Содержание

1 Введение

Обзор курса: понятие информации, энтропия Шеннона, колмогоровская сложность, коды, исправляющие ошибки, коммуникационная сложность.

Примерный адрес страницы курса: `/shad/base/Spring2017`.

2 Тривиум

Информация по Хартли (1928): текст из n символов из алфавита Σ кодируется $\log_2 |\Sigma|^n$ битами (далее логарифмы по умолчанию двоичные). Определение незамысловатое, но уже полезное.

Пример 1. Известно, что $x \in A$, сказано, что $x \in B$. Сколько информации передано? Ясно, что было $\log |A|$ информации, стало $\log |A \cap B|$. Значит передано $\log \frac{|A|}{|A \cap B|}$ бит.

Пример 2. Имеем n монет, одна из них фальшивая, легче остальных. Сколько нужно взвешиваний, чтобы её найти? Исходно не хватает $\log n$ информации, каждое взвешивание имеет 3 исхода, стало быть меньше, чем за $\frac{\log n}{\log 3}$ взвешиваний найти не получится.

Пример 3. $x \in S_n$ — перестановка. Можно сравнивать два элемента. Сколько нужно сравнений, чтобы найти перестановку?

$\log n! = \log \sqrt{2\pi n} \left(\frac{n}{e}\right)^n (1 + o(1)) = n \log n - n \log e + \frac{1}{2} \log n + O(1)$. Можно ли асимптотически приблизиться к этой границе?

Естественный алгоритм: сортировка вставками (выглядит довольно оптимально по сравнениям, не учитываем сдвиги). Используется $\lceil \log 1 \rceil + \lceil \log 2 \rceil + \ldots + \lceil \log n - 1 \rceil \leqslant (n-1) + \log(n-1)! = OPT(n) + n - 1 - \log n$.

3 Бинарный поиск

$A = [1, \ldots, m]$, нужно найти в нём $y \in \{1, \ldots, m\}$ с помощью сравнения $x \overset{?}{<} y$. Ясно, что нужно $\lceil \log_2 m \rceil$ вопросов. А что будет, если оппонент может соврать 1 раз? Легко придумать алгоритм, который даёт $3 \log n$ сравнений и $2 \log n$ (можно и лучше). Нас будет интересовать постановка, когда Responder (R) может соврать Questioner'у (Q) в доле вопросов не более ε.

Более формально, игра проходит с объявлением числа раундов n в самом начале игры и не более $n\varepsilon$ неверных ответов. Вопрос ставится так: при каких n существует стратегия у Q, которая гарантированно угадывает число? Утверждается, что можно предъявить алгоритм, работающий за $c(\varepsilon) \log n$ сравнений, чем мы и займёмся.

Ясно, что состояние бинарного поиска — это вершина бинарного дерева. Устроим алгоритм не в виде спуска по дереву, а в виде блуждания. Находясь в вершине, соответствующей числам $\{l, \ldots, r\}$, зададим вопросы $l \leqslant x, x \leqslant r$? Если получен хотя бы один отрицательный ответ, пойдём вверх. Далее, кроме случая, когда мы стоим в листе, задаём вопрос $m \leqslant x$ и идём в нужную сторону.

Утверждение 1. Лист, в который мы попадали чаще всего, есть ответ (при достаточно большой длине блуждания).

Доказательство. Подвесим дерево за лист x, тогда, если ориентировать рёбра к этому листу, то против этого направления можно идти только если среди ответов на данном шаге была ложь. В самом деле, разбор случаев помогает в этом убедиться.

Разделим все наши шаги на шаги вперёд f, назад b, l_x — шаги в листе x, l_{other} — шаги в других листах. Тогда можем утверждать, что $f \leqslant b + \log m, b + l_{other} \leqslant \varepsilon n, f + b + l_x + l_{other} = cn, c \geqslant \frac{1}{3}$. Итого $l_x \geqslant cn - (b + \log n) - \varepsilon n \leqslant cn - \log m - 2\varepsilon n$, а $l_{other} \leqslant \varepsilon n$.

$cn - \log m - 2\varepsilon n \geqslant \varepsilon n \Rightarrow cn - \log m \geqslant 3\varepsilon n \Rightarrow n \geqslant \frac{\log m}{c - 3\varepsilon}$. □

Константы, ясное дело, оценены грубо. Также про задачу известно, что при $\varepsilon > \frac{1}{2}$ всё плохо (ответ найти нельзя), при достаточно малых $\varepsilon < \frac{1}{10}$ всё совсем хорошо, при промежуточных можно получит вариации (например, экспоненциальный рост). Известны точные ответы для небольшого константного числа ошибок, и для некоторых вариаций (например, оффлайн поиск). Задача имеет связи с кодами, исправляющими ошибки.

4 Энтропия Шеннона

Пусть ξ — дискретная случайная величина, принимающая свои значения с вероятностями $(p_1, \ldots, p_m)$, тогда энтропия по Шеннону есть $H(\xi) = \sum p_i \log_2 \frac{1}{p_i}$. Если величина задана на $(\mathbb{R}, \mathcal{B}(\mathbb{R}), P)$ и умеет плотность f_ξ, то $H(\xi) = -E \log_2 f_\xi$.

Ясно, что $H(\xi) \geqslant 0$. Покажем, что $H(\xi) \leqslant \log m, H(\xi) = \log m \Leftrightarrow p_i = \frac{1}{m}$. Это очевидно следует из следующей леммы:

Лемма (неравенство Гиббса)**.** *Если* $\sum q_i \leqslant 1, q_i > 0$*, то*

$$-\sum p_i \log p_i \leqslant -\sum p_i \log q_i,$$

Доказательство. По неравенству Йенсена $\sum p_i \log \frac{q_i}{p_i} \leqslant \log \sum p_i \frac{q_i}{p_i} \leqslant 0$. □

5 Посимвольное кодирование

Пусть Σ — алфавит, $T \in \Sigma^*$, $f : \Sigma \to \{0,1\}^*$ — некоторое кодирование символов, а $f^{(n)} : \Sigma^n \to \{0,1\}^*$ — посимвольное сжатие текста.

Определение 1. f — префиксное кодирование, если $\forall a, b \in \Sigma$ неверно, что $f(a) \sqsubset f(b)$ или $f(b) \sqsubset f(a)$.

Определение 2. Код f называется неоднозначным, если $\exists x \neq y \in \Sigma^*$: $f(x) = f(y)$.

Оптимальность будем рассматривать в следующем смысле: пусть даны частоты $p_1, \dots, p_m$, с которыми встречаются символы, тогда средней длиной кода f называется $c(f) = \sum\limits_{i=1}^{m} p_i |f(a_i)|$.

Лемма (Крафта-Макмилана)**.** *Пусть $n_i = |f(a_i)|$. Пусть заданы частоты $n_1, \dots, n_m$, тогда однозначный код с такими длинами существует тогда и только тогда, когда $\sum 2^{-n_i} \leqslant 1$.*

Теорема 1. *Пусть f — однозначный код, тогда*

- *f — однозначный код $\Rightarrow c(f) \geqslant H(p_1, \dots, p_m)$*
- *$\exists$ однозначный код $f : c(f) < H(p_1, \dots, p_m) + 1$*

Доказательство. В одну сторону воспользуемся леммой Крафта-Макмилана и неравенством Гиббса: $c(f) = \sum p_i n_i = -\sum p_i \log 2^{-n_i} \geqslant -\sum p_i \log p_i = H(p_1, \dots, p_m)$.

В другую сторону, положим $n_i = \lceil \log \frac{1}{p_i} \rceil$, тогда по лемме Крафта-Макмилана существует код f с такими длинами. Тогда $c(f) = \sum p_i n_i = \sum p_i \lceil \log \frac{1}{p_i} \rceil < \sum p_i (\log \frac{1}{p_i} + 1) = H(p_1, \dots, p_m) + 1$. □

Доказательство (леммы Крафта-Макмилана). Построим по данным длинам код. Пусть $n_1 \geqslant \ldots \geqslant n_m$. Отложим отрезки длин 2^{-n_i} на отрезке $[0; 1]$ (это возможно по условию на сумму длин). Тогда $[\sum\limits_{i=1}^{k} 2^{-n_i}; \sum\limits_{i=1}^{k+1}]$ — двоичный отрезок, то есть имеет вид $[\frac{l}{2^i}; \frac{l+1}{2^i}]$. Зная это, предъявим код следующим образом: будем спускаться по обычному бинарному отрезка, осуществляющему дихотомию отрезка $[0; 1]$ и в тот момент, когда мы приходим в двоичный отрезок, завершаемся, выдавая соответствующий символ. Таким образом мы построили однозначный (более того, префиксный) код.

В обратную сторону, рассмотрим

$$(\sum_{i=1}^{m} 2^{-n_i})^k = \sum_{w \in \Sigma^*} 2^{-|f(w)|} = \sum_{l} \sum_{w : |f(w)| = l} 2^{-l}.$$

Заметим, что в силу различности кодов длины l, внутренняя сумма не больше 1, а также, что $l \leqslant k \cdot d, d = \max\limits_{i=1 \dots m} n_i$, тогда получим что, исходное выражение не больше dk. Тогда для, если $\sum 2^{-n_i} > 1$, то для достаточно большого k получим противоречие с неравенством. □

6 Оптимальный код

Хотим решить задачу минимизации $\sum p_i n_i \to \min$ при условии $\sum 2^{-n_i} \leqslant 1$. Договоримся сразу $p_1 \geqslant \ldots \geqslant p_m, n_1 \leqslant \ldots \leqslant n_m$.

Утверждение 2. Пусть f — оптимальный код, тогда, с очевидностью, $n_{m-1} = n_m$.

Тогда перейдём к задаче $\sum\limits_{i=1}^{m-1} p'_i n_i \to \min$ с условием $\sum\limits_{i=1}^{m-1} 2^{-n_i} \leqslant 1$, где p'_i — исправленные вероятности. Пусть $\hat{n_i}$ — оптимальное решение второй задачи, тогда понятно, что по ним можно восстановить решение исходной: $n_i = \hat{n_i}$ для $i \leqslant m-2, n_{m-1} = n_m = \widehat{n_{m-1}} + 1$.

Таким образом, мы получили оптимальный код (это, очевидно, код Хаффмана).

Рассмотрим теперь следующую задачу: дана дискретная случайная величина $X \sim (p_1, \ldots, p_m)$ и честная монетка: $Z_1, \ldots \sim Bern\left(\frac{1}{2}\right)$. Нужно придумать алгоритм, который по $Z_1, \ldots$ моделирует величину X. Его естественно представлять деревом (возможно, бесконечным).

Если Y — некоторая случайная величина, уже заданная таким деревом, притом значения во всех листьях различны, тогда ясно, что ожидаемая глубина его $ET = \sum\limits_y d(y)P(Y=y) = \sum\limits_y d(y)2^{-d(y)} = H(Y)$.

Утверждение 3. Пусть p_i — двоично-рациональные. $p_i = \sum\limits_j 2^{-n_{ij}}$. Тогда $H(X) \leqslant ET < H(X) + 2$.

Доказательство. Оценка снизу явствует из того, что у построенного дерева в некоторых листах значения одинаковы, при замене их на разные, энтропия не уменьшится. □

Тоже самое можно сказать и про не двоично-рациональные вероятности.