторый набор $\mathbf{S}$, на котором она выполняется. $\mathbf{A}_i$ — текущий набор, $\mathbf{A}_0 = \mathbf{0}$. ξ_i = количество дизъюнктов, выполненных на шаге i, то есть на наборе $\mathbf{A}_i$.
Рассмотрим, как меняется ξ_i:

$\xi_i = 0$		$\xi_{i+1} = 1$
$\xi_i > 0$	в выбранном дизъюнкте обе переменные не совпадают по значению с $\mathbf{S}$	$\xi_{i+1} = \xi_i + 1$
$\xi_i > 0$	в выбранном дизъюнкте одна совпадает с $\mathbf{S}$, другая нет	$\xi_{i+1} = \xi_i + 1$ или $\xi_{i+1} = \xi_i - 1$ равновероятно

Из этой таблицы видно, что $\mathbb{P}\{\xi_{i+1} = \xi_i + 1\} \geqslant \frac{1}{2}$, $\mathbb{P}\{\xi_{i+1} = \xi_i - 1\} \leqslant \frac{1}{2}$. Рассматриваем худший случай, когда вероятности равны $\frac{1}{2}$.
Введём $T(i)$ — матожидание количества шагов от $\xi_k = i$ до $\xi_m = n$.
Раскладывая это матожидание по формуле полной вероятности, получаем такую систему:

$$\begin{cases} T(n) &= 0 \\ T(0) &= T(1) + 1 \\ T(i) &= \dfrac{T(i+1) + T(i-1)}{2} + 1 \end{cases} \implies T(i) = n^2 - i^2$$

То есть в среднем нужно сделать n^2 шагов , чтобы таким случайным блужданием получить выполняющий набор. С помощью неравенства Маркова можно оценить вероятность того, что, если выполняющий набор есть, он не найден:

$$\mathbb{P}\{\exists \mathbf{S},\ \text{сделано } \geqslant 2n^2 \text{шагов}\} \leqslant \frac{T(0)}{2n^2} = \frac{1}{2}$$

Таким образом, если сделать $100n^2$ шагов, вероятность не найти набор будет $\frac{1}{2^{50}}$. Отсюда можно найти нужную нам точность. Если с нужной точностью набор не найден, можно говорить, что его не существует.