Entropy and mutual information

Last modified: 2026-11-01 19:38:09 (PST)

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 $\mathbb{P}(X = x) > 0$ is

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

With base-2 logarithms, surprisal is also called the **Shannon information content** of the outcome x (MacKay 2003, sec. 2.4).

Definition 0.2 (Bit)**.** A **bit** is the unit of information when the logarithm in surprisal, and in every quantity built from it, has base 2.
One bit is the surprisal of an outcome with probability 1/2: for an outcome x of a discrete random variable X,

$$\begin{aligned} s(x) = 1 \text{ bit} &\iff -\log_2 \mathbb{P}(X = x) = 1 \\ &\iff \mathbb{P}(X = x) = 2^{-1} = \tfrac{1}{2}. \end{aligned}$$

In words, a bit is the surprisal of the outcome of one binary trial whose two outcomes are equally likely, such as one toss of a fair coin.

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

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

Definition 0.3 (Nat)**.** A **nat** is the unit of information when the logarithm in surprisal, and in every quantity built from it, is the natural logarithm.
One nat is the surprisal of an outcome with probability $1/\mathrm{e}$: for an outcome x of a discrete random variable X,

$$\begin{aligned} s(x) = 1 \text{ nat} &\iff -\ln \mathbb{P}(X = x) = 1 \\ &\iff \mathbb{P}(X = x) = \mathrm{e}^{-1} = 1/\mathrm{e} \approx 0.368. \end{aligned}$$

Example 0.2 (One fair coin toss in nats)**.** By Example 0.1 with $k = 1$, one toss of a fair coin landing heads has a surprisal of 1 bit. In nats, the same outcome has a surprisal of

$$\begin{aligned} s(\text{heads}) &= -\ln \tfrac{1}{2} \\ &= \ln 2 \\ &\approx 0.693 \text{ nats}. \end{aligned}$$

Remark 0.1 (Converting between bits and nats). For any probability $p \in (0, 1]$, the change-of-base rule for logarithms gives

$$\begin{aligned} -\log_2 p &= \frac{-\ln p}{\ln 2} \\ &\approx 1.443 \times (-\ln p). \end{aligned}$$

So an outcome with a surprisal of 1 nat has a surprisal of $1/\ln 2 \approx 1.443$ bits, and the $\ln 2 \approx 0.693$ nats of Example 0.2 are exactly $\ln 2/\ln 2 = 1$ bit.
A bit as a unit of information is not the same thing as a bit as a binary digit. A binary digit whose two values are equally likely has a surprisal of exactly 1 bit, whichever value it takes. A binary digit that is 0 nine times in ten does not: the value 0 has a surprisal of $-\log_2 0.9 \approx 0.152$ bits, and the value 1 has a surprisal of $-\log_2 0.1 \approx 3.322$ bits.

Remark 0.2 (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.

Remark 0.3 (Entropy is expected surprisal). The entropy of a discrete random variable X is the expectation of its surprisal:

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

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.

Remark 0.4 (Differential entropy differs from entropy). A density value is not a probability, so the surprisal of a continuous value and the entropy of a continuous random variable (the differential entropy) behave differently from their discrete counterparts:

- **Sign.** A density can exceed 1, so the surprisal $s(x)$ and the differential entropy can be negative, as in the uniform example (Cover and Thomas 2006, sec. 8.1). The entropy of a discrete random variable is never negative.
- **Units.** Rescaling changes the differential entropy: $H(aX) = H(X) + \log|a|$ for a constant $a \neq 0$ (Cover and Thomas 2006, Theorem 8.6.4). Measuring the same quantity in metres rather than centimetres therefore changes it. Adding a constant does not: $H(X + c) = H(X)$ (Cover and Thomas 2006, Theorem 8.6.3).
- **Not a limit of entropy.** Round a continuous X to bins of width Δ, which gives a discrete random variable X^{Δ}. Then $H(X^{\Delta}) + \log\Delta \to H(X)$ as $\Delta \to 0$ (Cover and Thomas 2006, Theorem 8.3.1). The entropy $H(X^{\Delta})$ itself grows without bound as $\Delta \to 0$, so the differential entropy is not its limit.

Remark 0.5 (Surprisal and the Fisher information). When the probabilities depend on a parameter θ, the surprisal of an outcome x is $-\log \mathrm{p}(x \mid \theta)$, which is minus the log-likelihood. Its derivative with respect to θ is minus the score. The expected squared score is the expected (Fisher) information, which is also the second derivative, at the true parameter value, of the Kullback–Leibler divergence between the model at that value and the model at the varying value (proved in the sds notes).

Joint and conditional entropy

Definition 0.4 (Joint entropy)**.** Let X and Y be discrete random variables with joint probability mass function $p(x, y) \stackrel{\text{def}}{=} \mathbb{P}(X = x, Y = y)$. 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 0.5 (Entropy given one value)**.** For X and Y as in Definition 0.4, write $p(y \mid x) \stackrel{\text{def}}{=} \mathbb{P}(Y = y \mid X = x)$. The **entropy of Y given $X = x$**, for a value x with $\mathbb{P}(X = x) > 0$, is

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

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

Definition 0.6 (Conditional entropy)**.** For X and Y as in Definition 0.4, write $p(x) \stackrel{\text{def}}{=} \mathbb{P}(X = x)$. The **conditional entropy** of Y given X averages the entropy of Y given each value of X over the distribution of X:

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

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,*

$$H(X, Y) = H(X) + H(Y \mid X).$$

Proof

Proof. Use the notation of Definition 0.5 and Definition 0.6, with every sum over the pairs with $p(x, y) > 0$. Since $p(x, y) = p(x)\, p(y \mid x)$,

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

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

Mutual information

Definition 0.7 (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:

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

where $H(Y\mid X)$ is the conditional entropy.

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

$$I(X;Y) = H(X) - H(X\mid Y) = I(Y;X).$$

Proof

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

$$H(X) + H(Y\mid X) = H(X,Y) = H(Y) + H(X\mid Y).$$

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

$$H(X) - H(X\mid Y) = H(Y) - H(Y\mid X).$$

The right side is $I(X;Y)$ by Definition 0.7, and the left side is $I(Y;X)$ by the same

definition. □

Remark 0.6 (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 $H(Y) - H(Y \mid X)$ (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 0.8 (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 0.7 (Information gain is mutual information). Draw one point uniformly at random from the node of Definition 0.8, 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 $H(Y \mid X)$, 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 0.7). The information gain of each split (Definition 0.8) is then $I(X;Y)$. Use base-2 logarithms throughout.

1. Compute $H(Y)$.
2. For each split, compute $H(Y \mid X = x)$ for both branches, and then $H(Y \mid X)$.
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

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

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:

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

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

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

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

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

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

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

3. By Definition 0.7, 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.