Код Шеннона

5 января 2023 г. 1:22

Рассмотрим ансамбль $X = \{1, 2, \dots, M\}$ с вероятностями $\{p_1, p_2, \dots, p_M\}$. Предположим, что $p_1 \geq p_2 \geq \dots \geq p_M$. Для каждого $x \in X$ вычислим *кумулятивную вероятность* как

$$q_1 = 0$$

$$q_i = \sum_{j=1}^{i-1} p_j,\ i = 2, \dots, M.$$

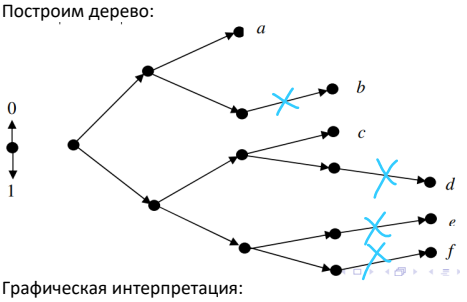
Кодовое слово Шеннона для x_i – двоичная запись первых $l_i = \lceil -\log p_i \rceil$ бит после запятой двоичного представления q_i .

Пример:

x	$p(x)$	$q(x)$	$l(x)$	$c(x)$
a	0.35	0	2	00...
b	0.20	0.35	3	010...
c	0.15	0.55	3	100...
d	0.1	0.70	4	1011...
e	0.1	0.80	4	1100...
f	0.1	0.90	4	1110...

$$\bar{l} = \sum_x p(x) l(x) = 2.95 > H = 2.4016$$

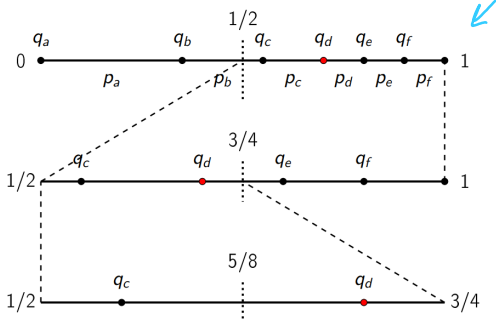
Построим дерево:



Графическая интерпретация:

1. Рисуем отрезок от 0 до 1, отмечаем на нем все посчитанные кумулятивные вероятности q
2. Делим отрезок пополам и смотрим, в какой половине находится q нужного символа (если справа, то первый бит кодового слова 1, иначе 0), далее рассматриваем ее
3. Делим половину пополам, повторяем действия. Выполняем, пока на отрезке не останется только одна кумулятивная вероятность

$x = d;\ q(x) = 0.7$



Свойства кода Шеннона:

1. По теореме побуквенного кодирования т.к. $l_i = \lceil -log(p_i) \rceil < -log(p_i) + 1$, то $\bar{l} < H + 1$ (т.е. побуквенный неравномерный префиксный код "код Шеннона" существует)
2. Т.к. код является префиксным, он однозначно декодируемый
3. Требует сортировки вероятностей

-

Свойства префиксного кода:

1. Любое кодовое слово не должно быть началом другого
2. Декодирование однозначно (причем однозначно декодируемый код не обязательно префиксный)
3. Кодовым словам соответствуют только листья двоичного кодового дерева
4. Древовидный код является префиксным

-

Преимущество по сравнению с кодом Хаффмана - алгоритм реализуется без построения дерева

Проблема: если символов очень много (близко к ∞), то сортировка вероятностей усложняет всю процедуру
Мы имеем побуквенный код и стремимся расширить алфавит для улучшения сжатия, т.е. пытаемся приблизиться к наименьшей возможной избыточности. Код Шеннона такую возможность не дает, т.к. требует сортировки
Если не сортировать, получится шляпа:

x	$p(x)$	$q(x)$	$l(x)$	$c(x)$
a	0.1	0	4	0000...
b	0.3	0.1	2	00...

c	0.6	0.4	1	0...

Используя код Гилберта-Мура, можно избежать необходимость сортировки вероятностей за счет введения дополнительной избыточности
Код Гилберта-Мура сжимает еще хуже, чем код Шеннона, зато не требует сортировки