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

Тамарин Вячеслав

June 18, 2021

Contents

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

Chapter 1

Информация по Хартли

1.1 Базовые свойства

Лекция 1
1 April

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. Хотим ввести некоторую *меру информации*, то есть сколько информации мы узнаем, когда вытягиваем некоторый элемент из A.

Самый простой вариант — число битов требуемое для записи объекта.

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

1. χ дает нам оценку на длину описаний
2. $\chi(A \cup B) \leqslant \chi(A) + \chi(B)$
3. Если $A \coloneqq B \times C$, то чтобы описать элемент из A, достаточно описать элемент из B и из C:

$$\chi(A) \leqslant \chi(B) + \chi(C).$$

Определение 1: Информация по Хартли

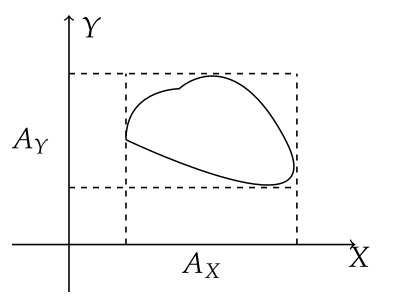
Пусть A — некоторое конечное множество. За ***информацию в множестве*** A будем принимать следующую величину:[a] $\chi(A) \coloneqq \log|A|$

[a]Здесь и далее под $\log x$ подразумевается двоичный логарифм.

Замечание. Очевидно, второе свойство выполнено для такого определения (проблемы есть, только когда одно из множеств одноэлементно и не содержится во втором). В третьем даже равенство.

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

Пусть $A \subseteq X \times Y$. Обозначим ***проекции*** A_X и A_Y.



Утверждение. Пусть $A \subseteq X \times Y$ и конечно. Тогда для $\chi(A)$ выполнены следующие свойства:

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

□

1. Очевидно
2. $|A_X| \leqslant |A|$
3. $\chi(A) \leqslant \chi(A_X \times A_Y) \leqslant \chi(A_X) + \chi(A_Y)$

■

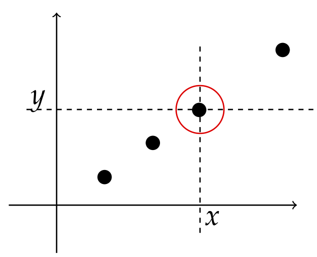


Figure 1.1: Диагональное множество

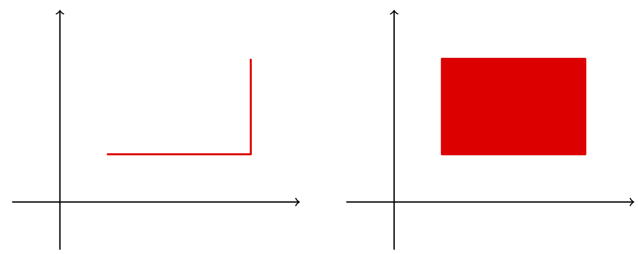


Figure 1.2: Неотличимые множества

Пример 1.1.1

Рассмотрим такой пример (см. рисунок **??**): множество из точек на диагонали. Здесь, зная первую координату, можно сразу понять вторую.

Попробуем усилить третье свойство:

3'. $\chi(A) \leqslant \chi(A_X) + \chi_{Y|X}(A)$, 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 множества $\chi_{Y|X}$ просто обнуляется и неравенство переходит в равенство.

Пример 1.1.2

Но если взять одно множество из двух отрезков, а второе из всего прямоугольника, возникнут проблемы (см. рисунок **??**).

Во-первых, на первой картинке, передав x-координату столбца, придется передавать и y тоже. Во-вторых, мы не сможем отличить эти множества по одной координате.

Лемма 1. Пусть $A \subseteq A_1 \times A_2 \times A_3$ конечно. Тогда[a]

$$2\chi(A) \leqslant \chi(A_{12}) + \chi(A_{13}) + \chi(A_{23}).$$

[a] Здесь $A_{ij} = \{(x, y) \mid x \in A_i \wedge y \in A_j \wedge \exists z \colon (x, y, z) \in A\}$

□ Перепишем неравенство из условия:

$$\begin{aligned} 2\log|A| &\leqslant \log|A_{12}| + \log|A_{13}| + \log|A_{23}| \\ \log|A|^2 &\leqslant \log(|A_{12}| \cdot |A_{13}| \cdot |A_{23}|) \\ |A|^2 &\leqslant |A_{12}| \cdot |A_{13}| \cdot |A_{23}| \qquad \text{(log — возрастающая функция)} \end{aligned}$$

Пусть $A = \{(x, y, z) \mid x \in A_1,\ y \in A_2,\ z \in A_3\}$

Сгруппируем элементы A следующим образом: $A = \bigsqcup B_{ij}$, где

$$B_{ij} = \{(x_i, y_j, z_k) \mid 1 \leqslant k \leqslant K_{ij},\ x_i \in A_1,\ y_j \in A_2,\ z_k \in A_3\}.$$

То есть фиксируем x_i, y_j, а z_k меняется в допустимых пределах. При этом $\sum_{i,j} K_{ij} = |A|$, так как это сумма размеров B_{ij}, которая равна размеру A.

Теперь рассмотрим множество

$$A_{13,23} = \{(x_i, z_s, y_j, z_l) \mid (x_i, z_s) \in A_{13},\ (y_j, z_l) \in A_{23}\}.$$

Если внимательно посмотреть на определение, можно понять, что $|A_{13,23}| = |A_{13}| \cdot |A_{23}|$.

Теперь опишем элементы, которые заведомо есть в $A_{13,23}$, чтобы оценить его размер снизу. Для этого зафиксируем x_i, y_j, тогда все z_k из B_{ij} точно сюда подходят в качестве z_s или z_l. То есть таких четверок не меньше чем $|B_{ij}|^2 = K_{ij}^2$. А по всем i, j получается, что $|A_{13,23}| \geqslant \sum K_{ij}^2$.

Таким образом,

$$|A_{13}| \cdot |A_{23}| = |A_{13,23}| \geqslant \sum K_{ij}^2.$$

При этом заметим, что $|A_{12}| = |\{B_{ij} \neq \varnothing\}|$, поскольку если $B_{ij} \neq \varnothing$, то пара (x_i, y_j) лежит в A_{12}. В таком случае

$$|A_{12}| = |\{B_{ij} \neq \varnothing\}| = |\{K_{ij} \neq 0\}| = n$$

Получим итоговое неравенство

$$\begin{aligned} |A_{12}| \cdot (|A_{13}| \cdot |A_{23}|) \geqslant n \cdot \sum K_{ij}^2 &\geqslant \qquad \text{(неравенство о средних)} \\ &\geqslant \left(\sum K_{ij}\right)^2 = |A|^2 \end{aligned}$$

А это то, что мы и хотели доказать.

Пояснение неравенства о средних:

$$n \cdot \sum K_{ij}^2 \geqslant \left(\sum K_{ij}\right)^2$$
$$\sqrt{\left(\sum K_{ij}\right)^2} \leqslant \sqrt{n \cdot \sum K_{ij}^2}$$
$$\frac{\sum K_{ij}}{n} \leqslant \frac{\sqrt{n \cdot \sum K_{ij}^2}}{n}$$
$$\frac{\sum K_{ij}}{n} \leqslant \sqrt{\frac{\sum K_{ij}^2}{n}}$$

Это неравенство между средним арифметическим и средним квадратичным. ■

1.2 Угадывание числа

Обозначим $[n] \coloneqq \{1, \ldots, n\}$.

Симметричный вариант Пусть есть два игрока, первый загадывает число от 1 до n, а второй должен его угадать. Сколько вопросов необходимо задать, чтобы угадать число?

Есть два варианта игры:

- *Адаптивная*, когда второй игрок получает ответ на вопрос сразу
- *Неадаптивная*, когда он пишет сначала все вопросы, а потом получает на них все ответы.

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

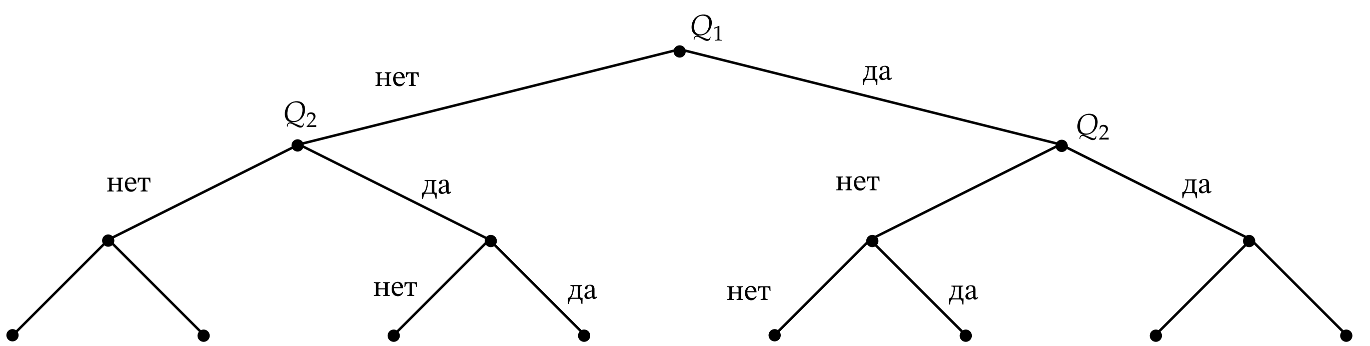


Figure 1.3: Граф вопросов

Теперь подумаем с точки зрения теории информации. Пусть h — число запросов, Q_i — ответ на i-ый запрос (один бит) по некоторому протоколу,

$$B \coloneqq Q_1 \times \ldots \times Q_h,$$

Мы хотим минимизировать h.

Рассмотрим $[n] \times B$ — все возможные пары по всем возможным значениям искомого числа и B. Нас интересует множество $A \subseteq [n] \times B$, соответствующее некоторым корректным запросам:

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

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. $\chi(A) \geqslant \log n$, так как уже первая проекция A имеет n элементов.

   С другой стороны, $\chi(A) \leqslant \chi(A_B) + \chi_{[n]|B}(A) = \chi_B(A) \leqslant \chi(B) \leqslant \sum_{i=1}^{h} \chi(Q_i)$.

   А так как $\chi(Q_i) = 1$, $h \geqslant \log n$.

Асимметричный вариант Пусть теперь за ответ «да» мы платим 1, а за «нет» 2, вариант игры адаптивный. И мы хотим минимизировать не число запросов, а стоимость в худшем случае.

Будем делить запросом множество «пополам» с точки зрения стоимости.

Пусть Q_i — ответ на вопрос «верно ли, что загаданное число m лежит в множестве T_i?

Пусть A_i — множество элементов, в котором может лежать m после первых $i-1$ вопросов. 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 $T_i \subseteq A_i$, что

$$2\big(\chi(A_i) - \chi(\underbrace{T_i}_{A_{i+1}})\big) = \chi(A_i) - \chi(\underbrace{A_i \setminus T_i}_{A_{i+1}}).$$

Где разность $\chi(A_i) - \chi(A_{i+1})$ обозначает количество информации, которое мы получаем задав i-ый вопрос. То есть слева получили ответ «нет» ($m \in T_i$), а справа «да» ($m \in A_i \setminus T_i$).

Распишем по определению:

$$\begin{aligned} 2\big(\log|A_i| - \log|T_i|\big) &= \log|A_i| - \log|A_i \setminus T_i| \\ \log|A_i| &= 2\log|T_i| - \log|A_i \setminus T_i| \\ |A_i| &= \frac{|T_i|^2}{|A_i \setminus T_i|} \end{aligned}$$

Обозначим $|A_i| = k$, $|T_i| = t$ и решим уравнение:

$$k = \frac{t^2}{k-t} \iff t^2 = k(k-t) = k^2 - kt \iff$$
$$t^2 + kt - k^2 = 0 \iff t = \frac{-k \pm \sqrt{k^2 + 4k^2}}{2} = k \cdot \frac{-1+\sqrt{5}}{2} = k \cdot \varphi$$

То есть нужно выбирать T_i, чтобы $|T_i| = \varphi \cdot |A_i|$.

Тогда «средняя цена» бита будет равна $2(\chi(A_i) - \chi(T_i)) = 2\log\frac{1}{\varphi}$

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.

Упражнение (Задача про взвешивания монеток). Есть n монеток и рычажные весы. Хотим найти фальшивую (она одна).

1. Пусть $n = 30$ и мы знаем тяжелее фальшивка или легче. 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.

То есть информация по Хартли нам помогла понять два раза, что 3 взвешиваний не хватит. Но её не хватает для того, чтобы решить 3 задачу.

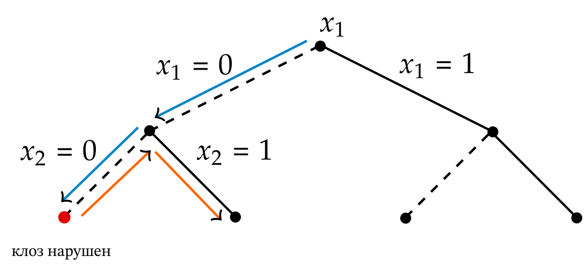
Лекция 2
8 April

1.3 Задача выполнимости

Вход: $\Phi = \bigwedge_{i=1}^{n} C_i$ — формула в КНФ.

Будем подставлять значения во все переменные по очереди, тем самым перемещаться по двоичному дереву.

Подставим $x_i = 0$. Если пока клозы не нарушены, подставляем далее $x_{i+1} = 0$ и проверяем клозы аналогично. Если же один из клозов нарушился, вернемся на шаг назад и подставим $x_i = 1$.



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

Таким образом, если есть выполняющий набор, и мы на первом шаге подберем верное значение первой переменной, на второй для второй и так далее, то получим оптимальное решение. Однако не всегда такой алгоритм работает быстро. Пример: задача про рассадку голубей.

1.4 Рассадка голубей

У нас есть $n + 1$ голубь и n клеток, хотим показать, что нельзя рассадить в клетку по одному голубю.

Введем для каждой пары (голубь, клетка) переменную x_{ij}, которая равна 1, если i-ый голубь сидит в j-ой клетке, и $x_{ij} = 0$ иначе.

Тогда, чтобы рассадка была удачной, должны выполняться следующие условие:

- для всех $i \in [n + 1]$ верно $\prod_{j=1}^{n}(1 - x_{ij}) = 0$. То есть для каждого голубя нашлась клетка.
- для всех $i, i' \in [n + 1]$ и $j \in [n]$, где $i \neq i'$ верно $x_{ij} \cdot x_{i'j} = 0$. То есть никакие два голубя не сидят в одной клетке.

Пусть один игрок загадал расстановку голубей, а второй хочет найти дизъюнкт, для которого нарушается эта расстановка. Игра с монетками позволяет показать, что предыдущий алгоритм плохо работает на этой модели.

Chapter 2

Информация по Шеннону

2.1 Определения и свойства

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.[2]

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

1. *монотонность*: $\mu(U_M) \geqslant \mu(U_{M'})$, если $|M| \geqslant |M'|$;
2. *аддитивность*: $\mu(\alpha, \beta) = \mu(\alpha) + \mu(\beta)$, если α и β независимы;
3. *непрерывность*: $\mu(B_p)$ непрерывно по $p \in [0, 1]$, где B_p — распределение Бернулли для p.
4. *согласованность с условной вероятностью*:

$$\mu(B_p, \alpha) = \mu(B_p) + \Pr[B_p = 0] \cdot \mu(\alpha \mid B_p = 0) + \Pr[B_p = 1] \cdot \mu(\alpha \mid B_p = 1).$$

Этим аксиомам удовлетворяет примерно одна функция $\mu(X) \coloneqq \sum p_i \log \frac{1}{p_i}$ с точностью до домножения на константу.

2.1.1 Энтропия

Определение 2: Энтропия

Для случайной величины α с вероятностями событий $(p_1, p_2, \ldots, p_n)$ меру

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

будем называть ***энтропия*** и обозначать H.[a]

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

[2] (α, β) — распределение на парах.

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

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

Замечание. Энтропия равномерного распределения U_n равна $\log n$, так как вероятность каждого события $\frac{1}{n}$.

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

$$H(B_p) = H(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}.$$

Теорема 2.1.1. $H(\alpha) \leqslant \log|\operatorname{supp}(\alpha)|$

☐ Применим неравенство Йенсена

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

■

Теорема 2.1.2. $H(\alpha, \beta) \leqslant H(\alpha) + H(\beta)$

☐

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

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

Заметим, что $p_i = \sum\limits_j p_{i,j}$ и $p_j = \sum\limits_i p_{i,j}$.

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

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

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

■

2.1.2 Условная энтропия

Определение 3: Условная энтропия

Энтропией α ***при*** $\beta = b$ будем называть энтропию распределения α при условии, что $\beta = b$, то есть:

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

Тогда ***энтропией*** α ***при условии*** β назовем среднее значение по β энтропии α при $\beta = b$:

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

Свойства.

1. $H(\alpha \mid \beta) \geqslant 0$

2. $H(\alpha \mid \beta) = 0 \iff \alpha$ однозначно определяется β

   □

$$H(\alpha \mid \beta) = 0 \iff \forall b\; H(\alpha \mid \beta = b) = 0 \iff \forall b: \; \Pr[\alpha = i \mid \beta = b] = 1$$

■

3. $\forall f\colon\; H(\alpha \mid \beta) \geqslant H(f(\alpha) \mid \beta)$

   □ Заметим, что достаточно показать неравенство $H(\alpha \mid \beta = b) \geqslant H(f(\alpha) \mid \beta = b)$, поэтому будем считать, что все дальнейшие рассуждения проводятся при условии $\beta = b$.

   Пусть α принимает значения $a_1, \ldots, a_n$ с вероятностями $p_1, \ldots, p_n$ соответственно, а $f(\alpha)$ принимает значения $f_1, \ldots, f_k$ так, что для $\forall j\; f_i \coloneqq f(a_{i_j})$ (разбили a_i-ые на k частей). Обозначим $P_i \coloneqq \sum_j p_{i_j}$ - вероятность f_i.

$$\begin{aligned}
H(\alpha \mid \beta = b) &= \sum_i \Pr[\alpha = i \mid \beta = b] \cdot \log \frac{1}{\Pr[\alpha = i \mid \beta = b]} && \text{(по определению } H\text{)}\\
&= \sum_i p_i \cdot \log \frac{1}{p_i} && \text{(по определению } p_i\text{)}\\
&= \sum_i \left(\sum_j p_{i_j} \cdot \log \frac{1}{p_{i_j}} \right) && \text{(перегруппировали слагаемые)}\\
&\geqslant \sum_i \left(\log \frac{1}{P_i} \cdot \sum_j p_{i_j} \right) && \text{(оценили } P_i \geqslant p_{i_j}\text{)}\\
&= \sum_i P_i \cdot \log \frac{1}{P_i} && \text{(свернули сумму в } P_i\text{)}\\
&= H(f(\alpha) \mid \beta = b)
\end{aligned}$$

■

4. $H(\alpha, \beta) = H(\alpha) + H(\beta \mid \alpha) = H(\beta) + H(\alpha \mid \beta)$

   □

$$\begin{aligned}
H(\beta) + H(\alpha \mid \beta) &= \sum_j p_j \log \frac{1}{p_j} + \sum_j H(\alpha \mid \beta = j) \cdot \Pr[\beta = j] =\\
&= \sum_j p_j \log \frac{1}{p_j} + \sum_j \left(\sum_i \Pr[\alpha = i \mid \beta = j] \cdot \log \frac{1}{\Pr[\alpha = i \mid \beta = j]} \right) \cdot \Pr[\beta = j] =\\
&= \sum_j p_j \log \frac{1}{p_j} + \sum_j \left(\sum_i \Pr[\alpha = i, \beta = j] \cdot \log \frac{\Pr[\beta = j]}{\Pr[\alpha = i, \beta = j]} \right) =\\
&= \sum_j \sum_i p_{i,j} \log \frac{1}{p_j} + \sum_j \left(\sum_i p_{i,j} \cdot \log \frac{p_j}{p_{i,j}} \right) = \sum_{i,j} p_{i,j} \log \frac{1}{p_{i,j}} = H(\alpha, \beta)
\end{aligned}$$

   подсказка: $\Pr[A, B] = \Pr[A \mid B] \cdot \Pr[B]$ ■

5. $H(\alpha, \beta) \geqslant H(\alpha)$

   □ Очевидно из предыдущего свойства. ■

6. $H(\alpha) \geqslant H(\alpha \mid \beta)$
   □

$$\begin{aligned}
H(\alpha \mid \beta) - H(\alpha) &= \sum_j \sum_i \left(\Pr[\alpha = i \mid \beta = j] \log \frac{1}{\Pr[\alpha = i \mid \beta = j]}\right) \cdot \Pr[\beta = j] - \sum p_i \log \frac{1}{p_i} \\
&= \sum p_{i,j} \log \frac{p_j}{p_{i,j}} - \sum p_{i,j} \log \frac{1}{p_i} \\
&= \sum p_{i,j} \log \frac{p_i p_i}{p_{i,j}} \\
&\leqslant \log \sum p_i p_j = \log 1 = 0 && \text{(по неравенству Йенсена)}
\end{aligned}$$

■

7. Формула условной энтропии через отдельные вероятности
   □

$$\begin{aligned}
H(\alpha \mid \beta) &= \sum_j H(\alpha \mid \beta = b_j) \cdot \Pr[\beta = b_j] \\
&= \sum_j \left(\Pr[\beta = b_j] \cdot \sum_i \Pr[\alpha = a_i \mid \beta = b_j] \cdot \log \frac{1}{\Pr[\alpha = a_i \mid \beta = b_j]}\right) \\
&= \sum_{i,j} p_{i,j} \cdot \log \frac{p_j}{p_{i,j}}
\end{aligned}$$

■

8. $H(\alpha \mid \beta) \geqslant H(\alpha \mid \beta, \gamma)$, причем равенство достигается, если $(\gamma \mid \beta)$ не зависимо с $(\alpha \mid \beta)$.
   □ Обозначим $p_{ijk} := \Pr[\alpha = i, \beta = j, \gamma = k]$, аналогично p_{ij}, p_{jk}.
   Распишем левую и правую части по определению:

$$H(\alpha \mid \beta) = \sum_{ij} p_{ij} \log \frac{p_j}{p_{ij}}$$

$$H(\alpha \mid \beta, \gamma) = \sum_{ij} \left(\sum_k p_{ijk} \log \frac{p_{jk}}{p_{ijk}}\right)$$

Достаточно показать неравенство для фиксированных i, j:

$$\begin{aligned}
\sum_k p_{ijk} \log \frac{p_{jk}}{p_{ijk}} &= p_{ij} \sum_k \frac{p_{ijk}}{p_{ij}} \log \frac{p_{jk}}{p_{ijk}} && \text{(чтобы сумма коэффициентов была 1)} \\
&\leqslant p_{ij} \log \left(\sum_k \frac{p_{jk}}{p_{ij}}\right) && \text{(по неравенству Йенсена)} \\
&= p_{ij} \log \left(\frac{\sum_k p_{jk}}{p_{ij}}\right) \\
&= p_{ij} \log \frac{p_j}{p_{ij}}
\end{aligned}$$

Для независимых $(\alpha \mid \beta)$ и $(\gamma \mid \beta)$, то есть:

$$\Pr[\alpha, \gamma \mid \beta] = \Pr[\alpha \mid \beta] \cdot \Pr[\gamma \mid \beta] \implies$$
$$\frac{p_{ijk}}{p_j} = \frac{p_{ij}}{p_j} \cdot \frac{p_{jk}}{p_j} \implies \frac{p_{ijk}}{p_{jk}} = \frac{p_{ij}}{p_j} \implies$$
$$\Pr[\alpha = i \mid \beta = j, \gamma = k] = \Pr[\alpha = i \mid \beta = j]$$

Теперь можем расписать энтропию:

$$\begin{aligned}
H(\alpha \mid \beta, \gamma) &= \sum_{jk} p_{jk} H(\alpha \mid \beta = j, \gamma = k) \\
&= \sum_{jk} p_{jk} \left(\sum_i \Pr[\alpha = i \mid \beta = j, \gamma = k] \log \frac{1}{\Pr[\alpha = i \mid \beta = j, \gamma = k]} \right) \\
&= \sum_{jk} p_{jk} \left(\sum_i \Pr[\alpha = i \mid \beta = j] \log \frac{1}{\Pr[\alpha = i \mid \beta = j]} \right) && \text{(по независимости)} \\
&= \sum_j \left(\sum_k p_{jk} \right) \cdot \left(\sum_i \Pr[\alpha = i \mid \beta = j] \log \frac{1}{\Pr[\alpha = i \mid \beta = j]} \right) \\
&= \sum_j p_j H(\alpha \mid \beta = j) \\
&= H(\alpha \mid \beta)
\end{aligned}$$

■

9. $2H(\alpha, \beta, \gamma) \leqslant H(\alpha, \beta) + H(\alpha, \gamma) + H(\beta, \gamma)$.

   □
   Заметим, что $H(\alpha, \beta, \gamma) = H(\alpha) + H(\beta, \gamma \mid \alpha) = H(\alpha) + H(\beta \mid \alpha) + H(\gamma \mid \alpha, \beta)$.
   Теперь распишем аналогичное для правой части и воспользуемся $H(\alpha) \geqslant H(\alpha \mid \beta)$:

$$\begin{aligned}
H(\alpha, \beta) &= H(\alpha) + H(\beta \mid \alpha) \\
H(\alpha, \gamma) &= H(\alpha) + H(\gamma \mid \alpha) \geqslant H(\alpha) + H(\gamma \mid \alpha, \beta) \\
H(\beta, \gamma) &= H(\beta) + H(\gamma \mid \beta) \geqslant H(\beta \mid \alpha) + H(\gamma \mid \alpha, \beta)
\end{aligned}$$

   Теперь, если сложить все три формулы, то получим то, что надо. ■

2.2 Взаимная информация

Определение 4

Взаимная информация между случайными величинами α и β:

$$I(\alpha : \beta) \coloneqq H(\alpha) - H(\alpha \mid \beta).$$

Взаимная информация в α ***и*** β ***при условии*** γ :

$$I(\alpha : \beta \mid \gamma) \coloneqq H(\alpha \mid \gamma) - H(\alpha \mid \beta, \gamma).$$

Свойства.

1. $I(\alpha : \beta) = I(\beta : \alpha)$.

   □ $I(\alpha : \beta) = H(\alpha) - H(\alpha \mid \beta) = H(\alpha) - H(\alpha, \beta) + H(\beta) = H(\beta) - H(\beta \mid \alpha) = I(\beta : \alpha)$ ■

2. α и β независимы тогда и только тогда, когда $I(\alpha : \beta) = 0$.

   □ $I(\alpha : \beta) = 0 \iff H(\alpha) = H(\alpha \mid \beta) \iff$ независимы α и β. ■

3. $I(f(\alpha) : \beta) \leqslant I(\alpha : \beta)$ для любой функции f.

□ Сначала распишем левую часть:

$$\begin{aligned}
I(\alpha:\beta) &= H(\alpha) - H(\alpha \mid \beta) \\
&= \sum_{i,j} p_{i,j} \log \frac{1}{p_i} - \sum_{i,j} p_{i,j} \log \frac{p_j}{p_{i,j}} && \text{(расписали энтропию через вероятности)} \\
&= \sum_{i,j} p_{i,j} \log \frac{p_{i,j}}{p_i p_j} \\
&= \sum_{k,j} \left(\sum_{i \in f^{-1}(k)} p_{i,j} \log \frac{p_{i,j}}{p_i p_j} \right) && \text{(сгруппировали с помощью прообразов)}
\end{aligned}$$

Аналогично правую часть:

$$\begin{aligned}
I(f(\alpha):\beta) &= \sum_{k,j} \Pr[f(\alpha)=k, \beta=j] \log \frac{\Pr[f(\alpha)=k,\beta=j]}{\Pr[f(\alpha)=k]\Pr[\beta=j]} \\
&= \sum_{k,j} \left(\sum_{i \in f^{-1}(k)} p_{i,j} \right) \log \frac{\left(\sum_{i\in f^{-1}(k)} p_{i,j}\right)}{\left(\sum_{i\in f^{-1}(k)} p_i\right) p_j} \\
&= \sum_{kj} P_{kj} \log \frac{P_{kj}}{P_k p_j}
\end{aligned}$$

Где $P_k := \Pr[f(\alpha)=k] = \sum_{i\in f^{-1}(k)} p_i$, аналогично $P_{kj} := \Pr[f(\alpha)=k, \beta=j]$.
Теперь для каждой пары k, j покажем неравенство отдельно

$$\begin{aligned}
&-\sum_{i\in f^{-1}(k)} p_{i,j} \log \frac{p_{i,j}}{p_i p_j} = \sum_{i\in f^{-1}(k)} p_{i,j} \log \frac{p_i p_j}{p_{i,j}} \\
&= P_{kj} \sum_{i\in f^{-1}(k)} \frac{p_{i,j}}{P_{kj}} \log \frac{p_i p_j}{p_{i,j}} && \text{(поделили и домножили, чтобы сумма коэффициентов была 1)} \\
&\leqslant P_{kj} \log \left(\sum_{i\in f^{-1}(k)} \frac{p_i p_j}{P_{kj}} \right) && \text{(по неравенству Йенсена)} \\
&= P_{kj} \log \frac{P_k p_j}{P_{kj}} = -P_{kj} \log \frac{P_{kj}}{P_k p_j}
\end{aligned}$$

■

4. Следующие формулы для взаимной информации эквивалентны

$$\begin{aligned}
I(\alpha:\beta) &= H(\alpha) - H(\alpha \mid \beta) \\
&= H(\beta) - H(\beta \mid \alpha) \\
&= H(\alpha) + H(\beta) - H(\alpha,\beta) \\
&= H(\alpha,\beta) - H(\alpha \mid \beta) - H(\beta \mid \alpha)
\end{aligned}$$

2.3 Применение энтропии

2.3.1 Взвешивания монеток

Попробуем решить задачу с монетками. Мы взвешиваем 14 монеток и хотим найти фальшивую за три взвешивания, причем неизвестен относительный вес. Всего будет $14 \cdot 2$ исходов (одна из 14 монет фальшивая и относительный вес).

Предположим, что есть стратегия, позволяющая решить задачу.

Построим граф, соответствующий взвешиваниям: каждое имеет три исхода, всего три взвешивания. Получается дерево с 27 листьями, где во всех листьях, кроме одного, записана пара: номер фальшивой монетки и ее относительный вес. В единственной ветке, где все взвешивания давали равенство, будет записан только номер фальшивой монетки (теперь количество листьев совпадает с нужным количеством исходов[3]). Построим равномерное распределение на этих исходах.

При равномерном распределении энтропия $\log 27$.

Если стратегия верная, то

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

Так как $H(q_i) \leqslant \log 3$, для всех i должно быть $H(q_i) = \log 3$.

Чтобы было так, мы должны в каждый ход равновероятно получать все три ответа.

Пусть мы первым ходом взвешиваем кучки из k монет[4]. Вероятность того, что левая кучка будет легче в результате взвешивания $\frac{2k}{27}$ (так как у нас либо в левой кучке фальшивая, и она легче, либо в правой кучке фальшивая, и она тяжелее). Тогда $\frac{2k}{27} = \frac{1}{3}$. k получилось нецелым. Противоречие.

2.3.2 Оценка на биномиальные коэффициенты

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

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

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

Пояснение к последнему переходу. $H(\frac{k}{n}) = \frac{k}{n} \log \frac{n}{k} + (1 - \frac{k}{n}) \log \frac{1}{1-\frac{k}{n}}$. С другой стороны $H(X_1) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}$, где p — вероятность того, что первый элемент попал в множество. Заметим, что функция $f(x) = x \log \frac{1}{x} + (1-x) \log \frac{1}{1-x}$ возрастает на $(0, \frac{1}{2})$. То есть нужно лишь показать, что $p \leqslant \frac{k}{n}$. Для этого посчитаем матожидание числа элементов, которые вошли в выбранное множество.

$$k \geqslant \mathbb{E}(X) = \mathbb{E}\left(\sum X_i\right) = \sum \mathbb{E}(X_i) = \sum p_i$$

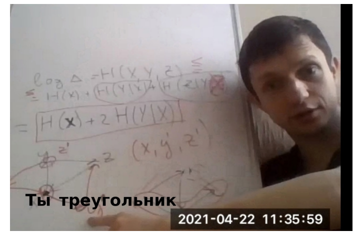
Но каждый элемент войдет в множество равновероятно, тогда $p \leqslant \frac{k}{n}$.

[3]Далее будем считать, что исходов 27

[4]Очевидно, что если взвешивать кучки разного размера, информацию извлечь не получиться даже по Хартли, так как мы не знаем ничего про относительный вес

Лекция 3
15 April

2.3.3 Подсчет углов в графе



Рассмотрим **ориентированный** граф без кратных ребер и петель. Назовем *треугольником* упорядоченную 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).

Замечание. Треугольником мы называем именно упорядоченную тройку вершин, которые соединены ребрами между собой. Поэтому просто треугольник даёт вклад равный 3 «упорядоченным» треугольникам.

Теорема 2.3.1. Число углов в графе всегда больше или равно числа треугольников.

$\square$ Пусть случайная величина $\alpha = (X, Y, Z)$ равна случайному треугольнику, где X, Y, Z — с.в., соответствующие первой, второй и третьей вершине треугольника в равномерном распределении на треугольниках.

Так как распределение количества треугольников равномерно,

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

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 случайный треугольник с x и возьмем 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 какое-то распределение на углах с такой же энтропией, следовательно, углов не меньше, чем треугольников. $\blacksquare$

Chapter 3

Теория кодирования

Определение 5: Код

Будем называть ***кодом*** функцию $C\colon \Sigma = \{a_1, \dots, a_n\} \to \{0,1\}^*$, сопоставляющую буквам некоторого алфавита кодовые слова.

Если любое сообщение, полученное применением кода C, однозначно декодируется, то такой код будем называть ***однозначно декодируемым***.

3.1 Префиксные коды

Определение 6

Код называется ***префиксным***, если никакое кодовое слово не является префиксом другого кодового слова.

Замечание. Очевидно, из этого следует однозначная декодируемость.

Теорема 3.1.1. Пусть набор целых чисел $l_1, \dots, l_n$ удовлетворяет неравенству

$$\sum_{i=1}^{n} 2^{-l_i} \leqslant 1.$$

Тогда существует префиксный код с кодовыми словами $c_1, \dots, c_n$, где $|c_i| \leqslant l_i$.

□ Индукция по n.

- База: $n = 1$. Очевидно.
- Переход: $n \to n+1$. Пусть нам даны числа $l_1, \dots, l_{n+1}$.
  - Если среди них есть два одинаковых, $l_1 = l_2$, заменим их на одно $l = l_1 - 1$, чтобы сохранилась сумма обратных степеней двойки. Теперь можем применить предположение для n и найти префиксный код со словами $c, c_3, \dots, c_{n+1}$. Заменим слово c на $\overline{c0}$ и $\overline{c1}$, которые будут соответствовать l_1 и l_2.
  - Если все различны, то можем выбрать слова длины $l_1, l_2, \dots, l_n$ из слов вида $0, 10, 110, \dots$. Этот код будет префиксным, так как у всех разное число единиц перед нулем.

■

Теорема 3.1.2 (Неравенство Крафта-Макмиллана)**.** Для любого однозначно декодируемого кода с n кодовыми словами $c_1, \ldots, c_n$ выполнено неравенство

$$\sum_{i=1}^{n} 2^{-|c_i|} \leqslant 1.$$

□ Построим многочлен для всех слов длины L.

Для этого воспользуемся следующей идеей.

Строка длины L переходит в конкатенацию кодовых слов для каждого символа исходной строки. Давайте в кодовых словах заменим 0 на x, а 1 на y.

Тогда всем кодовым словам соответствуют некоторые мономы $p_i(x, y)$ (например, $c_i = 010$ соответствует моном xyx).

Считаем, что x и y не коммутируют, чтобы удобнее было различать слова xyx, соответствующие 010 и xxy, соответствующие 001.

Будем говорить, что слову c_i *соответствует* $p_i(x, y)$. Тогда строка длины L есть произведение L таких мономов. Тогда коды всех слов длины L можно представить в виде многочлена $P(x, y)$, в котором операция сложения будет разделять разные коды:

$$P(x, y) = \left(\sum_i p_i(x, y)\right)^L = \sum_{j=L}^{L \cdot \max|c_i|} M_j(x, y).$$

В $M_j(x, y)$ сгруппированы в сумму все мономы степени j, получающиеся при раскрытии скобок (то есть все различные слова длины j, получаемые из кодирования слов длины L).

Так как код однозначно декодируемый, каждое слово является образом не более чем одной исходной строки. Следовательно, в $M_j(x, y)$ не более 2^j мономов.

Посчитаем $P(\frac{1}{2}, \frac{1}{2})$:

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

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 L она обгонит линейную. Противоречие. ■

Теорема 3.1.3. Любой однозначно декодируемый код можно переделать в префиксный с сохранением длин кодовых слов.

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

По теореме **??** (неравенство Крафта-Макмиллана) для исходного декодируемого кода выполнено неравенство $\sum 2^{-|c_i|} \leqslant 1$.

Тогда по еще одной теореме **??** существует префиксный код с кодовыми словами такой же длины (если какие-то слова стали меньше, можно просто дополнить чем-то в конце, на префиксы это не повлияет). ■

Теорема 3.1.4 (Шеннон). Для любого распределения p[a] и однозначно декодируемого кода выполнено неравенство:

$$\sum_i p_i|c_i| \geqslant H(p).$$

[a]То есть каждому символу из конечного алфавита сопоставлена его встречаемость

□

$$\begin{aligned} H(p) - \sum_i p_i|c_i| &= \sum_i p_i \log \frac{2^{-|c_i|}}{p_i} \\ &\leqslant \log \sum_i p_i \cdot \frac{2^{-|c_i|}}{p_i} && \text{(Неравенство Йенсена)} \\ &\leqslant 0 && \text{(Неравенство Крафта-Макмиллана)} \end{aligned}$$

■

Теорема 3.1.5 (Шеннон). Для любого распределения p существует такой префиксный код, что[a]

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

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

□ Угадаем длины кодов, чтобы выполнялось неравенство Крафта-Макмиллана. Пусть $|c_i| = \lceil \log \frac{1}{p_i} \rceil$, тогда неравенство из условия выполнено, так как $p_i|c_i| \leqslant p_i(\log \frac{1}{p_i} + 1) = p_i \log \frac{1}{p_i} + p_i$ и $\sum p_i = 1$. Кроме этого:

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

Теперь по теореме **??** такой код действительно существует и удовлетворяет условию теоремы. ■

3.2 Примеры эффективных кодов

3.2.1 Код Ше́ннона-Фано́

Отсортируем вероятности по убыванию $p_1 \geqslant p_2 \geqslant \ldots \geqslant p_n$. Затем «уложим» их в отрезок $[0, 1]$, получая при этом такие точки:

$$0 \leqslant p_1 < p_1 + p_2 < \ldots < p_1 + p_2 + \ldots + p_n \leqslant 1.$$

Разделим отрезок $[0, 1]$ 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 (никогда не выполняется), если p_n, и, наконец, иначе выбираем любое значение.

Далее рекурсивно продолжаем процесс на группе нулей и на группе единиц.

Когда остался один кусок, останавливаемся.

Теорема 3.2.1.

$$\sum_{i=0}^{n} p_i \cdot |c_i| \leqslant H(p) + \mathcal{O}(1), \quad n \to \infty,\ \mathcal{O}(1) \approx 3 \text{ или } 5.$$

Без доказательства. Дано как «упражнение со звездочкой».

3.2.2 Код Хаффмана

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

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

Теорема 3.2.2. Для кода Хаффмана выполнено неравенство:

$$\sum_{i=1}^{n} p_i|c_i| \leqslant H(p) + 1,$$

причем для любого другого однозначно декодируемого кода c_i' верно:

$$\sum_{i=1}^{n} p_i|c_i| \leqslant \sum_{i=1}^{n} p_i|c_i'|.$$

□ Достаточно доказать второе, а потом сравнить с кодом, который нам дает теорема Шеннона **??**, и получить нужное неравенство.

Рассмотрим набор $c_1', \ldots c_n'$. Будем считать, что этот код префиксный, так как мы научились любой декодируемый код переделывать в префиксный с той же длиной кодовых слов.

- **Шаг 1.** Возьмем два наименее вероятных кодовых слова c_{n-1}' и c_n'.

  Заметим, что можно поменять их с символами максимальной длины c_i' и c_j', при этом средняя длина кода не увеличится.

  Значит, перестроили так, что код не ухудшился и c_n', c_{n-1}' соответствуют символам с самой маленькой вероятностью и самой большой длиной кодовых слов.

- **Шаг 2.** Изучим коды c_{n-1}' и c_n'. Пусть они не имеют вид $\overline{v0}$ и $\overline{v1}$. И пусть $|c_{n-1}'| \leqslant |c_n'|$.

  Посмотрим на c_{n-1}': не умаляя общности он будет заканчиваться на 0 ($c_{n-1}' = \overline{s0}$).

  Заменим c_n' на $\overline{s1}$. *Что при этом могло сломаться?*

  Так как наш код префиксный, нам нужно проверить, что он префиксным и остался.

  Заметим, что так как $c_{n-1}' = \overline{s0}$, префиксов s нет среди кодовых слов.

  Единственная проблема тогда: среди кодовых слов может быть само $\overline{s1}$.

  Тогда поступим следующим образом. Заменим c_n' на слово длины $|s| + 1$, а затем поменяем его местами с кодовым словом, равным $\overline{s1}$.

  Почему мы можем найти новое слово подходящей длины?

  Например, обрежем c_n' до длины $|s| + 1$ и обозначим результат как $\tilde{c}$. Получили то, что требовалось:

  - У $\tilde{c}$ нет префиксов среди остальных кодовых слов, так как их не было у c_n'.
  - $\tilde{c}$ само не является префиксом другого слова. Пусть не так и $\exists d$ такое, что $\tilde{c}$ его префикс. Тогда возможны 2 варианта:
    1. $|d| \geqslant |s| + 2 = \tilde{c} + 1 \geqslant |c_{n-1}'| + 1$, что невозможно исходя из максимальности длин c_{n-1}' и c_n';
    2. $d = \tilde{c}$, что тоже невозможно, так как иначе d - префикс c_n'.

  В итоге перестроили так, что средняя длина кода не увеличилась и $c_{n-1} = \overline{v0}$, а $c_n = \overline{v1}$.

- **Шаг 3.** Заменим c_{n-1} и c_n на один символ с кодовым словом v. И применим предположение, что код Хаффмана оптимален для алфавита из $n-1$ символа.

  Тогда, заменив v обратно на c_{n-1} и c_n получим, что код Хаффмана оптимален и для n..

■

3.2.3 Арифметическое кодирование

Уложим вероятности p_i (частоты) аналогично на отрезок, при этом не обязательно в порядке убывания.

Назовем ***стандартным*** полуинтервал с таким бинарным представлением: $[\overline{0.v0}, \overline{0.v1})$.

Найдем максимальный по длине стандартный интервал в отрезке $[p_1 + \dots + p_{i-1}, p_1 + \dots + p_i]$. Пусть это $(0.v_i0, 0.v_i1)$.

Сопоставим i-ой букве код v_i0.

Заметим, что такой код будет префиксным, так как отрезки не пересекаются, а, чтобы v_i было префиксом v_j, интервал $(0.v_j0, 0.v_j1)$ должен быть вложен в $(0.v_i0, 0.v_i1)$.

Лемма 2. В отрезке $[a, b]$ длина наибольшего стандартного интервала не меньше $\frac{b-a}{8}$.

□ Пусть $2^{-k-1} < \frac{b-a}{8} < 2^{-k}$. Рассмотрим все стандартные интервалы длины 2^{-k}. Заметим, что соседние интервалы находятся на расстоянии 2^{-k} (от конца левого до начала правого).

Предположим, что ни один стандартный интервал длины 2^{-k} не попал полностью в $[a, b]$. Тогда длина $[a, b]$ меньше суммы длин двух стандартных и расстояния между ними, то есть

$$2^{-k} \cdot 2 + 2^{-k} < 2^{-k+2} \implies 2^{-k+2} > b - a \implies 2^{-k-1} > \frac{b-a}{8}$$

Противоречие. Следовательно, какой-то отрезок попал внутрь $[a, b]$. ■

Теорема 3.2.3. Для арифметического кода выполнено неравенство

$$\sum_i p_i |v_i| \leqslant H(p) + 2.$$

□ Из леммы **??** следует, что если $|v_i| = k$, то

$$0.v_i1 - 0.v_i0 = 2^{-k-1} \geqslant \frac{p_i}{8}.$$

Поэтому $k + 1 \leqslant \log \frac{8}{p_i}$, следовательно, $|v_i| = k \leqslant \log \frac{1}{p_i} + 2$ и

$$\sum_i p_i |v_i| \leqslant H(p) + 2(p_1 + \ldots + p_n) \leqslant H(p) + 2.$$

■

Лекция 4
22 April

3.3 Кодирование с ошибками

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 $p_1, p_2, \dots p_k$. [1]

Откажемся от условия однозначного декодирования, но будем требовать

$$\varepsilon_n \underset{n\to\infty}{\to} 0$$ [2]

где $|x| = n, \varepsilon_n \coloneqq \Pr[D(E(x)) \neq x]$

[1] считаем, что слово состоит из независимых букв

[2] Если сделать равенство нулю, то особого сжатия не будет

Теорема 3.3.1 (Шеннон).

1. Если $L_n > \lceil hn \rceil$, где $h > H(p)$, то кодирование есть, то есть существуют такие функции E и D, что $\varepsilon_n \to 0$.
2. Если $L_n < \lceil hn \rceil$, где $h < H(p)$, то $\varepsilon_n \to 1$ для любых E и D.

☐ Зафиксируем $\delta = n^{-0.02}$.

Определение 7

Будем называть слово w δ-***типичным***, если

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

- Докажем, что можем закодировать такие типичные слова.
  - Пусть X_{ij} — характеристическая функция того, что в слове на позиции j находится буква i. Для случайной величины $X_i \coloneqq \sum\limits_j X_{ij}$ применим неравенство Чебышева для вероятности того, что условие типичности не выполняется для конкретной буквы:

$$\Pr[|X_i - np_i| \geqslant \delta n] \leqslant \frac{\mathbb{D}[X_i]}{(\delta n)^2} = \frac{np_i(1-p_i)}{(\delta n)^2} = \mathcal{O}\left(\frac{1}{\delta^2 n}\right),$$

    так как $\mathbb{E}X_i = np_i$.
    Мы расписали вероятность, что какой-то символ нарушает условие δ-типичности.

$$\begin{aligned}\Pr[\text{слово не } \delta - \text{типично}] &= \Pr[\exists i \text{ такой, что на нём нарушается неравенство}]\\ &= \Pr\left[\exists i : \left|\frac{X_i}{n} - p_i\right| > \delta\right]\\ &\leqslant \sum_i \Pr\left[\left|\frac{X_i}{n} - p_i\right| > \delta\right]\\ &= \mathcal{O}\left(\frac{k}{\delta^2 n}\right) = \mathcal{O}\left(\frac{1}{\delta^2 n}\right)\end{aligned}$$

    То есть, раз букв константное количество, вероятность нетипичности все равно останется очень маленькой и будет стремиться к нулю.
  - Теперь докажем, что типичных слов не очень много. Число слов, где буквы встречаются в количествах $n_1, \ldots, n_k$ равно

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

    Обозначим $\delta_i \coloneqq \frac{n_i}{n} - p_i$.

$$\begin{aligned}\log N_{n_1,\ldots n_k} &= && \left(\text{так как } n! = \mathrm{poly}(n)\left(\tfrac{n}{e}\right)^n\right)\\ &= \log\left(\left(\frac{n}{n_1}\right)^{n_1} \cdot \left(\frac{n}{n_2}\right)^{n_2} \cdot \ldots \cdot \left(\frac{n}{n_k}\right)^{n_k}\right) + \mathcal{O}(\log n) = \\ &= \sum_i n_i \log \frac{n}{n_i} + \mathcal{O}(\log n) = \\ &= n \sum_i (p_i + \delta_i) \cdot \log \frac{1}{p_i + \delta_i} + \mathcal{O}(\log n) && (n_i \text{ по определению})\end{aligned}$$

  - Теперь оценим число типичных слов.

Для этого можно посчитать для каждого распределения букв, сколько слов с таким распределением будут типичными.

Число способов разбить n на k слагаемых грубо оценивается сверху как n^k[3]. А число слов в соответствующем разбиении можно оценить максимумом по всем δ_i, таким что $|\delta_i| \leqslant \delta$ (так как слово δ-типичное).

$$\begin{aligned}\log\Big(\#(\delta\text{-типичных слов})\Big) &\leqslant \log\left(n^k \cdot \max_{\delta_i} N_{n_1,\ldots n_k}\right) \leqslant \\ &\leqslant n \cdot \max_{\delta_i} \sum_i (p_i + \delta_i) \cdot \log \frac{1}{p_i + \delta_i} + \mathcal{O}(\log n) \leqslant \\ &\leqslant n \cdot \left(\sum_i p_i \log \frac{1}{p_i} + \delta \cdot \log \frac{1}{p_i}\right) + \mathcal{O}(\log n) = \\ &= nH(p) + \mathcal{O}(\delta \cdot n)\end{aligned}$$

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.

$$\frac{\log\left(\#\delta\text{-типичных слов}\right)}{L_n} = \frac{nH(p) + \mathcal{O}(\delta n)}{L_n} \leqslant \frac{nH(p) + \mathcal{O}(n^{0.98})}{hn} = \frac{H(p)}{h} + \mathcal{O}(n^{-0.02})$$

Так как $H(p) < h$ и n растет быстрее $\mathcal{O}(\delta n) = \mathcal{O}(n^{0.98})$, то с какого то момента выражение меньше единицы

- Во второй части докажем, что вероятность ошибки декодера на δ-типичных словах равна 1. Понятно, что этого будет достаточно.

  Покажем, что вероятность того, что нам на вход придет конкретное δ-типичное слово очень мала.

  Пусть $L_n \leqslant hn$. Посмотрим на любое распределение кодовых слов. Покажем, что вероятность по нашему определению для δ-типичных слов больше, чем $\frac{1}{2^{L_n}}$.

$$\begin{aligned}\Pr[w = x] &= p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_k^{n_k} = && (x \text{ – конкретное слово}) \\ &= 2^{\log \Pi_i p_i^{n_i}} = \\ &= 2^{-\sum n_i \log \frac{1}{p_i}} = \\ &= 2^{-n\sum_i (p_i + \delta_i) \log \frac{1}{p_i}} = \\ &= 2^{-H(p)n + \mathcal{O}(\delta n)}\end{aligned}$$

  С какой вероятностью декодер ответит правильно?

$$\begin{aligned}\Pr[D(E(w)) = w] &= \sum_x \Pr[D(x) = w, E(w) = x] \\ &\leqslant \sum_x \Pr[D(x) = w] \leqslant && (D(x) \text{ — какое-то фиксированное слово}) \\ &\leqslant 2^{L_n} \cdot \max_{t \text{ – слово}} \Pr[w = t] \leqslant \\ &\leqslant 2^{L_n - H(p)n + \mathcal{O}(\delta n)} \to 0\end{aligned}$$

  Предельный переход верен, потому что $L_n - H(p)n + O(\delta n) \leqslant (h - H(p))n + O(\delta n) \to -\infty$, так как $h - H(p) < 0$ константа.

■

[3] Выбираем распределение n букв по k возможным значениям.

Chapter 4

Коммуникационная сложность

Лекция 5
29 April

Пусть у нас есть два игрока: Алиса и Боб. Они могут отправлять друг другу сообщения и хотят посчитать функцию (или отношение) $f\colon X \times Y \to Z$. Будем говорить, что они *решают коммуникационную задачу* для функции f, если:

1. множества X, Y, Z и функция f известны обоим игрокам,
2. Алиса знает некоторое $x \in X$,
3. Боб знает некоторое $y \in Y$,
4. Алиса и Боб стремятся вычислить $f(x, y)$.

4.1 Детерминированная модель

Определение 8: Коммуникационный протокол

Коммуникационный протокол для функции $f\colon X \times Y \to Z$ — корневое двоичное дерево, которое описывает совместное вычисление Алисой и Бобом функции f. В этом дереве:

- каждая внутренняя вершина v помечена меткой a или b, означающей очередь хода Алисы или Боба соответственно;
- для каждой вершины v, помеченной a, определена функция $g_v\colon X \to \{0, 1\}$, которая задает бит, который Алиса отправит Бобу, если вычисление будет находится в v; аналогично для каждой вершины v с пометкой b определена функция $h_v\colon Y \to \{0, 1\}$ для сообщений Боба;
- у каждой внутренней вершины есть два потомка, ребро к первому помечено 0, ко второму — 1;
- каждый лист помечен значением из множества Z.

Каждая пара входов (x, y) определяют путь от корня до листа в этом дереве. Будем говорить, что коммуникационный протокол ***вычисляет*** функцию f, если для всех пар $(x, y) \in X \times Y$ этот путь заканчивается в листе с пометкой $f(x, y)$.

Коммуникационной сложностью функции f называется наименьшая глубина протокола, вычисляющего f. Обозначается $D(f)$ или $D^{cc}(f)$.

Каждой функции можем сопоставить матрицу $X \times Y$, где в клетке (x_i, y_j) стоит значение $f(x_i, y_j)$.

4.1.1 Нижние оценки для детерминированного случая

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

Рассмотрим такое подмножество R_v множества $X \times Y$, что $(x, y) \in R_v \iff$ протокол приходит в v.

Лемма 3. $R_v = X_v \times Y_v$ — комбинаторный прямоугольник.

Рассмотрим **два** доказательства:

□ 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 $b(y) = b(y')$, где a и b - ответы Алисы и Боба.

А из этого следует, что это комбинаторный прямоугольник. ■

□ Посмотрим на множество $X \times Y$ как на таблицу.

Пусть Алиса перешла по какому-то ребру. Вся таблица разделилась на две части по горизонтали: какие-то строки соответствуют $x \in X$, для которых Алиса отсылает 0, а какие-то для 1.

Если потом ход делает Боб, то он делит все текущие прямоугольники на две части по вертикали.

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. ■

Давайте продолжим смотреть на матрицу. Зафиксируем какой-то $O_1 \in Z$. Тогда есть входы (x, y) такие, что протокол ведет их к O_1. По предыдущей лемме множество всех таких входных данных образует комбинаторный прямоугольник. Тогда, если $Z = \{O_1, \ldots, O_{|Z|}\}$ и для каждого O_i смотрим на соответствующий комбинаторный прямоугольник мы получаем разбиение исходной матрицы.

Пусть $Z = \{0, 1\}$. Рассмотрим величины $\chi_0(f)$ и $\chi_1(f)$, первая равна минимальному числу непересекающихся комбинаторных прямоугольников, которыми можно покрыть все нули в таблице, в вторая — все единицы.

Тогда листьев в двоичном дереве протокола будет хотя бы $\chi_0(f) + \chi_1(f)$. Следовательно, $D(f) \geqslant \log(\chi_0(f) + \chi_1(f))$.

Но эта оценка не всегда точна. Можно рассмотреть следующее разбиение на картинке **??**.

Здесь $\chi_0(f) + \chi_1(f) = 4 + 1 = 5$. Заметим, что для такого разбиения не существует дерева протокола.

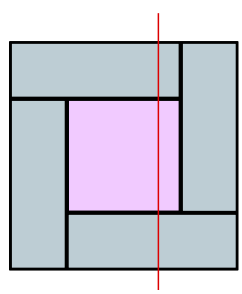


Figure 4.1: Пример неточной оценки

Посмотрим на первое действие игроков. Прямоугольник должен разделиться на две части, но любой разрез вдоль сторон разрежет один из внутренних прямоугольников.

Поэтому любой протокол не будет соответствовать этому разбиению. А тогда листьев будет больше пяти.

Теорема 4.1.1 (G,PW 16)**.** Пусть χ — минимальное число прямоугольников в разбиении. Существует f для которой
$$D(f) \geqslant \log^{2-\varepsilon} \chi(M_f).$$
Без доказательства.

4.2 Методы оценки коммуникационной сложности

4.2.1 Метод ранга

Пусть M_f — таблица некоторой функции f со значениями $\{0,1\}$.

Обозначим за $x_1, \dots x_n$ листья дерева коммуникационного протокола для функции f, а $R_{x_1}, \dots R_{x_n}$ соответствующие им прямоугольники в M_f. $\mathrm{rk}(M_f)$ — ранг матрицы M_f.

$$\mathrm{rk}(M_f) \leqslant \sum_{i=1}^{n} \mathrm{rk}\, R_{x_i} \leqslant \sum_{i=1}^{n} 1 = \#\text{ одноцветных прямоугольников в разбиении.}$$

Тогда коммуникационная сложность не меньше $\log \mathrm{rk}(M_f)$.

Этот метод называется ***методом ранга***.

Лемма 4. Для любой функции $f\colon \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^k$ коммуникационная сложность не больше $\min\{n+k, n+m, m+k\}$.

□ Пусть Алиса и Боб хотят вычислить значение $f(x,y)$, причем x знает только Алиса, а y — только Боб.

- Для $n+m$ они просто отправляют друг другу свои числа и локально вычисляют f.
- Для $n+k$ и $m+k$ достаточно просто отправить свое число второму, он посчитает значение функции и отправит обратно.

■

Оценка для EQ Определим функцию $\mathrm{EQ}(x,y) = (x = y)\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

Для функции EQ матрица M_{EQ} будет диагональной $2^n \times 2^n$.

Поэтому ее ранг равен 2^n, а тогда коммуникационная сложность $D(\mathrm{EQ}) \geqslant \log 2^n + 1 = n + 1$. [1]

С другой стороны, она не больше $n+1$, так как за $n+1$ вопрос можно просто сравнить все биты (n) и отослать результат обратно (1).

Оценка для GT Определим функцию $\mathrm{GT}(x,y) = (x \geqslant y)\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

Докажем, что ее коммуникационная сложность $D(\mathrm{GT}) = n+1$.

Матрица будет треугольной, поэтому $\mathrm{rk}\, M_{\mathrm{GT}} = 2^n$. А тогда $D(\mathrm{GT}) \geqslant \log 2^n + 1 = n + 1$.

По лемме **??** $D(\mathrm{GT}) \leqslant n + 1$, поэтому она в точности $n+1$.

4.2.2 Fooling Set

Рассмотрим коммуникационную матрицу некоторой функции f. Пусть мы хотим выбрать некоторое множество клеток

$$S = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\},$$

такое что каждая пара точек не лежит в одном прямоугольнике.

Чтобы все были в разных прямоугольниках, для всех $i \neq j \in [n]$ либо клетка (x_i, y_j), либо (x_j, y_i) была другого цвета[2].

Из условия на разные прямоугольники следует, что в дереве не менее n листьев, а поэтому высота хотя бы $\log n$.

[1] Дополнительная единица возникает из-за того, что нам нужен еще хотя бы один лист для нуля.

[2] Цвет = 0 или 1

Оценка для EQ Применим этот метод к EQ. Берем в качестве точек главную диагональ из единиц. Никакие из них не находятся в одном одноцветном прямоугольнике.

Следовательно, высота дерева не меньше n.

Плюс, так как нужен хотя бы один лист для нуля, n не хватит, следовательно, $D(\mathrm{EQ}) \geqslant n + 1$.

Теорема 4.2.1. Если существует **Fooling set** размера s, то $\mathrm{rk}_R\, s \geqslant \sqrt{s} - o(1)$. Без доказательства.

4.3 Балансировка протоколов

Теорема 4.3.1. Пусть для функции f существует коммуникационный протокол P с l листьями. Тогда $D(f) \leqslant 3 \log l$.[a]

[a]Вроде на лекциях этого не было, зато есть в билетах

Лемма 5. В двоичном дереве с l листьями можно найти внутреннюю вершину u, которая разбивает граф на три части, каждая из которых содержит не более $\frac{l}{2}$ листьев.

Обозначим через l_u поддерево с корнем во внутренней вершине u.

Найдем самую низкую вершину u, для которой $|l_u| > \frac{l}{2}$. Пусть ее дети v и w, причем $l_v \leqslant \frac{l}{2}$ и $l_w \leqslant \frac{l}{2}$ (если бы было не так, то могли бы взять ребенка у которого нарушается неравенство).

При этом количество листьев не из поддерева u будет меньше $l - l_u < \frac{l}{2}$.

Следовательно, u подходит.

Замечание. Такая вершина называется *центроид*.

Интуиция: Хотим перебалансировать дерево так, чтобы прыгать только по центроидам, в следствие чего глубина нового протокола уменьшиться (центроидная декомпозиция).

Опишем протокол P_0 для вычисления f.

- Зафиксируем вершину u, которая делит исходный протокол на три части, в каждой из которых не более $\frac{l}{2}$ листьев.
- Пусть вершина u соответствует ходу Алисы. Сначала проверим, что и Алиса, и Боб доходят до этой вершины в исходном протоколе P: отправляем по одному биту (сначала Алиса, потом Боб обратно отсылает результат).
  - Если и Боб, и Алиса могут добраться (по лемме **??**), то Алиса говорит Бобу, в какое поддерево она бы пошла с ее информацией. Тем самым мы опускаемся в одно из поддеревьев, листьев в котором не больше $\frac{l}{2}$.
  - Если нет, то нам точно не подходят оба поддерева u (в них нет листа ответа для (x, y)), а поэтому нужно подняться в третье поддерево (к родителю u) — для этого Алиса может ничего не отправлять, так как Боб уже знает, что спуска вниз не будет.

  Таким образом, обменявшись $\leqslant$ тремя битами, Алиса и Боб уменьшают количество листьев вдвое. Теперь мы можем не рассматривать недостижимые вершины и составить рекурсивно новый протокол P_1, где листьев $l_1 \leqslant \frac{l}{2}$.

  В итоге на один «шаг» нужно не более трех битов, при этом количество листьев уменьшается в два раза, следовательно, сложность такого протокола будет не более $3 \log l$.

4.4 Сложная функция. Теорема Шеннона

Определение 9

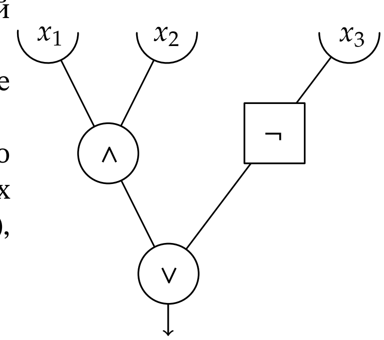
Формульная (схемная) сложность $L(f)$ формулы f — минимальное возможное число вершин двоичного дерева, вычисляющего f.

Теорема 4.4.1 (Шеннон). Существует $f\colon \{0,1\}^n \to \{0,1\}$, такая что

$$L(f) \geqslant \Omega\left(\frac{2^n}{n}\right).$$

□

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



Посчитаем число схем. Каждую схему можно представить в виде ациклического графа и того, что записано в его узлах.
Пусть каждая вершина (считаем, что размер схемы S, поэтому их столько же) выбирает себе двух предков. Для этого она должна хранить их номера. Так же в каждую вершину нужно записать операцию ($\wedge$ или $\vee$), а на ребре можно ставить отрицание. Чтобы это закодировать хватит

$$\underbrace{S \cdot 2(\log S + \log n)}_{\text{у вершины 2 предка: либо другие вершины, либо вход}} + \underbrace{3S}_{\text{операция плюс отрицания}}$$

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

Чтобы мы могли закодировать все функции в виде схем тогда и только тогда, когда схем не меньше количества функций:

$$2^{S\cdot 2(\log S+\log n)+3S} \geqslant 2^{2^n}$$
$$S\cdot(2\log S + 2\log n + 3) \geqslant 2^n$$

Пусть для всех схем $S < \frac{1}{4}\cdot\frac{2^n}{n}$, тогда

$$S\cdot(2\log S + 2\log n + 3) < \frac{2^n}{4n}(2n - 4 + 3) = 2^n\left(\frac{1}{2} - \frac{1}{4n}\right) \leqslant 2^n \cdot \frac{1}{2} < 2^n$$

Противоречие. Значит есть схема со сложностью $\geqslant \frac{1}{4}\cdot\frac{2^n}{n}$.

■

Теорема 4.4.2 (Существование сложной монотонной функции). Существует монотонная $f\colon \{0,1\}^n \to \{0,1\}$, такая что

$$L(f) \geqslant 2^n - o(2^n).$$

□ Оценим снизу количество монотонных функций. Рассмотрим такие функции g, что $g(u) = 0$, если в $u < \lfloor \frac{n}{2} \rfloor$ единиц, и $g(u) = 1$, если в $u > \lfloor \frac{n}{2} \rfloor$ единиц. Если же единиц ровно $\lfloor \frac{n}{2} \rfloor$, то функция может быть определена как угодно. Заметим, что всего функций такого вида будет $2^{\binom{n}{\lfloor \frac{n}{2} \rfloor}}$, так как на входах, где $\lfloor \frac{n}{2} \rfloor$ единиц, ответ может быть любым из 0 или 1, входов с $\lfloor \frac{n}{2} \rfloor$ единиц как раз $\binom{n}{\lfloor \frac{n}{2} \rfloor}$. Можем оценить $\binom{n}{\lfloor \frac{n}{2} \rfloor} \leqslant 2^n$, например, через бином Ньютона: $(1+1)^n = \sum_{k=0}^{n}\binom{n}{k} \geqslant \binom{n}{\lfloor \frac{n}{2} \rfloor}$.

В предыдущей теореме **??** показали, что всего схем сложности S может быть $2^{S\cdot 2(\log S+\log n)+3S}$. Тогда для схем, вычисляющих монотонные функции, имеем неравенство:

$$2^{S\cdot 2(\log S+\log n)+3S} \geqslant 2^{\binom{n}{\lfloor \frac{n}{2}\rfloor}}$$

$$S\cdot(2\log S+2\log n+3) \geqslant 2^n \geqslant \binom{n}{\lfloor \frac{n}{2}\rfloor}$$

$$\log S \geqslant n-\log\bigl(2\log S+2\log n+3\bigr) = n-o(n)$$

■

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

4.5 Теорема Карчмера-Вигдерсона

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

Алиса получает число $x \in f^{-1}(1)$, а Боб $y \in f^{-1}(0)$. Их цель найти любой бит, в котором x и y отличаются.

Теорема 4.5.1 (Karchmer-Wigderson, 1990). $L(f)$ — размер минимальной формулы для f, тогда и только тогда $L(f)$ — размер минимального протокола для KW_f

□

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

Тогда $f(x) = 1$, $f(y) = 0$. Тогда $g(y) = h(y) = 0$, а для Алисы хотя бы одно из двух значений единица.Тогда Алиса посылает информацию, где именно единица, то есть куда нам нужно спуститься (потому что в этом отрезке битов точно есть отличие). Далее игра продолжается по тем же правила рекурсивно. Стоит заметить, что если в вершине конъюнкция, то Боб пошлет информацию, где происходит обнуление. Так за глубину формулы мы нашли решение для KW_f той же глубины. Т.е. оптимальный размер протокола меньше или равен $L(f)$.

$\boxed{2 \Longrightarrow 1}$ Пусть у нас есть некоторый протокол для игры. Это некоторое дерево. Обозначим за $R_v \coloneqq X_v \times Y_v$ — прямоугольник входов для вершины v, из которых мы получаем v.

Мы хотим построить $f\colon \{0,1\}^n \to \{0,1\}$.

Будем подниматься снизу и строить формулу по протоколу. Обозначим за f_v построенную формулу в вершине v. Хотим получить следующие свойства для формулы: $f_v(X_v) = 1$ и $f_v(Y_v) = 0$.

Если они выполняются, то $\forall x \in X_v\colon f(x) = 1$ и $\forall y \in Y_v\colon f(y) = 0$.

В корне $f = f_r$.

- Если мы в листе l. Здесь написан некоторый ответ i. То есть $\forall x \in X_l\ \forall y \in Y_l\colon x_i \neq y_i$.
  Тогда либо $x_i = 0 \wedge y_i = 1$, либо наоборот. (Причём одинаково для всех пар x_i, y_i)
  В качестве $f_l(z)$ можем в первом случае взять $\neg z_i$, во втором z_i.
- Теперь мы находимся в вершине v с потомками a и b.
  Если ходит Алиса, то прямоугольник R_v разрезается на R_a и R_b горизонтально (если Боб, то наоборот, вертикально).
  У нас уже есть две функции f_a и f_b, построенные по предположению индукции. $\forall y \in Y_v\colon f_a(y) = f_b(y) = 0$, так как $Y_v = Y_a = Y_b$. А так как $X_v \subseteq X_a \cup X_b$, $\forall x \in X_v$ либо $f_a(x) = 1$, либо $f_b(x) = 1$.
  Поэтому нам подходит $f_v \coloneqq f_a \vee f_b$.
  Если же ходит Боб нужно будет сделать конъюнкцию.

■

4.6 Вероятностная модель

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

Каждый шаг будет зависеть от входа x и случайной строки $r \in \{0,1\}^l$.

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

Вероятность ошибки на входе (x, y) — доля случайных слов r, при которых протокол выдает неправильный ответ.

Скажем, что ***протокол отработал корректно***, если[3]

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

Определение 10

Вероятностный протокол ε ***-вычисляет*** f, если для любой пары x, y с вероятностью не менее $1-\varepsilon$ результат протокола равен $f(x,y)$ (с точки зрения обоих игроков).

Через $R_\varepsilon(f)$ обозначается минимальная высота вероятностного протокола ε-вычисляющего f.

Через R_ε^{pub} — минимальная высота в случае публичного протокола.

4.6.1 Эффективные протоколы для EQ и GT

Лемма 6. $R_\varepsilon^{pub}(\mathrm{EQ}) = \mathcal{O}(\log \varepsilon^{-1})$

□ Рассмотрим следующий протокол. На входах $x, y \in \{0,1\}^n$ Алиса и Боб выбирают случайную строку $r \in \{0,1\}^n$, затем Алиса считает скалярное произведение $\langle x, r\rangle$ над $\mathbb{F}_2$ и передает результат (один бит) Бобу, который считает свое скалярное произведение $\langle y, r\rangle$. Если биты равны, отвечает 1, иначе 0.

- Если $x = y$, то ответ точно будет одинаковый и протокол работает правильно.
- Если $x \neq y$, протокол ошибается (то есть выдает единицу, хотя должен нуль), если $\langle x-y, r\rangle = 0$. Это линейное уравнение на r. Соответственно, его решения — некоторое линейное пространство размерности $n-1$ над $\mathbb{F}_2$.

  Тогда решений будет ровно 2^{n-1}. А это половина от всех возможных r. Следовательно, вероятность ошибки $\frac{1}{2}$.

Теперь просто повторяем этот протокол k раз с независимыми битами r, причем будем выдавать 1, если получили все k единиц. За счет этого теперь вероятность ошибки станет 2^{-k}.

Если мы хотим добиться ошибки ε, достаточно повторить протокол $\log \varepsilon^{-1}$ раз. ■

Лемма 7. $R_\varepsilon^{pub}(\mathrm{GT}) = \mathcal{O}(\log n \cdot \log \frac{\log n}{\varepsilon})$

□ Чтобы сравнить $x, y \in \{0,1\}^n$, Алисе и Бобу нужно найти первый (самый старший) бит, в котором их входы отличаются, и обменяться этими битами.

Будем искать этот бит бинпоиском. Каждый раз мы делим текущий блок на две части и проводит тест на равенство левых частей (который мы умеем делать по лемме **??**). Если они оказались равны, то выбираем правые, иначе левые, и запускаемся рекурсивно.

[3] Вместо $\frac{2}{3}$ может быть любое $\varepsilon \in (\frac{1}{2}, 1)$

После каждого деления блок уменьшается не меньше, чем в два раза, следовательно, тест на равенство нужно провести $\mathcal{O}(\log n)$ раз.

Чтобы общая вероятность была меньше ε, вероятность ошибки на одной проверке должна быть меньше $\delta = \frac{\varepsilon}{\log n}$.

В итоге получается $R_\varepsilon^{pub}(\mathrm{GT}) = \mathcal{O}(\log n \cdot \log \frac{\log n}{\varepsilon})$.

4.7 Теория информации в коммуникационной сложности

4.7.1 Подсчет функции индексов

Лекция 6
6 May

Определение 11

Определим ***функцию индексирования*** следующим образом:

$$\mathrm{Ind}\colon [n] \times \{0,1\}^n \to \{0,1\}, \qquad \mathrm{Ind}(x,y) = y_x.$$

Алиса и Боб хотят посчитать эту величину, причем x у Алисы, а y у Боба.

Пусть сообщения идут только от Боба до Алисы. Несложно понять, что Бобу придется послать всю информацию.

Теперь предположим, что они хотят, чтобы Алиса посчитала Ind верно с вероятностью $\frac{1}{2} + \delta$, если x и y выбираются равномерно. Очевидно, для $\delta = 0$, просто ничего не нужно пересылать. А вот для других положительных значений все испортится.

. . . .

Пусть $M(y)$ — случайная величина, это сообщение, которое передает Боб Алисе в зависимости от своего входа. Легко получить

$$D(\pi) \geqslant \log|L| \geqslant H(M) \geqslant I(M : y)$$

Где L — множество различных сообщений, которые может отправить Боб (то есть множество различных листьев в протоколе), $D(\pi)$ — глубина протокола (то есть количество бит, которое нужно отправить).

$$\begin{aligned}
I(M : y) &= \sum_i I(M : y_i \mid y_{<i}) = && \text{(Chain rule)}\\
&= \sum_i H(y_i \mid y_{<i}) - H(y_i \mid M, y_{<i}) = && (y_i \text{ независимы})\\
&= \sum_i H(y_i) - H(y_i \mid M, y_{<i}) \geqslant && \text{(Выкинули часть условий)}\\
&\geqslant \sum_i H(y_i) - H(y_i \mid M) = &&\\
&= \sum I(M : y_i) &&
\end{aligned}$$

Покажем, что полученная сумма большая.

Зафиксируем i и распишем по определению взаимной информации:

$$\begin{aligned}
\sum_i I(M : y_i) &= \sum_i H(y_i) - H(y_i \mid M) = && (H(y_i) = 1)\\
&= \sum_i 1 - \mathbb{E}_m \left(H(y_i \mid M = m)\right) = \\
& && (x \text{ не зависит от } (M, y), \text{ значит } (y \mid M) \text{ не зависимо с } (x \mid M))\\
&= \sum_i 1 - \mathbb{E}_m \left(H(y_i \mid M = m, x = i)\right) = \\
&= \sum_i 1 - \mathbb{E}_m(H(r_m^i)) \geqslant && (\text{Неравенство Йенсена для выпуклой вверх } H(q))\\
&\geqslant \sum_i 1 - H(\mathbb{E}_m(r_m^i)) = \\
&= n - n\sum_i \frac{1}{n} H(\mathbb{E}_m(r_m^i)) \geqslant && (\text{Опять неравенство Йенсена для } H(q))\\
&\geqslant n \cdot \left(1 - H\left(\sum_i \tfrac{1}{n}\mathbb{E}_m(r_m^i)\right)\right) = n \cdot \left(1 - H\left(\mathbb{E}_i(\mathbb{E}_m(r_m^i))\right)\right)\\
&\geqslant n \cdot \left(1 - H\big(\mathbb{E}_{m,i}(r_m^i)\big)\right) \geqslant && (\text{Монотонность энтропии на } (0, \tfrac{1}{2}))\\
&\geqslant n \cdot \left(1 - H\left(\tfrac{1}{2} - \delta\right)\right) = && (\text{Ряд Тейлора для энтропии})\\
&= \Omega(\delta^2 n)
\end{aligned}$$

Здесь r_m^i — характеристическая функция ошибки $M = m$, $x = i$.

Пояснение к переходу к r_m^i: Так как у Алисы алгоритм детерминированный, то при фиксированных x, M она решает, что y_x равно конкретному значению (0 или 1) при этом, если $\Pr[y_i = 1 \mid M = m, x = i] = p$, то рассмотрим 2 случая:

1. Алиса решает, что $y_x = 1$, то она ошибется с вероятностью $1 - p$
2. Алиса решает, что $y_x = 0$, то она ошибется с вероятностью p

То есть в обоих случаях распределения совпадают, а значит энтропия одинакова.

Чтобы алгоритм был корректен, $\mathbb{E}_{i,m}(r_m^i) \leqslant \frac{1}{2} - \delta$.

Теперь $D(\pi) \geqslant \Omega(\delta^2 n)$. То есть нам нужно передать хотя бы $\Omega(\delta^2 n)$ бит. С другой стороны можем построить простой протокол с $2\delta n$ битами. Для этого Боб передаст первые $2\delta n$ бит, а затем Алиса, если нужный ей бит есть, выдаст его, иначе выдаст 0. Вероятность ошибки в таком случае ровно такая, как мы хотели: $\frac{1}{2}(1 - 2\delta)$. Эта верхняя оценка больше нашей нижней. На самом деле она улучшается.

Определение 12

Пусть μ — вероятностная мера на $X \times Y$.

$\mathrm{IC}_\mu^{ext}(\pi) := I(\pi(X, Y) : (X, Y))$ — внешнее информационное разглашение.

$\mathrm{IC}_\mu^{int}(\pi) := I(\pi(X, Y) : X \mid Y) + I(\pi(X, Y) : Y \mid X)$ – внутреннее информационное разглашение.

Интуиция:

1. Внешнее информационное разглашение — сколько информации получит внешний наблюдатель об (X, Y), зная значение протокола $\pi(X, Y)$.
2. Внутреннее информационное разглашение — сколько информации получат внутренние наблюдатели о входах друг друга, посмотрев значения протокола.

Теорема 4.7.1. $D(\pi) \geqslant \mathrm{IC}_{\mu}^{ext}(\pi) \geqslant \mathrm{IC}_{\mu}^{int}(\pi)$

□

- Первое неравенство очевидно, так как

$$I(\pi(X,Y) : (X,Y)) \leqslant H(\pi(X,Y)) \leqslant \log |L| \leqslant D(\pi(X,Y))$$

  L — листья протокола.

- Докажем вторую часть теоремы. Для этого покажем $I(\pi : x,y) \geqslant I(\pi : x \mid y) + I(\pi : y \mid x)$.

  Если оставить только одно слагаемое, слева останется $H(\pi) - H(\pi \mid x,y)$, а справа $H(\pi \mid y) - H(\pi \mid x,y)$, которое точно не больше.

  Пусть π_i – i-ый бит π, то есть то, что Алиса и Боб отправили за i-ый раунд.

$$I(\pi : x,y) = \sum_i I(\pi_i : x,y \mid \pi_{<i}) \qquad \text{(Chain rule)}$$

  Аналогично попробуем нарезать слагаемые правой части и оценим каждую часть по отдельности:

$$I(\pi : x \mid y) = \sum_i I(\pi_i : x \mid y, \pi_{<i})$$
$$I(\pi : y \mid x) = \sum_i I(\pi_i : y \mid x, \pi_{<i})$$

  В ход Боба первое слагаемое «равно нулю», так как π_i определяется y-ом. Аналогично второе «равно нулю», когда ходит Алиса. Поэтому каждый раз неравенство сохраняется. На самом деле есть ошибка в рассуждениях, т.к. $\pi_{<i}$ с.в., то при разных значениях $\pi_{<i}$ ходить может либо Боб, либо Алиса, а не всегда кто-то один при всех значениях, как мы пытались обосновать.

  Чтобы исправить скрытую фундаментальную ошибку распишем через матожидания

$$I(\pi : x,y) = \sum_i \mathbb{E}_m I(\pi_i : x,y \mid \pi_{<i} = m)$$
$$I(\pi : x \mid y) = \sum_i \mathbb{E}_m I(\pi_i : x \mid y, \pi_{<i} = m)$$
$$I(\pi : y \mid x) = \sum_i \mathbb{E}_m I(\pi_i : y \mid x, \pi_{<i} = m)$$

  Это уже корректное утверждение, так как для каждого конкретного m одно из слагаемых действительно обнуляется, значит для матожиданий неравенство есть, ну и значит для суммы тоже.

■

Теорема 4.7.2 (Храпченко). $L(XOR) \geqslant \Omega(n^2)$

□ Покажем, что для любого протокола π задачи $\mathrm{KW}_{\oplus_n}$ существует такое распределение μ, что $\mathrm{IC}_{\mu}^{int}(\pi) \geqslant 2\log n$. Отсюда будет следовать, что (пользуясь выводами в теореме **??**) $D(\pi) \geqslant \log m \geqslant \log |L| \geqslant 2\log n$, где m – кол-во вершин, $|L|$ - кол-во листьев, а значит и $L(\oplus_n) \geqslant n^2$. Пусть распределение μ будет равномерным на всех парах вида $(x, x \oplus e_i)$, где $\oplus_n(x) = 1$, то есть в x нечётное число единиц, а строка e_i имеет единицу в позиции i и нули во всех остальных. Таким образом, пары входов из распределения μ всегда будут отличаться только в одном бите. По определению,

$$\mathrm{IC}_{\mu}^{int}(\pi) = I(\pi : X \mid Y) + I(\pi : Y \mid X).$$

Рассмотрим одно из слагаемых $I(\pi : Y \mid X)$. Имеем:

$$I(\pi : Y \mid X) = I(\pi : X \oplus e_i \mid X) = H(X \oplus e_i \mid X) - \underbrace{H(X \oplus e_i \mid X, \pi)}_{=0} = H(e_i) = \log n$$

Аналогичное равенство верно и для $I(\pi : Y \mid X)$. Таким образом, $\mathrm{IC}_{\mu}^{int}(\pi) \geq 2 \log n$, что и требовалось. Верхнюю оценку на $D(\pi)$ нетрудно доказать, явно построив функцию. ∎

Chapter 5

Колмогоровская сложность

5.1 Определения

Лекция 7
20 May

Пусть F — вычислимая декодирующая функция. Определим $K_F(x) \coloneqq \min\{|p| \mid F(p) = x\}$. Это «критерий сжимаемости» функции F.

Будем считать, что $F..G$ (F не хуже G), если $\forall x\; K_F(x) \leqslant K_G(x) + c_{FG}$.

Назовем способ описания (функцию) оптимальным, если она не хуже всех остальных.

Теорема 5.1.1. Существует оптимальный способ описания.

Определение 13

Колмогоровская сложность для x — $K(x) \coloneqq K_U(x)$, где U — оптимальный способ описания.

Свойства.

1. $K(x) \leqslant |x| + c$, так как можно взять $K_{id}(x) = |x|$ I
2. $K(XX) \leqslant |x| + c$, можно взять описание $F(p) = pp$
3. Пусть Gx не более p единиц, тогда $K(x) \leqslant H(p) \cdot |x| + c$.

   Можем взять $F(p) = p$-ое слово с не более pединицами.

$$\sum_{i=0}^{n} \binom{i}{n} \leqslant 2^{H(p)n}.$$

Теорема 5.1.2. Пусть M — всюду вычислимая функция. Если $M(x) \leqslant K(x)$ и $\forall c\; \exists x\colon M(x) \geqslant c$, то M не вычислима.

□ Зафиксируем c. Найдем x_c — первое слово, где $M(x_c) \geqslant c$. Так как M вычислима всюду, определено $F(c)$. Тогда из $F(c) = x_c$ следует, что $K(x) \leqslant \log c + c_0$ по определению. ■

Следствие 1. У почти всех слов колмогоровская сложность равна $n - const$.

5.2 Применение

Мы лишаемся вычислимости, поэтому мы лишаемся практических применений, как с энтропией. Но есть математическое применение, так как это оптимальный алгоритм.

Можно доказать, что одноленточная машина Тьюринга, копирующая вход будет работать $\Omega(|x|^2)$.

5.3 Условная сложность

$$K(x \mid y) = K_U(x \mid y).$$

Это способ описания, который может еще использовать y: $K_F(x \mid y) := \min\{|p| \mid F(p, y) = x\}$. Аналогично U — оптимальный способ описания[1].

Замечание. $K(x) - K(x \mid \varnothing) + const$

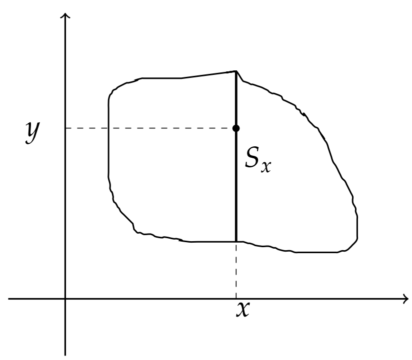
$$K(x, y) := K(\langle x, y \rangle) \qquad \langle \cdot, \cdot \rangle \text{ — какая-то кодировка пары.}$$

Свойства.

1. $K(x \mid y) \leqslant K(x) + \mathcal{O}(1)$
2. $K(x, y) \leqslant K(x) + K(y \mid x)$

Теорема 5.3.1 (Колмогоров, Левин). $K(x, y) = K(x) + K(y \mid x)$

☐ В одну сторону по свойству. Пусть $n = K(x, y)$ Пусть $S := \{(a, b) \mid K(a, b) \leqslant n\}$,



$$K(y \mid x) \leqslant \underbrace{\log|S_x|}_{m} + \mathcal{O}(\log n).$$

Рассмотрим все x, для которых $\log|S_x| \geqslant m$. Таких не более 2^{n-m+}, так как мы знаем, что $K(x, y) = n$, то внутри множества размером 2^n, есть элемент с большой сложностью, теперь множество S по размеру не более 2^{n+c}, но так как в одном сечении не более 2^m, получаем 2^{n-m+c}.

Тогда $K(x) \leqslant n - m$. Если мы подставим в неравенство выше, то получим то, что хотели. ■ Конец.

[1] Упражнение — аналогично доказать существования