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

Тамарин Вячеслав

June 12, 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. $\chi(A \cap B) \leqslant \chi(A) + \chi(B)$
3. Если наше множество $A := B \times C$, то можно описать для B и для C, поэтому можно ограничить:

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

Определение 1: Информация по Хартли

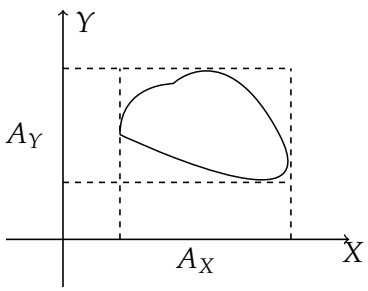
Пусть A — некоторое конечное множество. За ***информацию в множестве*** A будем принимать следующую величину:[a] $\chi(A) := \log|A|$

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

Замечание. Очевидно, второе свойство выполнено для такого определения. 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. $\chi(A_X) \leqslant \chi(A), \chi(A_Y) \leqslant \chi(A)$
3. $\chi(A) \leqslant \chi(A_X) + \chi(A_Y)$

▢ Очевидно, следует из определения. ■

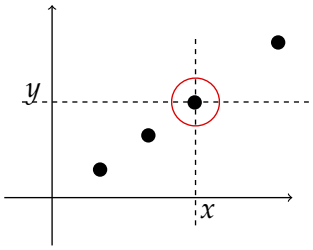


Figure 1.1: Диагональное множество

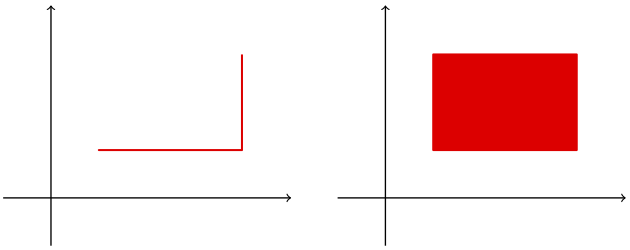


Figure 1.2: Неотличимые множества

Пример 1.1.1

Рассмотрим такой пример (см. рисунок 1.1): 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 1.2).

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 $A \subseteq A_1 \times A_2 \times A_3$ кончено. 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:

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

Пусть $A = \{(x, y, z) \mid x \in A_1,\ y \in A_2,\ z \in Z_3\}$

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

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

При этом $\sum_{i,j} K_{ij} = |A|$, так как это сумма размеров B_{ij}, которая равна размеру A.

Теперь рассмотрим множество

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

Так как некоторые z_k могли совпадать, элементов в этом множестве может быть меньше, чем $|A_{13}| \cdot |A_{23}|$, но четверки, в которых (x_i, y_j, z_k) и (x_i, y_j, z_l) принадлежат B_{ij}, будут различными.

Соответственно, $|A_{13,23}| \geqslant \sum K_{ij}^2$, так как возможны все пары z_k, z_l из B_{ij}, а их K_{ij}^2.

Таким образом,

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

Домножим на $|A_{12}|$, которое не меньше количества K_{ij} (обозначим его за n), так как каждый набор B_{ij} соответствует одному элементу A_{12}:

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

А это то, что мы и хотели доказать. ∎

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?

Если 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.

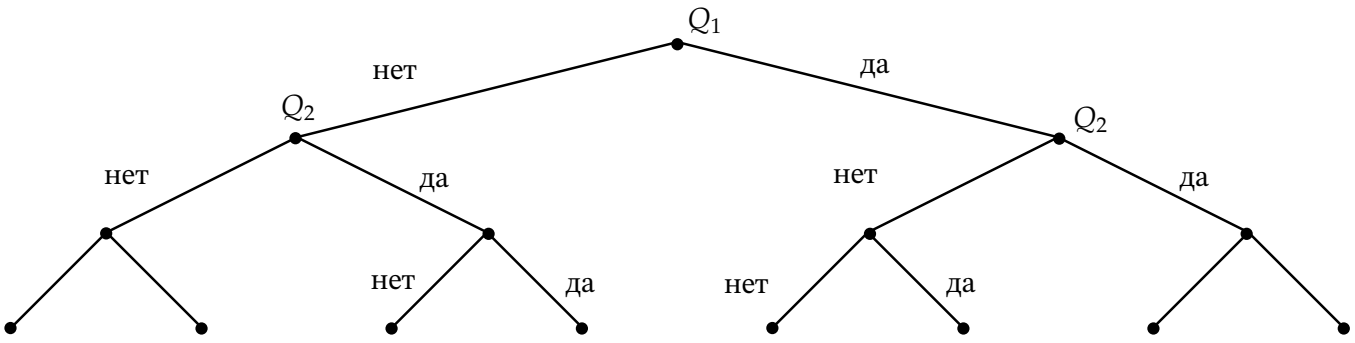


Figure 1.3: Граф вопросов

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

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

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

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

Асимметричный вариант Пусть теперь за ответ «да» мы платим 1, а за «нет» 2. 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 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, что

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

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:

$$k = \frac{t^2}{k-t} \iff t^2 = k(k-t) = k^2 - kt \iff$$

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

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

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 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.

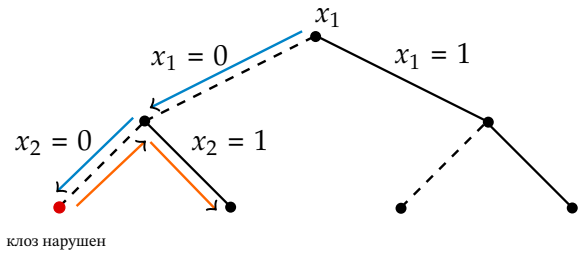
Лекция 2
8 April

1.3 Задача выполнимости

Вход: $\Phi = \wedge C_i$ — 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 т.д., 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 $(p_1, p_2, \ldots)$ 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 $p_j = \sum\limits_i p_{i,j}$.

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

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

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

■

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:

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

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

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

Свойства.

1. $H(\alpha \mid \beta) \geqslant 0$
2. $H(\alpha \mid \beta) = 0 \iff \alpha$ однозначно определяется β
3. $\forall f\colon\ H(\alpha \mid \beta) \geqslant H(f(\alpha) \mid \beta)$
4. $H(\alpha, \beta) = H(\alpha) + H(\beta \mid \alpha) = H(\beta) + H(\alpha \mid \beta)$
5. $H(\alpha, \beta) \geqslant H(\alpha)$
6. $H(\alpha) \geqslant H(\alpha \mid \beta)$

   □

   $$H(\alpha \mid \beta) - H(\alpha) = \sum p_{i,j} \frac{1}{\log \Pr[\alpha = i \mid \beta = j]} - \sum p_{i,j} \log \frac{1}{p_i} \leqslant \sum p_{i,j} \log \frac{p_i}{\Pr[\alpha = i \mid \beta = j]}$$

   По неравенству Йенсена полученное выражение меньше нуля. ■
7. Формула условной энтропии через отдельные вероятности

   $$\begin{aligned} H(\alpha \mid \beta) &= \sum_j \Pr[\beta = b_j] \cdot \sum_i \Pr[\alpha = a_i, \beta = b_j] \cdot \log \frac{1}{\Pr[\alpha = a_i, \beta = b_j]} \\ &= \sum_{i,j} p_{i,j} \cdot \log \frac{p_{*,j}}{p_{i,j}} \end{aligned}$$
8. $2H(\alpha, \beta, \gamma) H(\alpha, \beta) + H(\alpha, \gamma) + H(\beta, \gamma).$
9. $H(\alpha \mid \beta) \geqslant H(\alpha \mid \beta, \gamma)$

2.2 Применение энтропии

2.2.1 Взвешивания монеток

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 дана такая стратегия первого игрока. Тогда мы получаем 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. Построим равномерное распределение на этих исходах.

При равномерном распределении энтропия $\log 27$.

Если стратегия верная, то

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

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[3]. Вероятность того, что левая кучка будет легче в результате первого взвешивания $\frac{2k}{27}$

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

(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.

2.2.2 Оценка на биноминальные коэффициенты

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

Обозначим сумму за C.

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

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

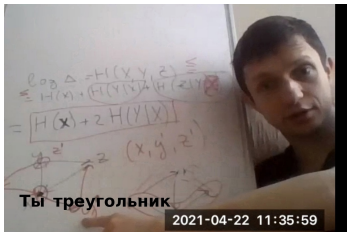
Пояснение к последнему переходу. $H(\frac{k}{n}) = \frac{k}{n}\log\frac{n}{k} + (1-\frac{k}{n})\log\frac{1}{1-\frac{k}{n}}$. С другой стороны $H(X_1) = p\log\frac{1}{p} + (1-p)\log\frac{1}{1-p}$, где p — вероятность того, что первый элемент попал в множество. Заметим, что функция $f(x) = x\log\frac{1}{x} + (1-x)\log\frac{1}{1-x}$ 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.

$$k \geqslant \mathbb{E}(N) = \mathbb{E}(\sum X_i) = \sum \mathbb{E}(X_i) = \sum p_i$$

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

Лекция 3
15 April

2.2.3 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 2.2.1. Число углов в графе всегда больше числа треугольников.

□ Пусть случная 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 найдем 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 меньше, чем треугольников. ■

Chapter 3

Теория кодирования

Определение 4: Код

Будем называть ***кодом*** функцию $C\colon \Sigma = \{a_1, \ldots, a_n\} \to \{0,1\}^*$, 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 5

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 $l_1, \ldots, l_n$ удовлетворяет неравенству

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

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

▢ Индукция по n.

- База: $n = 1$. Очевидно.
- Переход: $n \to n+1$. Пусть нам даны числа $l_1, \ldots, l_{n+1}$.
  - 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 $c, c_3, \ldots, c_{n+1}$. 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 $l_1, l_2, \ldots, l_n$ из слов вида $0, 10, 110, \ldots$. 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 кодовыми

словами $c_1, \ldots, c_n$ выполнено неравенство

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

□ Пусть $p_i(x, y)$ — моном соответствующий c_i[1], и L — большое натуральное число.

Обозначим за $M_i(x, y)$ сумму всевозможных мономов степени i.

Рассмотрим

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

Неравенство выполняется, так как код однозначно декодируем, каждый моном в левой части есть и в правой.

Подставим $x = y = \frac{1}{2}$:

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

Если неравенство Крафта-Макмиллана не выполнено для данного кода, то

$$\sum_i p_i(\tfrac{1}{2}, \tfrac{1}{2}) = \sum_i 2^{-|c_i|} = 1 + \varepsilon > 1.$$

Тогда $P^L = (1 + \varepsilon)^L > \mathcal{O}(L)$, а это противоречит рассуждению о линейности роста. ■

Теорема 3.1.3. 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 с такими длинами кодовых слов.

Докажем, что для любого декодируемого кода выполнено такое неравенство.

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 (например, $c_i = 010$ соответствует моном xyx). Давайте будем считать, 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 $M_j(x, y)$ сгруппированны 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. Тогда в $M_j(x, y)$ не более 2^j мономов.

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

[1]Например, для $c_i = 010$, $p_i = xyx$.

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

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.

Как, используя это неравенство, построить префиксный код по декодируемому? Достаточно научиться по набору длин кодовых слов, для которого выполняется неравенство, построить префиксный код. Для этого давайте будем строить бинарное дерево. Изначально у нас есть только непомеченный корень и множество $A = \{l_1, \ldots, _n\}$ — длины кодовых слов. На каждом шаге мы смотрим наше множество A и проверяем, нет ли непомеченной вершины на глубине l_i. Если есть, помечаем ее и удаляем l_i из множества. После того, как таких вершин не осталось, раздваиваем все непомеченные вершины и повторяем алгоритм, пока множество не останется пустым. Нетрудно убедиться, что если выполнено неравенство, то для каждой длины найдется слово, так как если мы в вершине на высоте h запишем 2^{-h}, то сумма в листьях будет 1. А у нас кодовым словам соответствуют помеченные вершины, которые являются листьями (это, к слову, важно для беспрефиксности). ■

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

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

□

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

■

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

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

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

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

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

Теперь по теореме 3.1.1 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, 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.

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

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 $p_n + p_{n-1}$, 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 3.1.5, 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 минимальных 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 $s1$. *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?

Воспользуемся неравенством Крафта-Макмиллана. Для целого q верно

$$\sum_i 2^{-|c_i|} = \frac{q}{2^{|c_{n-1}|}} + \frac{1}{2^{|c_n|}} \leqslant 1.$$

Но раз q целое, $\frac{q+1}{2^{|c_n|}} \leqslant 1$.

То есть мы уменьшили среднюю длину кода, сохранив при этом неравенство Крафта-Макмиллана.

Значит, 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, раскрыв обратно, получим, что код Хаффмана оптимален и для n.

■

3.2.3 Арифметическое кодирование

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

Назовем ***стандартным*** полуинтервал $\left[\overline{0.v0}, \overline{0.v1}\right)$.

Найдем максимальный стандартный интервал в отрезке $[p_1 + \cdots + p_{i-1}, p_1 + \cdots + p_i]$. 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+1}.

Предположим, что ни один стандартный интервал не попал полностью в $[a, b]$. Тогда длина $[a, b]$ не больше суммы длин двух стандартных и расстояния между ними, то есть $2^{-k} \cdot 2 + 2^{-k+1} = 2^{-k+2}$.

Но $2^{-k+2} < b - a$. 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 2 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

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

■

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 $p_1, p_2, \dots p_k$. [2]

Откажемся от условия однозначного декодирования, но будем требовать, чтобы $\varepsilon_n := \Pr[D(E(x)) \neq x]$, где $|x| = n$ стремилось к нулю $\varepsilon_n \to 0$ при $n \to \infty$.[3]

Теорема 3.3.1.
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 6

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

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

- 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 $X_i := \sum\limits_j X_{ij}$ применим неравенство Чебышева:[4]

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

    где $\mu = \mathbb{E}X_i = np_i$

    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 $n_1, \dots, n_k$ равно

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

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

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

  - Теперь оценим число типичных слов.

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

[2] считаем, что слово состоит из независимых букв
[3] Если сделать равенство нулю, то особого сжатия не будет
[4] Здесь $\operatorname{Var}[X_i]$ — дисперсия.

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

$$\begin{aligned}\log\Big(\#(\delta\text{-типичных слов})\Big) &\leqslant \log\left(n^k \cdot \max_{\delta_i} N\right) \leqslant \\ &\leqslant \max_{\delta_i} H(p_1+\delta_1, p_2+\delta_2, \ldots)\cdot n + \mathcal{O}(\log n) = \quad \text{(Переход за кадром[5])} \\ &\leqslant n \cdot \max \sum_i (p_i+\delta_i)\cdot \log\frac{1}{p_i+\delta_i} + \mathcal{O}(\log n) \leqslant \\ &\leqslant n\cdot\left(\max\sum_i p_i \log\frac{1}{p_i} + k\delta \max\log\frac{1}{p_i}\right) + \mathcal{O}(\log n) = \\ &= nH(p) + \mathcal{O}(\delta\cdot n)\end{aligned}$$

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.

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

  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?

$$\begin{aligned}\Pr[D(E(w)) = w] &= \sum_x \Pr[D(x) = \\ &= w \mid E(w) = x]\cdot\Pr[E(w)=x] \leqslant \sum_x \Pr[E(w)=x] \leqslant \\ &\leqslant 2^{L_n}\cdot\max_w \Pr[w] \leqslant 2^{((L_n - H(p))\cdot n+\mathcal{O}(\delta n)} \to 2^0\end{aligned}$$

■

Chapter 4

Коммуникационная сложность

Пусть у нас есть два игрока: Алиса и Боб. Они могут отправлять друг другу сообщения и хотят посчитать функцию (или отношение) $f\colon X \times Y \to Z$. 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 7: 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;
- каждая внутренняя вершина имеет двух потомков, ребро к первому помечено нулем, ко второму — единицей;
- каждый лист помечен значением из множества Z.

Каждая пара входов (x, y) определяют путь от корня до листа в этом дереве. Будем говорить, что коммуникационные 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 $X \times Y$, в каждой клетке (x_i, y_j) которой стоит значение $f(x_i, y_j)$.

4.1.1 Нижние оценки для детерминированного случая

Пусть наша функция $f: X \times Y \to Z$. 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 приводит 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 $b(y) = b(y')$.

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 тоже на две части, но 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. ■

4.2 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).

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

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

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

Определение 8

Будем говорить, что вероятностный протокол ε ***-вычисляет*** f, если для любой пары x, y с вероятностью r или s 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^{pub}_{eps} — минимальная высота в случае, если $r = s$.

4.3 Методы оценки коммуникационной сложности

4.3.1 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.

4.3.2 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 4.3.1. Если существует Fooling set размера s, то $\mathrm{rk}_R\, s \geqslant s$.

4.4 Пример, не соответствующий никакому протоколу

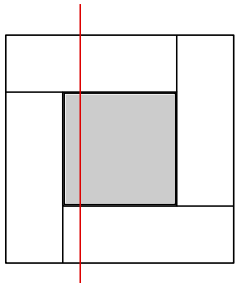


Figure 4.1: bad recd

Пусть χ — минимальное число одноцветных прямоугольников в разбиении.

Теорема 4.4.1 (GPW, 16)**.** Существует f для которой

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

Без доказательства.

4.5 Связь со схемами. Теорема Шеннона

Теорема 4.5.1 (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 $L(f) \geqslant 10n$

4.6 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 4.6.1 (Karchmer-Wigderson, 1990). *$L(f)$ — размер минимальной формулы для f, согда $L(f)$ — размер минимального протокола для KW_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 $f(x) = 1$, $f(y) = 0$. Тогда $g(y) = h(y) = 1$, а для Алисы хотя бы одно из двух значений единица.Тогда Алиса посылает информацию, где 1, 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. То есть $\forall x \in X_l\ \forall y \in Y_l\colon x \neq y$.
  Тогда либо $\forall i\colon x_i = 0 \wedge y_i = 1$, либо наоборот.
  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.

■

4.7 Теория информации в коммуникационной сложности

Лекция 6
6 May

4.7.1 Подсчет функции индексов

Определим

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

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.

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

Покажем, что полученная сумма большая.

Зафиксируем i и распишем по определению взаимной информации:

$$
\begin{aligned}
I(M : y_i) = H(y_i) - H(y_i \mid M) = & && (H(y_i) = 1) \\
= 1 - \mathbb{E}_m \left(H(y_i \mid M = m_i, x = i \right) = \\
= \sum_i 1 - H(r^i_m) = \\
= n - n \sum_i \frac{1}{n} H(\mathbb{E}(r^i_m)) = \\
= n \cdot (1 - H(\mathbb{E}_{m,i}(r^{'}_m))) \leqslant \\
\leqslant n \cdot (1 - H(\tfrac{1}{2} - \delta)) = \\
= \Omega(\delta^2 n)
\end{aligned}
$$

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

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

Теперь $\log|M| \geqslant H(M) \geqslant \Omega(\delta^2 n) \leqslant \delta n$.

$$\frac{1}{2}(1 - 2\delta) + o \leqslant 2\delta n.$$

Определение 9

Пусть μ — мера на $X \times Y$.

$\mathrm{IC}^{ext}_\mu := I(\pi(x,y) : (X,Y))$.

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

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

▢ Первое неравенство очевидно. Второе докажем потом. ■

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

□ ..... ■

Лекция 7
20 May

Докажем $I(\pi : x, y) \geqslant I(\pi : x \mid y) + I(\pi : y \mid x)$.

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 – префикс π, то есть то, что Алиса и Боб отправили за i-ый раунд (i-ый бит).

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

Аналогично попробуем нарезать слагаемые правой части:

$$I(\pi : x \mid y) = \sum_i I(\pi_i : x \mid y, \pi_i)$$

$$I(\pi : y \mid x) = \sum_i I(\pi_i : y \mid x, \pi_i)$$

В ход Боба первое слагаемое «равно нулю»[1], так как π_i определяется y-ом. Аналогично «второе равно» нулю, когда ходит Алиса. Поэтому каждый раз неравенство сохраняется.

Чтобы исправить скрытую фундаментальную ошибку распишем через матожидания

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

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

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

Это уже корректное утверждение.

[1]Это и есть баг, на самом деле это случная величина

Chapter 5

Колмогоровская сложность

5.1 Определения

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 10

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

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

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

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

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

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 $S \coloneqq \{(a,b) \mid K(a,b) \leqslant n\}$,

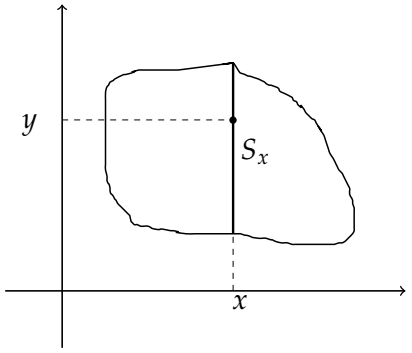


Figure 5.1: kolmo thm

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

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] Упражнении — аналогично доказать существования