Конспект по теории информации
IV семестр, 2021 год
Современное программирование, факультет математики и компьютерных наук, СПбГУ
(лекции Дмитрия Соколова)

Тамарин Вячеслав

April 29, 2021

Contents

Исходный код на https://github.com/tamarinvs19/theory_university

Index

Контакты: sokolov.dmt@gmail.com,

Видимо будет письменный экзамен.

Есть прошлогодний конспект, там есть существенные ошибки, плюс курс немного отличается.

Chapter 1

Введение

1.1 Информация по Хартли

Пусть у нас есть конечное множество объектов A. Выдернем какой-то элемент.

Мы хотим придумать описание этого элемента, которое будет отличать его от всех остальных.

Самый простой вариант — число битов требуемое для записи объекта.

Свойства, которые мы хотим получить от меры $\chi(A)$:

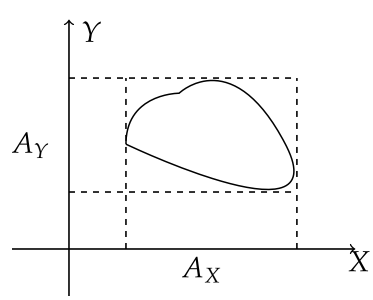
1. χ дает нам оценку на длину описаний
2. $\chi(A \cap B) \leqslant \chi(A) + \chi(B)$
3. Если наше множество $A \coloneqq B \times C$, то можно описать для B и для C, поэтому можно ограничить:
$$\chi(A) \leqslant \chi(B) + \chi(C).$$

Определение 1 (Информация по Хартли). $\chi(A) \coloneqq \log|A|$

Замечание. Очевидно, второе свойство выполнено для такого определения. В третьем даже равенство.

Описание — например, битовая строка. Если логарифм нецелый, округляем вверх.

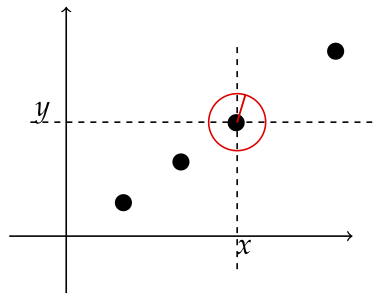
Пусть $A \subset X \times Y$. Обозначим проекции A_X и A_Y. Здесь



1. $\chi(A) \geqslant 0$
2. $\chi(A_X) \leqslant \chi(A)$
3. $\chi(A) \leqslant \chi(A_X) + \chi(A_Y)$

Рассмотрим такой пример: здесь, зная первую координату, можно сразу понять вторую.

Попробуем усилить третье свойство:

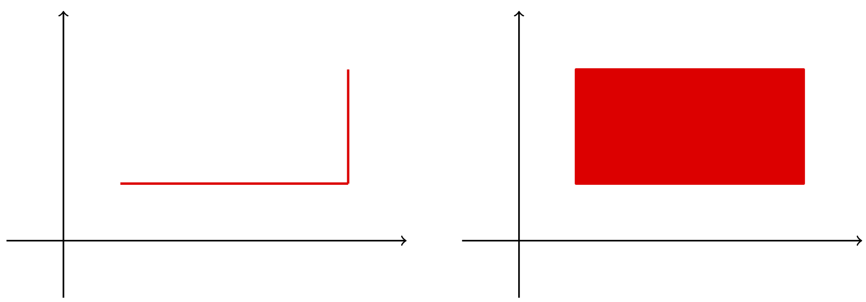


3'. $\chi(A) \leq \chi(A_X) + \chi_{Y|X}(A)$, где $\chi_{Y|X}(A)$ — описание Y при условии X.

Как будем определять $\chi_{Y|X}(A)$? Можно взять $\max_{x\in X} \log(A(x))$.

Теперь для диагонального множество $\chi_{Y|X}$ просто обнуляется и неравенство переходит в равенство.

Но если взять такие множества. Во-первых, на первой картинке передав x столбца придется передавать



и y тоже. Во-вторых, мы не сможем отличить эти множества.

Упражнение. Пусть $A \subset X \times Y \times Z$. Доказать

$$2\chi(A) \leq \chi(A_{XY}) + \chi(A_{XZ}) + \chi(A_{YZ}).$$

1.1.1 Применение информации

Обозначим $[n] \coloneqq \{1, \ldots, n\}$. Первый игрок выбирает одно число, а второй должен угадывать.

Если два варианта игры:

- Адаптивная — ответ сразу
- Сначала пишем все запросы, а потом получаем все ответы.

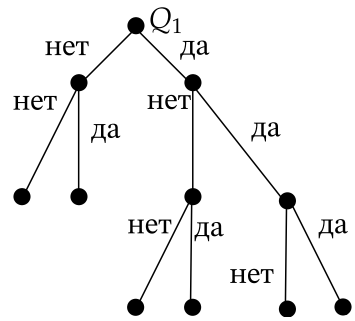
Очевидно, что нам потребуется не менее логарифма запросов: нарисуем дерево, где вершины – запросы, по двум ребрам можно перейти в зависимости от ответа. Листья должны содержать $[n]$, поэтому глубина дерева не менее логарифма.

Теперь подумаем с точки зрения теории информации. Пусть $B \coloneqq Q_1 \times \ldots \times Q_h$, h — число запросов, Q_i — ответ на запрос по некоторому протоколу. Хотим минимизировать h.

Рассмотрим $([n], B)$ — все возможные пары —. Нас интересует множество $A \subseteq ([n], B)$ — соответствует некоторым корректным запросам, здесь записаны ответы нашего протокола.

$$A = \{(m, b) \mid b = (q_1, \ldots, q_h),\ m \text{ — согласовано с ответом}\}.$$

1. $\chi_{[n]|B}(A) = 0$. Ответы на запросы должны однозначно определять число m. Это свойство говорит о корректности протокола, то есть нам ничего не нужно, чтобы, зная ответы, получить m.



2. $\log n \leqslant \chi(A)$, так как хотя бы столько мы запихнули. С другой стороны, $\chi(A) \leqslant \chi_B(A) + \chi_{[n]|B}(A) = 0$, а $\chi_B(A) \leqslant \chi(B) \leqslant \sum_{i=1}^h \chi(Q_i) = h$. Итого
$$\log n \leqslant h.$$

Другая формулировка

Пусть теперь за ответ «да» мы платим 1, а за «нет» 2. И мы хотим минимизировать не число запросов, а стоимость в худшем случае.
$$Q_i \overset{?}{\in} T_i.$$
Пусть A_i — множество возможных x (ответов) перед шагом i. В начале это все $[n]$, в конце – одно число.
$$A_i = \{a \in [n] \mid a \text{ согласовано с } Q_1, \dots Q_{i-1}\}.$$
Стратегия минимальной цены бита информации: берем такое T_i, что
$$2\big(\chi(A_i) - \chi(\underbrace{T_i}_{A_{i+1}})\big) = \chi(A_i) - \chi(\underbrace{A \setminus T_i}_{A_{i+1}}).$$
Докажем, что эта стратегия оптимальна. То есть для любой другой стратегии найдется число, с которым мы заплатим больше.

Если заплатили 1, то перешли в $A_i \to T_i$. Если заплатили 2, то $A_i \to A_i \setminus T_i$. Заметим, что каждый раз мы заплатили за каждый бит одинаково.

Докажем оптимальность. Пусть второй игрок меняет число, чтобы мы заплатили как можно больше, причем он знает нашу стратегию.

Если в нашем неравенстве знак $\geqslant$, он будет направлять на по «нет», а при $\leqslant$ «да», за счет чего каждый бит он будет отдавать по цене большей, чем, если бы мы действовали в точности по стратегии.

Следовательно, любая другая стратегия будет требовать большего вклада.

Можем решить уравнение на T_i, должно получиться:
$$\Phi(|T_i|) = |A_i|, \quad \Phi \text{ — золотое сечение.}$$
Упражнение (Задача про взвашивания монеток). Есть n монеток и рычажные весы. Хотим найти фальшивую (она одна).

1. Пусть $n = 30$ и весы показывают, что больше, что меньше. Теперь запрос приносит $\log 3$ информации, так как три ответа.
$$\log 30 \leqslant \sum_{i=1}^h \chi(a_i) \leqslant h \log 3.$$

2. $n = 15$, но мы не знаем относительный вес фальшивой монеты. В прошлом неравенстве можно заменить 30 на 29. Если в какой-то момент у нас было неравенство, можем в конце узнать не только номер, но и относительный вес, поэтому у нас 29 исходов.

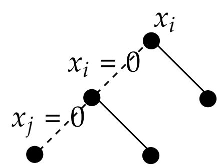
3. Вопрос: можно ли при $n = 14$? Нет.

1.1.2 Жизненные применения

Мы хотим решать задачу выполнимости.

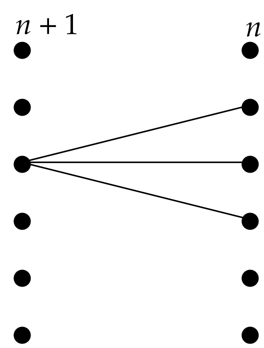
Вход: $\Phi = \wedge C_i$ — формула в КНФ.

Подставим $x_i = 0$. Если один из слозов нарушился, вернемся на шаг назад и подставим $x_j = 1$, а иначе подставляем дальше.



Это достаточно эффективный алгоритм, причем мы не ограничиваем выбор последовательности подстановок, порядок 0 и 1.

Рассадка голубей



Вопросы — сажаем ли мы голубя в клетку i?

Пусть один игрок загадал расстановку голубей, а второй хочет найти дизъюнкт, для которого нарушается эта расстановка.

1.2 Новая мера информации

На прошлой лекции поняли, что не всегда можем отличить некоторые множества.

Попробуем исправить данную ситуацию. Хотим понять состояния в Y, зная информацию об X. В среднем нам нужно сильно меньше информации, чем в крайнем случае.

Введем новую меру информации $\mu(\alpha)$, где α — распределение (множество и вероятности каждого элемента). Причем хотим, чтобы основные свойства были согласованы: [1]

1. $\mu(U_n) = \log n$;

2. $\mu(\alpha) \geqslant 0$;

3. $\mu(\alpha, \beta) = \mu(\alpha) + \mu(\beta)$, если α и β независимы.

Если действовать как настоящие математики, можно переписать эти свойства в более общие:

[1] $\mu(x, y) = \mu((x, y))$

1. $\mu(U_M) \geqslant \mu(U_{M'})$, если $|M| \geqslant |M'|$;
2. $\mu(\alpha, \beta) = \mu(\alpha) + \mu(\beta)$, если α и β независимы;
3. $\mu(B_p)$ непрерывно по $p \in [0, 1]$, где B_p — распределение для монетки, вероятность орла – p.
4. $\mu(B_p, \alpha) = \mu(B_p) + Pr[B_p = 0] \cdot \mu(\alpha \mid B_p = 0) + Pr[B_p = 1] \cdot \mu(\alpha \mid B_p = 1)$.

Определение 2 (Энтропия)**.** Этим аксиомам удовлетворяет примерно одна функция $\mu(\alpha) := k \cdot H(\alpha)$, где $H(\alpha)$ — **энтропия**.[a]

$$H(\alpha) = \sum_{i=1}^{|\mathrm{supp}(\alpha)|} p_i \log \frac{1}{p_i}.$$

Энтропия обозначает среднее по распределению α необходимое количество информации для записи элемента.

[a] $\mathrm{supp}\,\alpha$ — все возможные события, то есть имеющие ненулевую вероятность

Замечание. Энтропия равномерного распределения равна $\log n$, если $p_i = n$.

Замечание. Далее $H(p)$ обозначает энтропию для распределения монетки.

Теорема 1.2.1. $H(\alpha) \leqslant \log|\mathrm{supp}(\alpha)|$

□ Применим неравенство Йенсена

$$\sum_{i=1}^{|\mathrm{supp}(\alpha)|} p_i \log \frac{1}{p_i} \leqslant \log\left(\sum_i p_i \frac{1}{p_i}\right) = |\mathrm{supp}(\alpha)|$$

■

Теорема 1.2.2. $H(\alpha, \beta) \leqslant H(\alpha) + H(\beta)$

□

$$H(\alpha, \beta) = \sum_{i,j} p_{i,j} \log \frac{1}{p_{i,j}}$$

$$H(\alpha) + H(\beta) = \sum_i p_i \log \frac{1}{p_i} + \sum_j p_j \log \frac{1}{p_j}$$

Заметим, что $p_i = \sum_j p_{i,j}$ и $p_j = \sum_i p_{i,j}$.

$$H(\alpha, \beta) - H(\alpha) - H(\beta) = \sum_{i,j} p_{i,j} \log \frac{1}{p_{i,j}} - \sum_i p_i \log \frac{1}{p_i} + \sum_j p_j \log \frac{1}{p_j} = \sum_{i,j} p_{i,j} \log \frac{p_i p_j}{p_{i,j}}.$$

Если α и β независимы, то все логарифмы обнуляются. Иначе по неравенству Йенсена

$$\sum_{i,j} p_{i,j} \log \frac{p_i p_j}{p_{i,j}} \leqslant \log\left(\sum_{i,j} p_i p_j\right) = 0.$$

■

Определение 3 (Условная энтропия)**.**

$$H(\alpha \mid \beta = b) = \sum_i Pr[\alpha = i \mid \beta = b] \cdot \log \frac{1}{Pr[\alpha = i \mid \beta = b]}.$$

$$H(\alpha \mid \beta) = \mathbb{E}_{b=\beta} H(\alpha \mid p = b) = \sum_b H(\alpha \mid \beta = b) Pr[beta = b].$$

Свойства.

1. $\forall f\colon\ H(\alpha \mid \beta) \geqslant H(f(\alpha) \mid \beta)$
2. $H(\alpha, \beta) = H(\alpha) + H(\beta \mid \alpha)$
3. $H(\alpha) \geqslant H(\alpha \mid \beta)$
   □
   $$H(\alpha \mid \beta) - H(\alpha) = \sum p_{i,j} \frac{1}{\log Pr[\alpha = i \mid \beta = j]} - \sum p_{i,j} \log \frac{1}{p_i} \leqslant \sum p_{i,j} \log \frac{p_i}{Pr[\alpha = i \mid \beta = j]}.$$
   По неравенству Йенсена полученное выражение меньше нуля. ■
4. $H(\alpha \mid \beta) \geqslant H(\alpha \mid \beta, \gamma)$

Попробуем решить задачу с монетками. Мы взвешиваем 14 монеток и хотим найти фальшивую за три взвешивания, причем неизвестен относительный вес. В нашем графе есть только один исход со всеми равенствами. Докажем, что нет такой стратегии.

Пусть нам дали текущее состояние и стратегия Сделаем так, чтобы каждый лист был равновероятен. Вернем с вероятностью $\frac{1}{27}$, что i фальшивая, и с $\frac{1}{27}$ – больше ($l > i$ фальшивая), и также с $l < i - \frac{1}{27}$.

При равномерном распределении энтропия $3 \log 3$.

Если стратегия верная, то

$$\begin{aligned} \log 27 &\leqslant H(\alpha, q_1, q_2, q_3) \leqslant \\ &\leqslant H(q_1) + H(q_2 \mid q_1) + H(q_3 \mid q_1, q_2) + H(\alpha \mid q_1, q_2, q_3) \leqslant && \text{(Cain rule)} \\ &\leqslant H(q_1) + H(q_2) + H(q_3) + 0 \end{aligned}$$

Так как $H(q_i) \leqslant \log 3$, для все i выполнено равенство.

Чтобы было так, мы должны в каждый ход равновероятно получать все три ответа. Пусть мы взвешиваем кучки из k монет[2]. Вероятность равенства должна быть $\frac{2k}{27}\frac{1}{3}$, то есть $k \notin \mathbb{N}$. Противоречие.

1.3 Биномиальное распределение

$$\sum_{i=0}^{k} \binom{n}{i} \leqslant 2^{nH(\frac{k}{n})}.$$

Обозначим сумму за C.

Будем выбирать множество размера не больше k, а затем проверять, попало ли i наше множество. Пусть X — индикатор того, что i выбрали.

$$\begin{aligned} \log C = H(X) &\leqslant H(X_1, \ldots, X_n) \leqslant \\ &\leqslant \sum H(X_i \mid X_{<i}) \leqslant && \text{(Chain rule)} \\ &\leqslant \sum H(X_i) = nH(X_1) \leqslant && (\text{считаем, что } k \leqslant \tfrac{n}{2}) \\ &\leqslant nH\left(\tfrac{k}{n}\right) \end{aligned}$$

Лекция 3: †

15 April

1.4 Подсчет углов в графе

Рассмотрим ориентированный граф.

Назовем треугольником тройку (x, y, z), если это цикл из трех вершин. Углом назовем тройку (x, y, z), если есть ребра xy и xz, при этом y может совпадать z.

Чего в графе больше: углов или треугольников?

Замечание. Каждое ребро тоже угол, например, (x, y, y).

Теорема 1.4.1. Число углов в графе всегда больше числа треугольников.

☐ Пусть случная величинаα равна случайному треугольнику.

Так как распределение количества треугольников равномерно,

$$\begin{aligned}
\log(\#\Delta) &= H(x,y,z) = && \text{(Chain rule)}\\
&= H(x) + H(y \mid x) + H(z \mid y, x) \leqslant \\
&\leqslant H(x) + H(y \mid x) + H(z \mid y) = && \text{(циклический сдвиг в треугольнике)}\\
&= H(x) + 2H(Y \mid X)
\end{aligned}$$

Найдем какое-то распределение на углах, энтропия которого хотя бы $H(x) + 2H(Y \mid X)$, тогда эта сумма будет не более $\log(\#\angle))$.

Пусть мы выбрали случайный треугольник (x, y, z). Оставим x и выберем для него найдем случайный треугольник с x и берем из него следующую за x вершину y'. Повторяем эту операцию еще раз для x и находим z'. Тогда (x, y', z') — угол.

$$\begin{aligned}
H(x,y',z') &= H(x) + H(y' \mid x) + H(z' \mid x, y') = && \text{(Так как } y' \text{ и } z' \text{ независимы при выбранном } x\text{)}\\
&= H(x) + H(y' \mid x) + H(z' \mid x) = && \text{(Выбор аналогичный)}\\
&= H(x) + 2H(y' \mid x)
\end{aligned}$$

$H(x)$ здесь совпадает с $H(x)$ выше, так как мы выбираем треугольник и вершину аналогично.

y' выбирается при фиксированном x также, как и выше (выбрали случайный треугольник и в нем вершиной после x будет y').

Таким образом, мы нашли распределение с такой же энтропией. ■

1.5 Теория кодирования

Код — отображение алфавита $C\colon \Sigma \to \{0, 1\}^*$.

Что хочется требовать?

1. Однозначное декодирование. При этом не обязательно у каждой строки $\{0, 1\}^*$ есть слово, но склейки нет.

2. Префиксный код — то есть код каждого символа не является префиксом кода другого. Очевидно, из этого следует предыдущий пункт.

[2]Очевидно, что взвешивать кучки разного размера, информацию извлечь не получиться даже по Хартли

Теорема 1.5.1. Любой однозначно декодируемый код можно переделать в префиксный с сохранением длин кодовых слов.

□ Пусть есть $c_1, \ldots, c_n$ — кодовые слова.

Для префиксного кода $\sum 2^{-|c_i|} \leqslant 1$, причем, если если выполнено это неравенство, то есть префиксный код.

Докажем, что для любого декодируемого кода выполнено такое неравенство.

Построим многочлен для всех слов длины L.

$$p(x, y) = \left(\sum_i p_i(x, y)\right)^L = \sum_{j=L} M_j(x, y).$$

Здесь $p_i(x, y)$ — моном, соответствующий i-ому символу в алфавите и равный

Посчитаем $p(\frac{1}{2}, \frac{1}{2})$.

$$p(\tfrac{1}{2}, \tfrac{1}{2}) = \sum_{j=L} M_j(\tfrac{1}{2}, \tfrac{1}{2}) \leqslant \sum_{j=L}^{\max_i c_i} 2^j \cdot 2^{-j} = \mathcal{O}(L).$$

Посчитаем еще раз по второму представлению

$$p(\tfrac{1}{2}, \tfrac{1}{2}) = \left(\sum_i 2^{-|c_i|}\right)^L.$$

Если сумма в скобках больше 1, получаем экспоненциальную оценку снизу. Следовательно, для больших N она обгонит линейную. Противоречие. ■

Теорема 1.5.2 (Шеннон)**.** Пусть есть множество Σ, и с вероятностью p_i получаем i-й символ. Тогда

$$\sum_i p_i|c_i| \geqslant H(p) \qquad c_i \text{ — однозначно декодируемы.}$$

□

$$H(p) - \sum_i p_i|c_i| = \sum p_i \log \frac{2^{-|c_i|}}{p_i} \leqslant \qquad \text{(Неравенство Йенсена)}$$

$$\leqslant \log \sum p_i \cdot \frac{2^{-|c_i|}}{p_i} \leqslant \qquad \text{(Неравенство Крафта)}$$

$$<= 0$$

■

Теорема 1.5.3 (Шеннон)**.** Существует такой код, что [a]

$$\sum_i p_i \cdot |c_i| \leqslant H(p) + 1.$$

[a]Единичка обязательно возникает, так как мы приводим непрерывную энтропию к дискретной величине

□ Угадаем длины кодов, чтобы выполнялось неравенство ???. Пусть $|c_i| = \lceil \log \frac{1}{p_i} \rceil$,

$$\sum 2^{-|c_i|} = \sum 2^{-\rceil \frac{1}{p_i} \rceil} \leqslant \sum p_i \leqslant 1.$$

■

1.6 Код Шенона-Фано

Отсортируем вероятности по убыванию $p_1 \geqslant p_2 \geqslant \ldots \geqslant p_n$. Затем уложим их в отрезок $[0, 1]$.

Разделим отрезок пополам и скажем, что слева кодовые слова начинается с 0, справа с 1, а центральный p_i будет начинаться с нуля, если это p_1, с единицы, если p_n, и, наконец, иначе выдираем любое значение.

Далее рекурсивно запускаемся на группе нулей и на группе единиц.

Когда остался один кусок, останавливаемся.

Теорема 1.6.1.

$$\sum_0^n p_i \cdot |c_i| \leqslant H(p) + \mathcal{O}(1), \quad n \to \infty,\ \mathcal{O}(1) \approx 3 \text{ или } 5.$$

☐ Упражнение со зведочкой. ■

1.7 Код Хаффмана

Опять отсортируем $p_1 \geqslant p_2 \geqslant \ldots \geqslant p_n$. Возьмем p_{n-1} и p_n. Заменим их на один символ с вероятностью $p_n + p_n - 1$, теперь по индукции строим код для $n - 1$ символа.

Теперь если объединенному символу соответствовал код $\overline{c}$, то для p_{n-1} задаем код $\overline{c0}$, а для p_n код $\overline{c1}$

Проверим, что $\sum_{i=1}^n p_i|c_i| \leqslant H(p) + 1$, причем $\forall c_i'\colon \sum_{i=1}^0 p_i|c_i| \leqslant \sum_{i=1} p_i|c_i|$

Достаточно доказать второе, а потом сравнить с кодом Шеннона и получить нужное неравенство.

Рассмотрим набор $c_1', \ldots c_n'$. Возьмем два минимальных c_{n-1}' и c_n'. Заметим, что можно поменять их с символами максимальной длины c_i' и c_j', при этом длина кода не увеличится.

Изучим коды c_{n-1}' и c_n'. Пусть они не имеют вид $\overline{v0}$ и $\overline{v1}$.

- Пусть $|c_{n-1}'| \leqslant |c_n'|$. Посмотрим на c_{n-1}': не умаляя общности он будет заканчиваться на 0 ($\overline{s0}$). Заменим c_n' на $s1$. Если вдруг кто-то уже имел такой код, это и есть c_n', так как имеет максимальную длину.

1.8 Арифметическое кодирование

Уложим вероятности аналогично не отрезок, при этом не обязательно в порядке убывания.

Назовем **стандартным** интервал $[\overline{0v0}, \overline{0v1})$. Найдем максимальный стандартный интервал в отрезке p_i. Тогда v будет кодом p_i.

☐ Если рассмотреть отрезок $[a, b]$, есть стандартный интервал длиной $\frac{b-a}{8}$. Упражнение ■

Лекция 4: † 22 April

1.9

Пусть есть алфавит Σ размером k, ‘кодер‘ $E\colon [k]^n \to \{0, 1\}^{L_n}$ и ‘декодер‘ $D\colon \{0, 1\}^{L_n} \to [k]^n$.

Пусть есть распределение на буквах $p_1, p_2, \ldots p_k$. [3]

Обозначим $\varepsilon_n \coloneqq Pr[D(E(x))] \neq x$, где $|x| = n$. Хотим $\varepsilon_n \to 0$.[4]

[3]считаем, что слово состоит из независимых букв

[4]если сделать равенство, то особого сжатия не будет

Теорема 1.9.1. Если $L_n > \lceil hn \rceil$ и $h > H(p)$, то кодирование есть. Если $L_n < \lceil hn \rceil$ и $h < H(p)$, то $\varepsilon_n \to 1$.

□ Будем называть код W δ-типичным, если

$$\forall i \left| n_{\frac{i}{n}} - p_i \right| \leqslant \delta, \quad n_i = \#\text{входа буквы}.$$

Зафиксируем $\delta = n^{-0.02}$.

- Докажем, что можем закодировать такие типичные слова в первой части. Пусть X_{ij} — характеристическая функция того, что в слове на позиции j находится буква i.

  Также рассмотрим $X_i = \sum_j X_{ij}$ и применим неравенство Чебышева:

  $$Pr[|X_i - \mu| \geqslant \delta n] \leqslant \frac{\mathrm{Var}[x_i]}{(\delta n)^2} = \frac{np_i(1-p_i)}{(\delta n)^2} = \mathcal{O}\left(\frac{1}{\delta^2 n}\right)^{5}.$$

  Так как букв константное количество, вероятность нетипичности все равно останется очень маленьким и будет стремиться к нулю.

  Теперь докажем, что типичных слов не очень много. Количество слов, где буквы встречаются в количествах $n_1, \ldots, n_k$ равно

  $$N = \frac{n!}{n_1! \cdot \ldots \cdot n_k!}.$$

  $$\begin{aligned}\log N = & & \left(\text{так как } n! = \mathrm{poly}(n)\left(\tfrac{n}{e}\right)^n\right)\\ &= \log\left(\left(\frac{n}{n_1}\right)^{n_1} \cdot \left(\frac{n}{n_2}\right)^{n_2} \cdot \ldots \cdot \left(\frac{n}{n_k}\right)^{n_k}\right) + \mathcal{O}(\log n) = \\ &= \sum n_i \log \frac{n}{n_i} + \mathcal{O}(\log n) = & (n_i \text{ по определению})\\ &= n \sum (p_i + \delta_i) \cdot \log \frac{1}{p_i + \delta_i} + \mathcal{O}(\log n)\\ & & (|\delta_i| < \delta, \text{ так как типичное, } \delta_i \text{ — отклонение } i\text{-ой буквы в языке})\end{aligned}$$

  Теперь оценим число типичных слов

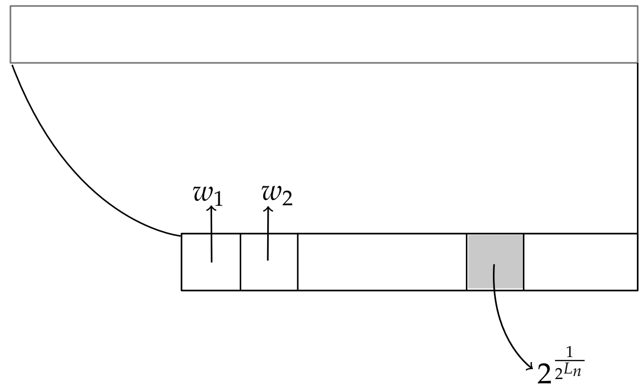
  $$\begin{aligned}&\log\Big(\#(\delta\text{-типичных слов})\Big) \leqslant\\ &\qquad \leqslant \log\left(n^k \cdot \max_{\delta_i} N\right) \leqslant\\ &\qquad \leqslant \max_{\delta_i} H(p_1 + \delta_1, p_2 + \delta_2, \ldots) \cdot n + \mathcal{O}(\log n) = \qquad (\text{Переход за кадром}^{6})\\ &= nH(p) + \mathcal{O}(\delta \cdot n)\end{aligned}$$

  Если теперь кодер может отобразить инъективно все типичные слова в набор битовых слов длины hn, при этом ошибаться он будет на нетипичных, количество которых стремиться к нулю.

- Во второй части докажем, что мы не сможем закодировать все типичные слова.

  Покажем, что вероятность того, что мы выкинем δ-типичное слово очень мала. Пусть $L_n \leqslant hn$. Посмотрим на любое кодовое распределение слов. Покажем, что вероятность по нашему определению для δ-типичных слов больше, чем $\frac{1}{2^{L_n}}$.

  $$\begin{aligned}Pr[w] &= p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_k^{n_k} =\\ &= 2^{-\sum_{i=1}^k (p_i + \delta) \log \frac{1}{p_i} n} \leqslant\\ &\leqslant 2^{-H(p)n + \mathcal{O}(\delta n)}\end{aligned}$$



С какой вероятностью декодер декодер ответит правильно?

$$Pr[\text{правильного ответа}] \leqslant 2^{L_n} \cdot \max_w Pr[w] \leqslant 2^{((L_n - H(p))\cdot n + \mathcal{O}(\delta n)} \to 2^0.$$

■

Лекция 5: †

29 April

Chapter 2

Коммуникационная сложность

Пусть у нас есть два игрока: Алиса и Боб. Они могут отправлять друг другу сообщения и хотят посчитать функцию (или отношение) $f\colon X \times Y \to \mathcal{O}$.

2.1 Модели

2.1.1 Детерминированная модель

Формализуем это в виде бинарного дерева.

В вершинах будем записывать ходящего игрока, в листьях результаты вычислений.

Левое ребро обозначает сообщение 0, а правое — 1.

Обозначение. *$D(f)$ — минимальная глубина дерева.*

2.1.2 Вероятностная модель

Теперь Алиса и Боб могут подбрасывать монетки. Либо эти монетки (оракулы) публичны (оба видят значения), либо приватными (тогда никто не видит, кроме пользователя).

Так как Алиса или Боб в случае публичного оракула, они могут закрыть глаза на сообщения другого, поэтому публичный протокол не меньше приватного.

Скажем, что **протокол отработал корректно**, если

$$\forall x, y\colon Pr_r[\pi(x,y) = f(x,y)] \geqslant \frac{2}{3}, \quad \pi(x,y) \text{ — результат работы.}$$

Обозначение. *$R^{pub}_{\frac{2}{3}}$ — аналогично оптимальная высота.*

2.2 Нижние оценки для детерминированного случая

Пусть наша функция $f\colon X \times Y \to \mathbb{O}$. Запишем для нее коммуникационную матрицу M размера $|X| \times |Y|$, где $M_{x,y} = f(x,y)$.

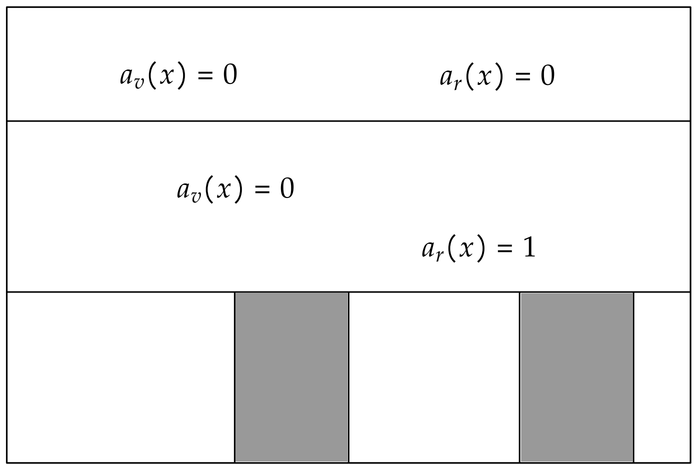
Рассмотрим $R_v \subseteq X \times Y\colon (x,y) \in R_v \iff$ протокол приводит в v.

Лемма 1. $R_v = X_v \times Y_v$ — прямоугольник.

□ Пусть (x,y) и (x',y') принадлежат R_v. Тогда (x,y') и (x',y) тоже принадлежат R_v, так как $a(x) = a(x')$ и $b(y) = b(y')$.

А из этого следует, что это комбинаторный прямоугольник. ■

□ Посмотрим на картинку. Пусть Алиса перешла по какому-то ребру. Вся таблица разделилась на две части.



И так далее. ■

В прямоугольнике для листа у всех элементов одинаковый ответ. То есть исходную матрицу можно разбить на комбинаторные прямоугольники, причем они естественно не пересекаются.

Хотим показать оценку на количество таких прямоугольников.

2.3 Метод ранга

Разбирался на практике:

$$\mathrm{rk}_{\mathbf{R}}(M_f) \leqslant \#\text{ одноцветных прямоугольников в разбиении.}$$

Здесь f — функция.

Для функции EQ $=: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, будет диагональная матрица. Поэтому одноцветных прямоугольников будет не меньше 2^n, а тогда коммуникационная сложность хотя бы n.

2.4 Fooling Set

Рассмотрим коммуникационную матрицу. Пусть мы хотим выбрать некоторое множество

$$S = \{(x_1, y_1), (x_2, y_2), \ldots\},$$

такое что каждая пара точек не лежит в одном прямоугольнике.

Если две клетки в одном прямоугольнике, оставшиеся вершины тоже лежат в нем.

Тогда нужно для всех $i, j, i \neq j$ либо (x_i, y_j), либо (x_j, y_i) покрашена в другой цвет.

Для EQ легко получить ту же оценку. Плюс, как как нужен хотя бы один лист для нуля, n не хватит, следовательно, $D(\mathrm{EQ}) = n + 1$.

Теорема 2.4.1. Если существует Fooling set размера s, то $\mathrm{rk}_R\, s \geqslant s$.

2.4.1 Пример, не соответствующий никакому протоколу

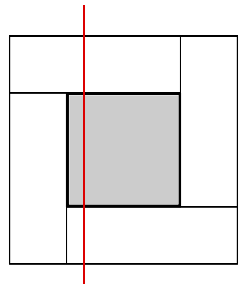


Figure 2.1: bad recd

Пусть χ — минимальное число одноцветных прямоугольников в разбиении.

Теорема 2.4.2 (GPW, 16). Существует f для которой

$$D(f) \geqslant \log^{2-\varepsilon} \chi(M_f).$$

☐ Без доказательства ■

2.5 Связь со схемами

Теорема 2.5.1 (Шеннон). Существует $f\colon \{0,1\}^n \to \{0,1\}$, такая что $L(f) \geqslant \Omega\left(\frac{2^n}{n}\right)$, где L — оптимальный размер схемы.

☐ Всего функций такого вида 2^{2^n}, так как можно задать таблицей истинности.

Посчитаем число схем. Это ациклический граф и то, что записано в его узлах.

Пусть каждая вершина (S штук) выбирает себе двух предков. Так же в каждую вершину нужно что-то записать и на ребре можно ставить отрицание: хватит 3 бит. Еще есть входные данные (n штук).

Итого: $2^{S\cdot 2(\log S+\log n)+3S}$.

Схем должно быть не меньше количества функций

$$2^{S\cdot 2(\log S+\log n)+2S} \geqslant 2^{2^n}.$$

Отсюда получаем нужное неравенство. ■

Открытый вопрос: Можно ли предъявить $f \in \mathrm{NP}$, что $L(f) \geqslant 10n$

2.6 KW_f

Пусть нам дана $f\colon \{0,1\}^n \to \{0,1\}$.

Алиса получает число $x \in f^{-1}(1)$, а Боб $y \in f^{-1}(0)$. Его цель

Теорема 2.6.1 (Rarchmer-Wigderson, 1990). $L(f)$ — размер минимальной формулы для f, согда $L(f)$ — размер минимального протокола для RW_f

□

$\boxed{1 \Longrightarrow 2}$ Нарисуем дерево вверх корнем. Также спустим все отрицания к листьям. Пусть в узле считается функция $f = g \vee h$, где g и h — соседи f.

Тогда $f(x) = 1$, $f(y) = 0$ и $f(y) = 0$. Пусть Алиса посылает информацию, где 1, то есть куда нам нужно спуститься. Далее Боб делает аналогично.

$\boxed{2 \Longrightarrow 1}$

■