Теория информации

Лекция 3

Количество информации в случайной величине

В прошлой лекции было определено Шенноновское определение количества информации

$$h(x=a_i) = \log_2 \frac{1}{p_i}$$

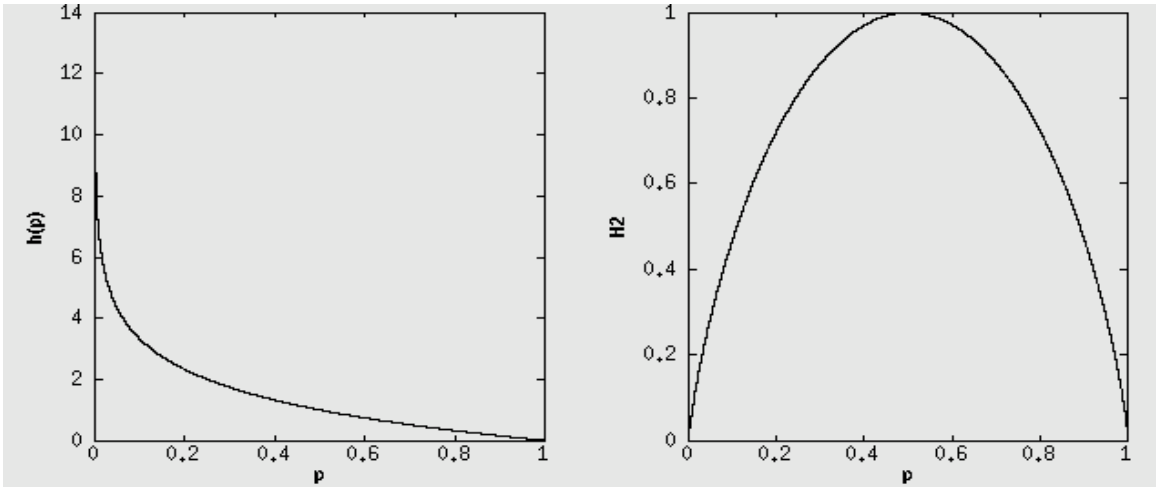
и энтропия ансамбля

$$H(X) = \sum_i p_i \log_2 \frac{1}{p_i}$$

как мера среднего количества информации по ансамблю. Убедимся, что эти величины действительно характеризуют количество информации. Определенная в первой лекции двоичная энтропия

$$H_2(p) = p\log_2 \frac{1}{p} + (1-p)\log_2 \frac{1}{1-p}$$

это среднее количество информации для ансамбля из двух элементов $A_x=\{a,b\}$ и распределением $P_x=\{p,1-p\}$. Ниже показаны графики количества информации и двоичной энтропии.



Из графиков видно, что чем менее вероятным оказывается значение случайной

переменной, тем большее количество информации оно несет, а также что максимум энтропии соответствует равномерному распределению.

Обратим внимание на одно свойство функции $\log_2\frac{1}{p}$. Если мы узнаем значение двух независимых переменных x и y , а поскольку они независимы $P(x,y)=P(x)P(y)$, то интуитивно ясно, что количество информации в этом случе должно быть аддитивной величиной. Этому свойству удовлетворяет функция $\log_2\frac{1}{p}$:

$$h(x,y) = \log_2\frac{1}{P(x,y)} = \log_2\frac{1}{P(x)P(y)} = \log_2\frac{1}{P(x)}+\log_2\frac{1}{P(y)} = h(x)+h(y) \quad . \quad (3.1)$$

Легко показать, что для независимых переменных выполняется и равенство

$$H(X,Y) = H(X) + H(Y) \quad , \quad (3.2)$$

т.е. энтропия аддитивна для независимых переменных.

Рассмотрим игру в угадывание. Один игрок загадывает предмет, а другой задавая вопросы, ответ на которые либо да либо нет, угадывает за заданное число ходов. Упростим эту игру до отгадывания одного числа от 0 до 63. Всего 64 возможности и для того, чтобы угадать, достаточно задать следующие 6 вопросов:

1. Верно ли, что $x \geq 32$?
2. Верно ли, что $x \bmod 32 \geq 16$?
3. Верно ли, что $x \bmod 16 \geq 8$?
4. Верно ли, что $x \bmod 8 \geq 4$?
5. Верно ли, что $x \bmod 4 \geq 2$?
6. Верно ли, что $x \bmod 2 = 1$?

Здесь $x \bmod y$ это остаток от целочисленного деления x на y , например $35 \bmod 32 = 3$. Если множество ответов на эти вопросы $\{да, нет\}$ поставить в соответствие множеству $\{1,0\}$, то собственно двоичная запись ответов и будет давать загаданное число. Например, число 35 даст набор ответов $100011 = 1*2^5+0*2^4+0*2^3+0*2^2+1*2^1+1*2^0 = 32+2+1 = 35$. Можно сказать, что наша последовательность вопросов кодирует загаданное число как двоичную последовательность (файл). Если предположить, что все числа от 0 до 63 равновероятны, тогда ответы на наши вопросы независимы и каждый ответ содержит информации

$$\log_2 \frac{1}{0.5} = 1 \ \ бит \ \ ,$$

а полное Шенноновское количество информации 6 бит (по одному на каждый вопрос). Таким образом, количество информации это длина двоичного файла (последовательности) кодирующего переменную x . Остается убедиться в этом для ансамбля имеющие неравные вероятности в распределении P_x .

Рассмотрим для этого другую игру, похожую на морской бой, в которой всего один корабль из одной клетки на поле в $8 \times 8 = 64$ клетки. Каждый выстрел формирует ансамбль, в котором $A_x = \{да, нет\}$. В самом начале игры

$$P(да) = \frac{1}{64}, \qquad P(нет) = \frac{63}{64} \ \ .$$

Если первый выстрел был мимо, то на втором ходе

$$P(да) = \frac{1}{63}, \qquad P(нет) = \frac{62}{63} \ \ ,$$

на третьем ходе (если первые два мимо)

$$P(да) = \frac{1}{62}, \qquad P(нет) = \frac{61}{62}$$

и т.д., пока не будет попадания. Если игрок удачлив и попал с первого раза, то для такого события

$$h(x) = h_1(да) = \log_2 64 = 6 \ \ бит \ \ .$$

Кажется странным, что однобитовое значение (единственное да) дало сразу 6 бит информации. Но игрок не только узнал, что на какой-то клетке есть «корабль», но и то, что на остальных 63-х клетках пусто.

Если же первый ход был промахом, то

$$h(x) = h_1(нет) = \log_2 \frac{1}{63/64} \approx 0.0227 \ \ бит \ \ .$$

Что же означает количество информации в 0.0227 бит? Если сложить все количество информации соответствующее первым 32 промахам, то получим следующее:

$$h_1(нет) + h_2(нет) + ... + h_{32}(нет) = \log_2 \frac{64}{63} + \log_2 \frac{63}{62} + ... + \log_2 \frac{33}{32} = 1 \ \ бит \ \ .$$

Что значит это круглое число? Если сравнить с предыдущей игрой, то можно увидеть, что за 32 неудачных хода отброшена половина гипотез, что и дало 1 бит информации. Это эквивалентно ответу на 1-ый вопрос в предыдущей игре. Если отбросить еще половину оставшихся гипотез, то на 48-ом неудачном ходу, количество информации будет

$$\log_2\frac{64}{63}+\ldots+\log_2\frac{17}{16}=2.0 \ \ бит \ .$$

Если на 49-м ходу игрок найдет «корабль», то такой ход принесет $\log_2 16=4.0 \ \ бит$ информации. А всего информации вместе с 48-ю промахами будет 6 бит. Обобщая этот результат на нахождение «корабля» на том ходу, когда останется n непроверенных клеток, получим следующее выражение для количества информации:

$$\log_2\frac{64}{63}+\ldots+\log_2\frac{n+1}{n}+\log\frac{n}{1}=\log_2\left(\frac{64}{63}\cdot\frac{63}{62}\cdot\ldots\cdot\frac{n+1}{n}\cdot\frac{n}{1}\right)=\log_2 64=6 \ \ бит \ .$$

Этот пример иллюстрирует то, что мера количества информации Шеннона $h(x)$ годится и для случая неравномерного распределения вероятностей P_x .

Сжатие данных

Рассмотрим для начала пример некоего искусственного языка, состоящего из $2^{15}=32768$ слов составленных из 5 случайно выбранных букв английского алфавита, а в качестве вероятности появления букв возьмем вероятности появления букв в реальном английском языке, в котором, например, $P(a)\approx 0.0575$, а $P(z)\approx 0.001$. Тогда в словаре из 2^{15} слов будет около 1900 слов начинающихся на *a*, и около 32-х слов, начинающихся на *z*. Начинающихся на *aa* примерно 108, а на *az* всего 1 или 2.

Теперь предположим, что мы читаем документ на этом абстрактном языке. Если за раз прочитывается одно слово, то количество информации в одном слове $\log_2 32768=15$ бит, при предположении, что все слова в этом языке используются равновероятно. Таким образом среднее число бит на символ $15/5=3$ бита. Однако если мы читаем документ по одному символу, то буква *a* приносит $\log_2(1/0.0575)\approx 4.12$ бит, а буква *z* приносит $\log_2(1/0.0575)\approx 9.96$ бит. То есть большая часть информации в случае слова, начинающегося на *z* сосредоточена в первой букве. Таким образом, чем более редкое сочетание в начале слова, тем с большей вероятностью можно угадать остаток слова. Все это

верно и для реального естественного языка. Например, слов, начинающихся с ло- (лотос, лодка, лопата, ...) в русском языке намного больше чем слов, начинающихся с йо- (йод, йог, йогурт).

Итак, подходим к идее сжатия данных. Маловероятное значение случайной переменной дает больше бит информации, чем вероятное. Если удастся какой-либо источник (текст например) закодировать в файл, размером L бит на символ, а затем надежно восстановить источник из этого файла, то можно будет с уверенностью сказать, что источник содержал информации не более L бит на символ. Простой пример это текстовый файл в кодировке ASCII, в котором, несмотря на 8-битовое представление символа, нужно только 7 бит. Значит, этот файл можно сжать с фактором 7/8.

Самый простой способ посчитать количество информации в случайной переменной это посчитать число возможных исходов, т.е. $|A_x|$. Если поставить соответствие между элементами множества A_x и двоичным числом, то длина этого числа в битах равна $\log_2|A_x|$ если конечно $|A_x|$ кратно 2. Таким образом, количество бит, необходимое для представления ансамбля X определим как

$$H_0(X) = \log_2|A_x| \quad . \tag{3.3}$$

Число $H_0(X)$ это нижняя граница числа вопросов в игре в угадывание на множестве A_x для гарантированного угадывания. Из (3.3) ясно, что $H_0(X)$ величина аддитивная. Для упорядоченной пары x, y , когда имеется $|A_x|\cdot|A_y|$ возможных исходов

$$H_0(X,Y) = H_0(X)+H_0(Y) \quad . \tag{3.4}$$

Как мера информации в ансамбле, величина $H_0(X)$ не учитывает вероятностные особенности A_x , а просто соответствует случаю отображения множества A_x в строку битов фиксированной длины. Очевидно, что нельзя реализовать алгоритм, сжимающий исходные данные из A_x в строку битов короче $H_0(X)$, так чтобы осуществить декомпрессию без ошибок. С другой стороны, возможно сжатие с потерями, которые являются приемлемыми. Сжатие с потерями широко используется при сжатии изображений или звуковых файлов.

Рассмотрим сжатие с потерями на примере. Пусть случайная величина принадлежит множеству $A_x=\{a,b,c,d,e,f,g,h\}$ с распределением вероятностей $P_x=\left\{\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{3}{16},\frac{1}{64},\frac{1}{64},\frac{1}{64},\frac{1}{64}\right\}$. В этом случае $H_0(X) = \log_2 8 = 3$ бита, то есть можно представить множество A_x 3-битовыми строками как показано ниже в таблице.

x	a	b	c	d	e	f	g	h
$c(x)$	000	001	010	011	100	101	110	111

Обратим внимание на то, что $P(x \in \{a,b,c,d\}) = \frac{15}{16}$, а это значит, что если допустить вероятность того, что битовая строка не найдет подходящего символа из набора $\delta = \frac{1}{16}$, то можно оставить только половину множества A_x. Таким образом, для битового кодирования в данном случае понадобится только 2 бита:

x	a	b	c	d
$c(x)$	00	01	10	11

Это, в свою очередь, будет означать сжатие с потерями с фактором 2/3 и вероятностью ошибки 1/16.