Source coding

Last modified: 2026-11-12 01:08:55 (PST)

Codes

Definition 0.1 (Binary code)**.** Let X be a discrete random variable. A **binary code** for X assigns each value x a *codeword*, a string of $\ell(x)$ bits.

Definition 0.2 (Uniquely decodable code)**.** A binary code is **uniquely decodable** if every finite sequence of codewords, written one after another, can be split back into codewords in only one way.

Definition 0.3 (Expected length)**.** The **expected length** of a binary code for a discrete random variable X is the expectation of its codeword length:

$$L \stackrel{\text{def}}{=} \mathbb{E}[\ell(X)].$$

Theorem 0.1 (Expected length as a sum over values)**.** *Let X be a discrete random variable with probability mass function $p(x) \stackrel{def}{=} \mathbb{P}(X = x)$. The expected length of a binary code for X is*

$$L = \sum_x p(x)\,\ell(x).$$

Proof

Proof. The codeword length $\ell(X)$ is a random variable whose possible values are the codeword lengths $\ell(x)$. By the definition of expectation,

$$L = \sum_k k\,\mathbb{P}(\ell(X) = k).$$

The event $\ell(X) = k$ is the union of the disjoint events $X = x$ over the values x with $\ell(x) = k$. So $\mathbb{P}(\ell(X) = k) = \sum_{x:\ell(x)=k} p(x)$, and

$$L = \sum_{k} \sum_{x:\ell(x)=k} k\, p(x).$$

On the inner sum $k = \ell(x)$. Every value x lies in exactly one inner sum, so the double sum runs once over each value:

$$L = \sum_{x} \ell(x)\, p(x).$$

□

Example 0.1 (Expected length of a code for a three-valued variable)**.**
Let X have probability mass function $p = (1/2, 1/4, 1/4)$, and let a binary code for X have codeword lengths $\ell = (1, 2, 2)$. The code below computes the expected length from its definition, as the expectation of $\ell(X)$, and as the sum in Theorem 0.1.

```
p <- c(1 / 2, 1 / 4, 1 / 4)
len <- c(1, 2, 2)
lengths_seen <- sort(unique(len))
prob_of_length <- vapply(
  lengths_seen,
  function(k) sum(p[len == k]),
  numeric(1)
)
from_definition <- sum(lengths_seen * prob_of_length)
from_sum <- sum(p * len)
c(from_definition = from_definition, from_sum = from_sum)
#> from_definition        from_sum
#>             1.5             1.5
```

Both give $L = 1.5$ bits, as Theorem 0.1 states.

Remark 0.1 (Short codewords for common values). A code with a small expected length gives frequent values short codewords and rare values long ones, since each value's $\ell(x)$ counts in L weighted by its probability.

The source coding theorem

Theorem 0.2 (Source coding theorem)**.** *Let X be a discrete random variable that takes a finite number of values, at least two of them with nonzero probability, and let $\mathbb{H}(X)$ be its entropy with base-2 logarithms.*

1. *Every uniquely decodable binary code for X has expected length $L \geq \mathbb{H}(X)$.*
2. *Some uniquely decodable binary code for X has expected length $L < \mathbb{H}(X) + 1$.*
3. *Coding n independent copies of X at a time, some uniquely decodable binary code spends, on average, fewer than $\mathbb{H}(X) + 1/n$ bits per copy.*

Remark 0.2 (Where the theorem comes from). Shannon (1948) proved that no compression of the output of a source can beat its entropy. Cover and Thomas (2006, sec. 5.3 to 5.5) prove each part of Theorem 0.2. For part 2, they give each value x with $p(x) > 0$ the length $\ell(x) = \lceil -\log_2 p(x) \rceil$, its surprisal in bits rounded up; Kraft's condition (Cover and Thomas 2006, sec. 5.2) guarantees that there is a code with these lengths.

Coding with the wrong distribution

Theorem 0.3 (The cost of coding with the wrong distribution)**.** *Let X take a finite number of values, with probability mass function p, and let q be such a function, possibly different from p, with $q(x) > 0$ wherever $p(x) > 0$. Suppose a code gives each value x with $p(x) > 0$ the length $\ell(x) = \lceil -\log_2 q(x) \rceil$. Then its expected length L under p satisfies*

$$\mathbb{H}(p, q) \leq L < \mathbb{H}(p, q) + 1,$$

where $\mathbb{H}(p, q)$ is the cross-entropy of q relative to p, with base-2 logarithms.

Proof

Proof. Every sum here runs over the values x with $p(x) > 0$. Rounding up adds less than one binary digit, so for each such x

$$-\log_2 q(x) \leq \ell(x) < -\log_2 q(x) + 1.$$

Multiply each part by $p(x) > 0$ and sum. Since $\sum_x p(x) = 1$,

$$-\sum_x p(x) \log_2 q(x) \leq \sum_x p(x)\, \ell(x) < -\sum_x p(x) \log_2 q(x) + 1.$$

The middle sum is L by Theorem 0.1, and the outer sums are $\mathbb{H}(p, q)$. □

Theorem 0.4 (Cross-entropy is entropy plus divergence)**.** *Let p and q be probability mass functions on the values $x_1, \ldots, x_n$, with $q(x) > 0$ wherever $p(x) > 0$. Write $\mathbb{H}(p)$ for the entropy of p, $\mathbb{H}(p, q)$ for the cross-entropy of q relative to p, and $D_{\mathrm{KL}}(p \,\|\, q)$ for the Kullback–Leibler divergence of q from p, all with base-2 logarithms. Then*

$$\mathbb{H}(p, q) = \mathbb{H}(p) + D_{\mathrm{KL}}(p \,\|\, q).$$

Proof

Proof. Every sum here runs over the values x with $p(x) > 0$. For every such x,

$$-\log_2 q(x) = -\log_2 p(x) + \log_2 \frac{p(x)}{q(x)}.$$

Multiply by $p(x)$ and add over x:

$$-\sum_x p(x) \log_2 q(x) = -\sum_x p(x) \log_2 p(x) + \sum_x p(x) \log_2 \frac{p(x)}{q(x)}.$$

Reading left to right, these are $\mathbb{H}(p, q)$, $\mathbb{H}(p)$ and $D_{\mathrm{KL}}(p \,\|\, q)$. □

Example 0.2 (Cross-entropy, entropy and divergence for a uniform code)**.**
Let $p = (1/2, 1/4, 1/4)$ be the probability mass function of a three-valued random variable, and let $q = (1/3, 1/3, 1/3)$ assign $1/3$ to each of the three values. The code below computes the cross-entropy, the entropy and the divergence in bits.

```
p <- c(1 / 2, 1 / 4, 1 / 4)
q <- rep(1 / 3, 3)
cross_entropy <- -sum(p * log2(q))
shannon_entropy <- -sum(p * log2(p))
divergence <- sum(p * log2(p / q))
c(
  cross_entropy = cross_entropy,
  shannon_entropy = shannon_entropy,
  divergence = divergence
)
#>   cross_entropy shannon_entropy      divergence
#>       1.5849625       1.5000000       0.0849625
```

The cross-entropy is $\mathbb{H}(p, q) = 1.585$ bits. The entropy is $\mathbb{H}(p) = 1.5$ bits, and the

divergence is $D_{\mathrm{KL}}(p \,\|\, q) = 0.085$ bits. They satisfy Theorem 0.4: $1.5 + 0.085 = 1.585$.

Remark 0.3 (Cross-entropy is the cost of the wrong code). Theorem 0.4 splits the cross-entropy into the entropy and the divergence. By Theorem 0.2, no uniquely decodable code averages fewer than $\mathbb{H}(p)$ bits. By Theorem 0.3, a code built for q costs about $D_{\mathrm{KL}}(p \,\|\, q)$ bits per value more than that. By the divergence theorem, the excess vanishes only when the code was built for p (Cover and Thomas 2006, sec. 2.6; Yeung 2008, sec. 2.5). Cover and Thomas (2006, sec. 5.4) state Theorem 0.3 in this form, with $\mathbb{H}(p) + D_{\mathrm{KL}}(p \,\|\, q)$ in place of $\mathbb{H}(p, q)$.

Remark 0.4 (Why models are fit by minimizing cross-entropy). Fitting a classifier by minimizing cross-entropy chooses the probabilities q the model assigns so that a code built for q would spend as few bits as possible on the observed labels. By Remark 0.3, that pushes q toward the probabilities the labels came from.
For a single labeled example, let e put all its mass on the observed label. Then $\mathbb{H}(e) = 0$ (as for a coin that always lands heads), so the example's cross-entropy $\mathbb{H}(e, q)$ equals $D_{\mathrm{KL}}(e \,\|\, q)$: every bit it reports is a wasted one.

References

Cover, Thomas M., and Joy A. Thomas. 2006. *Elements of Information Theory*. 2nd ed. Wiley-Interscience. https://doi.org/10.1002/047174882X.

Shannon, C. E. 1948. "A Mathematical Theory of Communication." *Bell System Technical Journal* 27 (3): 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Yeung, Raymond W. 2008. *Information Theory and Network Coding*. Springer. https://doi.org/10.1007/978-0-387-79234-7.