Функция скорость-искажение

5 января 2023 г. 1:27

Говоря о сжатии данных с потерями, мы принимаем во внимание ошибки, т.е. искажение данных.

Рассмотрим множество $X = \{x\}$, элементами которого являются сообщения, генерируемые источником, и множество $Y = \{y\}$, которое будем называть аппроксимирующим. Т.е. после генерации сообщений источником произойдет их аппроксимация, и приемник будет получать уже Y, а не X.

В результате аппроксимации появилось различие между X и Y. Это отличие и показывает мера искажения.

Мера искажения $d_n(\boldsymbol{x}, \boldsymbol{y})$ определяется как функция $X^n * Y^n = \{(\boldsymbol{x}, \boldsymbol{y}),\ \boldsymbol{x} \in X^n,\ \boldsymbol{y} \in Y^n\}$ ("произведение множества векторов X длины n и множества векторов Y длины n ... ")и обладает следующими ограничениями:

1. Для любых $\boldsymbol{x} = (x_1, \dots, x_n)$, $\boldsymbol{y} = (y_1, \dots, y_n)$:
   $d_n(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, y_i)$ искажение между конкретными x и y
2. $d(x, y) \geq 0$ для всех $x \in X,\ y \in Y$.

Существуют различные меры искажения.

1. Вероятностная (Хеммингова) мера искажения
   Когда символы x и y не похожи, искажение 1, если похожи - 0
   $$d(x,y) = \begin{cases} 0, & x = y, \\ 1, & x \neq y. \end{cases}$$
   $$d_n(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, y_i) \approx \mathrm{P}(x \neq y)$$
2. Квадратичная мера искажения
   $$d(x,y) = (x-y)^2$$
   $d_n(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2$ – мощность шума, который появляется в результате кодирования.
3. Абсолютная мера искажения
   (обычно встречается в случаях, когда нужно обойтись без умножения, поэтому используется вместо квадратичной)
   $$d(x,y) = |x-y|$$
   $$d_n(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i|$$

Кодирование с заданным критерием искажения (общая идея)

1. Последовательность сообщений источника разбивается на блоки $\boldsymbol{x}$ длины n.
2. Каждый из блоков $\boldsymbol{x}$ аппроксимируется последовательностью $\boldsymbol{y} \in Y^n$. Как это происходит?
3. Аппроксимирующие подмножества выбираются из *кодовой книги* (дискретного подмножества) $B = \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_M\} \subseteq Y^n$, $M = |B|$. для каждого входного X
4. Задача кодера: выбрать наилучшее слово $\boldsymbol{b}_m$ из B и передать по каналу (сохранить) m, где $m = \arg \min_{i \in \{1,\dots,M\}} d_n(\boldsymbol{b}_i, \boldsymbol{x})$.
5. Декодер тоже хранит B и по индексу m выдаёт аппроксимирующую последовательность $\boldsymbol{y} = \boldsymbol{b}_m$.

   Декодер тоже хранит эту кодовую книгу и выдает аппроксимирующую последовательность по известному индексу m

Что делает кодер:
Берет вектор $\boldsymbol{x}$ длины n, который нам пришел
Перебирает все векторы кодовой книги
Из них выбирает тот, который по метрике искажения минимален (то есть наиболее похожий, с наименьшим искажением)
Передает индекс найденного аппроксимирующего множества

- Для передачи номера m достаточно $\lceil \log M \rceil$ бит, поэтому скорость кода
  $$R = \frac{\lceil \log M \rceil}{n} \quad \text{бит/сообщение.}$$
  если передаем равномерным кодом
- При фиксированной кодовой книге и вероятностной модели источника $p(\boldsymbol{x})$ можно подсчитать *среднюю ошибку*
  $$D = \mathrm{E}\{d_n(\boldsymbol{x}, \boldsymbol{b})\} = \mathrm{E}\{d_n(\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x}))\},$$
  где $\boldsymbol{y}(\boldsymbol{x})$ – аппроксимирующая последовательность, которую декодер выбирает для $\boldsymbol{x}$.

Т.е. для каждого входного вектора мы знаем способ, которым найдем для него аппроксимирующее подмножество (по минимуму расстояний), соответственно, мы можем вычислить для него ошибку, а дальше умножаем на вероятность появления этого вектора и уже можем вычислить среднюю ошибку

R – скорость кода

D – средняя ошибка

Скорость кода выбирается по средней ошибке таким образом:
Перебираем все возможные кодовые книги, такие что для них искажение будет меньше или равно средней ошибке, и выбираем из этих вариантов тот, который дает минимальное количество бит, таким образом, минимизируем R при заданном D

- Для заданного n можно варьировать различными B. В результате возникнут различные пары (D, R).
- Задача: минимизировать D при заданном R или минимизировать R при заданном D.
- В итоге, можно получить функцию скорость-искажение $R(D)$ (rate-distortion function) вида

$$R(D) = \inf_n \min_{B:D(R) \leq D} R_n(B),$$

где inf – нижняя грань.
- Также можно определить функцию искажение-скорость $D(R)$.