

Entropy and mutual information

Last modified: 2026-10-12 02:31:20 (PDT)

This page builds on the definitions of [entropy](#), [cross-entropy](#) and [Kullback–Leibler divergence](#) in *Probability for Data Science*.

Surprisal

Definition 0.1 (Surprisal). The **surprisal** of an outcome x of a discrete random variable X with $P(X = x) > 0$ is

$$s(x) \stackrel{\text{def}}{=} -\log P(X = x).$$

With base-2 logarithms, surprisal is measured in *bits*; with natural logarithms, in *nats*.

Example 0.1 (Counting bits with fair coins). One toss of a fair coin has 2 equally likely outcomes, and one binary digit is enough to record which one happened. Two tosses have 4 outcomes, and two binary digits record them. In general, k tosses have $n = 2^k$ equally likely outcomes, and $k = \log_2 n$ binary digits record them.

Each of those n outcomes has probability $p = 1/n$, so

$$\begin{aligned} k &= \log_2 n \\ &= \log_2 \frac{1}{p} \\ &= -\log_2 p, \end{aligned}$$

which is the [surprisal](#) of the outcome, in bits.

Ten tosses that all land heads have probability 2^{-10} , so

$$\begin{aligned} s(\text{ten heads}) &= -\log_2 2^{-10} \\ &= 10 \text{ bits.} \end{aligned}$$

Remark 0.1 (Unlikely outcomes carry more information). Surprisal is large when an outcome was unlikely, and it falls to 0 as the outcome becomes certain. Correctly predicting heads on one toss of a fair coin is worth 1 bit; correctly predicting ten heads in a row is worth 10 bits. MacKay (2003, sec. 2.4) calls $s(x)$ in bits the *Shannon information content* of the outcome x .

Remark 0.2 (Entropy is expected surprisal). The **entropy** of a discrete random variable X is the **expectation** of its **surprisal**:

$$\begin{aligned} H(X) &= \mathbb{E}[-\log P(X)] \\ &= \mathbb{E}[s(X)]. \end{aligned}$$

So $H(X)$ is the average number of bits (with base-2 logarithms) needed to record an outcome of X when each outcome x costs $s(x)$ bits. The **source coding theorem**, on a later page of these notes, makes that average precise: no uniquely decodable code does better.

Joint and conditional entropy

Definition 0.2 (Joint entropy). Let X and Y be discrete random variables with joint probability mass function $p(x, y) \stackrel{\text{def}}{=} P(X = x, Y = y)$. Their **joint entropy** is the entropy of the pair (X, Y) :

$$H(X, Y) \stackrel{\text{def}}{=} - \sum_{x, y} p(x, y) \log p(x, y),$$

where the sum runs over the pairs with $p(x, y) > 0$.

Definition 0.3 (Entropy given one value). For X and Y as in Definition 0.2, write $p(y | x) \stackrel{\text{def}}{=} P(Y = y | X = x)$. The **entropy of Y given $X = x$** , for a value x with $P(X = x) > 0$, is

$$H(Y | X = x) \stackrel{\text{def}}{=} - \sum_y p(y | x) \log p(y | x),$$

where the sum runs over the values y with $p(y | x) > 0$.

Definition 0.4 (Conditional entropy). For X and Y as in Definition 0.2, write $p(x) \stackrel{\text{def}}{=} P(X = x)$. The **conditional entropy** of Y given X averages the **entropy of Y given each value of X** over the distribution of X :

$$H(Y | X) \stackrel{\text{def}}{=} \sum_x p(x) H(Y | X = x),$$

where the sum runs over the values x with $p(x) > 0$.

Theorem 0.1 (Chain rule for entropy). *For discrete random variables X and Y ,*

$$H(X, Y) = H(X) + H(Y | X).$$

i Proof

Proof. Use the notation of Definition 0.3 and Definition 0.4, with every sum over the pairs with $p(x, y) > 0$. Since $p(x, y) = p(x) p(y | x)$,

$$\begin{aligned} H(X, Y) &= - \sum_{x, y} p(x, y) \log p(x, y) \\ &= - \sum_{x, y} p(x, y) \log[p(x) p(y | x)] \\ &= - \sum_{x, y} p(x, y) \log p(x) - \sum_{x, y} p(x, y) \log p(y | x) \\ &= - \sum_x \log p(x) \sum_y p(x, y) - \sum_x p(x) \sum_y p(y | x) \log p(y | x) \\ &= - \sum_x p(x) \log p(x) + \sum_x p(x) H(Y | X = x) \\ &= H(X) + H(Y | X). \end{aligned}$$

The fourth line groups the terms by x and uses $p(x, y) = p(x) p(y | x)$ in the second sum; the fifth uses $\sum_y p(x, y) = p(x)$ and the definition of $H(Y | X = x)$. $\square$

Mutual information

Definition 0.5 (Mutual information). The **mutual information** between discrete random variables X and Y with finite entropies is the reduction in the **entropy** of Y from learning X :

$$I(X; Y) \stackrel{\text{def}}{=} H(Y) - H(Y | X),$$

where $H(Y | X)$ is the **conditional entropy**.

Theorem 0.2 (Mutual information is symmetric). *For discrete random variables X and Y with finite entropies,*

$$I(X; Y) = H(X) - H(X | Y) = I(Y; X).$$

i Proof

Proof. The [chain rule](#) expands $H(X, Y)$ in either order:

$$H(X) + H(Y | X) = H(X, Y) = H(Y) + H(X | Y).$$

Subtracting $H(X | Y) + H(Y | X)$ from the outer two expressions gives

$$H(X) - H(X | Y) = H(Y) - H(Y | X).$$

The right side is $I(X; Y)$ by Definition 0.5, and the left side is $I(Y; X)$ by the same definition. □

Remark 0.3 (Mutual information as a divergence). Cover and Thomas (2006, sec. 2.3) define mutual information as the [Kullback–Leibler divergence](#) of the joint distribution of (X, Y) from the product of its marginals, and show that this equals $H(Y) - H(Y | X)$ (Cover and Thomas 2006, sec. 2.4). Since a Kullback–Leibler divergence is never negative, and is 0 only when the two distributions are equal (Cover and Thomas 2006, sec. 2.6), $I(X; Y) \geq 0$, with equality exactly when X and Y are independent. Learning X never raises the entropy of Y on average.

Definition 0.6 (Information gain). A decision tree splits the points at a node into branches. The **information gain** of the split is the entropy of the class labels at the node, minus the entropy of the labels in each branch, weighted by the share of the node’s points that branch holds (Quinlan 1986).

Remark 0.4 (Information gain is mutual information). Draw one point uniformly at random from the node of Definition 0.6, and let Y be its class and X the branch it falls into. Then the entropy at the node is $H(Y)$, the weighted entropy left in the branches is $H(Y | X)$, and the information gain is the [mutual information](#) $I(X; Y)$.

Exercise 0.1 (Choosing between two splits). A node holds 20 points, 10 red and 10 green. Two candidate splits each send the points into two branches:

split	branch	red	green
A	yes	4	0
	no	6	10
B	yes	5	10
	no	5	0

Draw one point uniformly at random from the node, let Y be its color, and let X be the branch it falls into (Remark 0.4). The information gain of each split (Definition 0.6) is then $I(X; Y)$. Use base-2 logarithms throughout.

1. Compute $H(Y)$.
2. For each split, compute $H(Y | X = x)$ for both branches, and then $H(Y | X)$.
3. Compute the information gain $I(X; Y)$ for each split. Which split has the larger gain?

Solution

Solution 0.1.

1. Half the points are red, so

$$\begin{aligned} H(Y) &= -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} \\ &= 1 \text{ bit.} \end{aligned}$$

2. A branch holding one color only has entropy 0. For split A, the “no” branch holds 6 red and 10 green points of 16:

$$\begin{aligned} H(Y | X = \text{no}) &= -\frac{6}{16} \log_2 \frac{6}{16} - \frac{10}{16} \log_2 \frac{10}{16} \\ &\approx 0.531 + 0.424 \\ &\approx 0.954 \text{ bits.} \end{aligned}$$

The “yes” branch holds 4 of the 20 points and the “no” branch 16, so

$$\begin{aligned} H(Y | X) &= \frac{4}{20} \cdot 0 + \frac{16}{20} \cdot 0.954 \\ &\approx 0.764 \text{ bits.} \end{aligned}$$

For split B, the “yes” branch holds 5 red and 10 green points of 15:

$$\begin{aligned}
H(Y \mid X = \text{yes}) &= -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \\
&\approx 0.528 + 0.390 \\
&\approx 0.918 \text{ bits.}
\end{aligned}$$

The “yes” branch holds 15 of the 20 points and the “no” branch 5, so

$$\begin{aligned}
H(Y \mid X) &= \frac{15}{20} \cdot 0.918 + \frac{5}{20} \cdot 0 \\
&\approx 0.689 \text{ bits.}
\end{aligned}$$

3. By Definition 0.5, the gain is $H(Y) - H(Y \mid X)$:

- split A: $1 - 0.764 = 0.236$ bits
- split B: $1 - 0.689 = 0.311$ bits

Split B has the larger gain. Both splits isolate a branch of one color, so the gain depends on the other, mixed branch. B’s mixed branch, 5 red against 10 green, is more lopsided than A’s 6 against 10, so it has less entropy, and it also holds fewer of the points, 15 against 16, so its entropy gets less weight.

References

- Cover, Thomas M., and Joy A. Thomas. 2006. *Elements of Information Theory*. 2nd ed. Wiley-Interscience. <https://doi.org/10.1002/047174882X>.
- MacKay, David J. C. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.
- Quinlan, J. R. 1986. “Induction of Decision Trees.” *Machine Learning* 1 (1): 81–106. <https://doi.org/10.1007/BF00116251>.