Entropy and perplexity

Herman Kamper

2023-02,

Entropy

Example: The horse race

Perplexity

Entropy and perplexity examples

Cross entropy

Entropy rate

Entropy

If an outcome has a very low probability, that means that outcome carries a lot of information:

- `dog bites man`
- `man bites dog`
- `it snowed in Chicago`
- `it snowed in Cape Town`

The entropy of a random variable is the average level of information or uncertainty over the variable's possible outcomes (Wikipedia).

One way to derive entropy is to list what we want from a definition of information (Peebles, 2001, p. 80):

- Should be large for outcomes with low probability: $\frac{1}{P(x=k)}$

- Information from two independent sources should add.

- Decision: Information should be positive and should be 0 for a certain outcome.

- Logarithm is the only function with these properties: $\log \frac{1}{P(x=k)}$

- Decision: Base 2 since smallest choice is between two quantities.

- So information from $x = k$: $\log_2 \frac{1}{P(x=k)} = -\log_2 P(x = k)$

- Average information over outcomes: $\mathbb{E}\left[-\log_2 P(x)\right]$

The entropy of a discrete random variable x taking on possible outcomes $1, 2, \ldots, K$ is thus defined as

$$H(x) \triangleq -\sum_{k=1}^{K} P(x = k) \log_2 P(x = k)$$

With $\log_2$, entropy is measured in bits (but can also use other bases).

Properties:

- Maximum with most uncertainty:
  Uniform distribution over K outcomes.

- Minimum with least uncertainty:
  All probability mass on one outcome.

- The entropy of a uniform distribution over K outcomes: $\log_2 K$

Information-theoretic meaning:

The average length of the shortest description of a random variable.

Equivalent to:

- Minimum number of bits per outcome (on average) to encode a source.

- Minimum number of yes/no questions (on average) per outcome. Questions can be about more than one category, e.g. "is the outcome one of the categories $\{4, 5, 6, 7\}$?" (see the uniform horse example below).

Example: The horse race

Example from (Cover and Thomas, 2006). We are at a race track and want to send the winning horse of each race over a binary channel. There are eight horses in a race.

Uniform distribution

The probability of winning is equal over the horses, i.e.

$$\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}$$

One optimal encoding:

Horse	Codeword
1	001
2	010
3	011
4	100
5	101
6	110
7	111
8	000

Average number of bits: 3 bits.

Does this match the entropy?

$$H(x) = -\sum_{k=1}^{8} P(x = k) \log_2 P(x = k)$$
$$= -\sum_{k=1}^{8} \frac{1}{8} \log_2 \frac{1}{8}$$
$$= 3 \text{ bits}$$

Non-uniform distribution

Now the probabilities of winning are

$$\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}$$

What is the entropy?

$$
\begin{aligned}
H(x) = & -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} \\
& -\frac{1}{64}\log_2\frac{1}{64} - \frac{1}{64}\log_2\frac{1}{64} - \frac{1}{64}\log_2\frac{1}{64} - \frac{1}{64}\log_2\frac{1}{64} \\
= & \ 2 \text{ bits}
\end{aligned}
$$

Encoding achieving this:

Horse	Codeword
1	0
2	10
3	110
4	1110
5	111100
6	111101
7	111110
8	111111

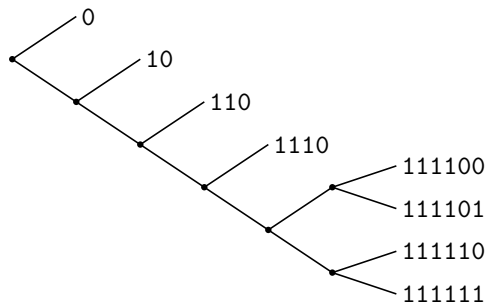Example of a prefix code: No codeword is a prefix of any other.

Outcome trees

The above example also illustrates that entropy is the minimum number of yes/no questions (on average) needed to transmit an outcome.

Yes/no questions for the uniform distribution:
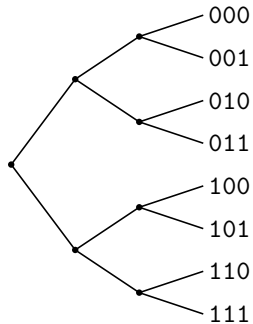
Yes/no questions for the non-uniform distribution:

Perplexity

Perplexity can be seen as the weighted number of choices we have to make for a random discrete variable x:

$$\text{PP}(x) \triangleq 2^{H(x)}$$

Example: The horse race

The perplexity for the two cases in the horse race:

- Uniform distribution: $\text{PP}(x) = 2^3 = 8$

- Non-uniform distribution: $\text{PP}(x) = 2^2 = 4$

In the examples below, ask yourself how many outcomes are you really deciding between.

Entropy and perplexity examples

Single outcome:

$$P(x = a) = 1$$

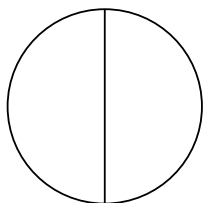$$\begin{aligned} H(x) &= -1 \log_2 1 \\ &= 0 \text{ bits} \\ \text{PP}(x) &= 2^0 \\ &= 1 \end{aligned}$$
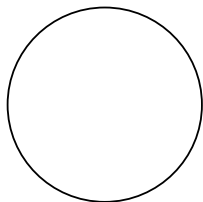
Two equally likely outcomes:

$$P(x = a) = 0.5$$
$$P(x = b) = 0.5$$

$$\begin{aligned} H(x) &= -0.5 \log_2 0.5 - 0.5 \log_2 0.5 \\ &= 1 \text{ bit} \\ \text{PP}(x) &= 2^1 \\ &= 2 \end{aligned}$$
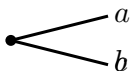
Four equally likely outcomes:
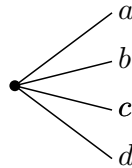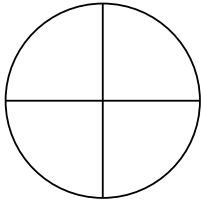
$$P(x = a) = 0.25 \qquad H(x) = 2 \text{ bits}$$
$$P(x = b) = 0.25 \qquad \text{PP}(x) = 4$$
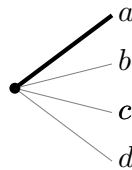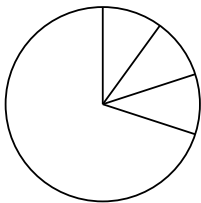$$P(x = c) = 0.25$$
$$P(x = d) = 0.25$$

Four non-uniform outcomes:

$$P(x = a) = 0.7 \qquad H(x) = -0.7 \log_2 0.7 - 3 \cdot 0.1 \log_2 0.1$$
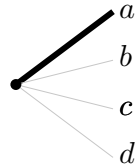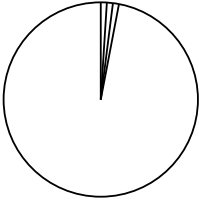$$P(x = b) = 0.1 \qquad\qquad = 1.35678 \text{ bits}$$
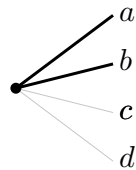$$P(x = c) = 0.1 \qquad \text{PP}(x) = 2^{1.35678}$$
$$P(x = d) = 0.1 \qquad\qquad = 2.5611$$

Four non-uniform outcomes:

$$P(x = a) = 0.97 \qquad H(x) = 0.2419 \text{ bits}$$
$$P(x = b) = 0.01 \qquad \text{PP}(x) = 1.1826$$
$$P(x = c) = 0.01$$
$$P(x = d) = 0.01$$

Four non-uniform outcomes:

$$P(x = a) = 0.49 \qquad H(x) = 1.1414 \text{ bits}$$
$$P(x = b) = 0.49 \qquad \text{PP}(x) = 2.2060$$
$$P(x = c) = 0.01$$
$$P(x = d) = 0.01$$
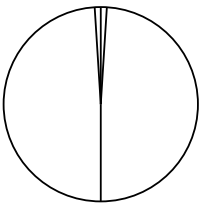
Entropy of uniform distribution over K outcomes is $\log_2 K$:

$$H(x) = 0 \qquad H(x) = 1 \qquad H(x) = 2$$
$$\mathrm{PP}(x) = 1 \qquad \mathrm{PP}(x) = 2 \qquad \mathrm{PP}(x) = 4$$
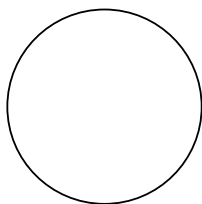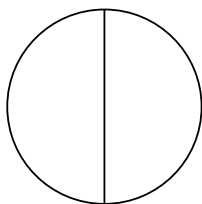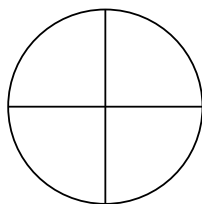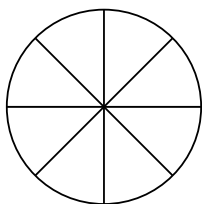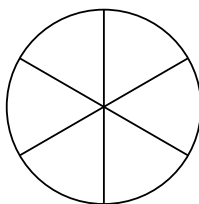
$$H(x) = 3 \qquad H(x) = 2.5850$$
$$\mathrm{PP}(x) = 8 \qquad \mathrm{PP}(x) = 6$$

Any non-uniform distribution over K outcomes has lower entropy than the corresponding uniform distribution:
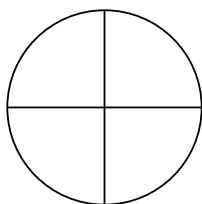
$$H(x) = 2 \qquad H(x) = 1.35678 \qquad H(x) = 0.2419$$
$$\mathrm{PP}(x) = 4 \qquad \mathrm{PP}(x) = 2.5611 \qquad \mathrm{PP}(x) = 1.1826$$

Cross entropy

We have two discrete distributions both over possible outcomes $1, 2, \ldots, K$. The masses of the one distribution is denoted as $\mathbf{p}$ and the other as $\mathbf{q}$. The cross entropy is then defined as

$$H(\mathbf{p}, \mathbf{q}) \triangleq - \sum_{k=1}^{K} P_{\mathbf{p}}(x = k) \log_2 P_{\mathbf{q}}(x = k)$$

$$= - \sum_{k=1}^{K} p_k \log_2 q_k$$

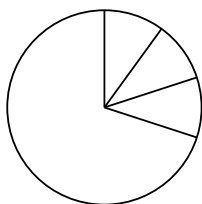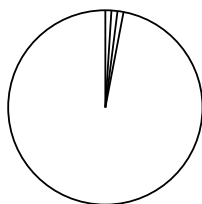The cross entropy is the minimum number of bits on average needed to encode outcomes coming from source $\mathbf{p}$ when we use another model $\mathbf{q}$ to construct the codebook.

The cross entropy is an upper bound on the entropy of the source:

$$H(\mathbf{p}) \leq H(\mathbf{p}, \mathbf{q})$$

The closer model $\mathbf{q}$ is to source $\mathbf{p}$, the closer the cross entropy will be to the entropy. Stated differently, the better the model, the lower the cross entropy.

Side note: Information theory and machine learning

- In information theory the goal is to get a good model of the unknown source $P(x)$ so that we can encode x with the shortest code.

- In machine learning, the goal is to get a good model of the unknown real-world distribution $P(x)$ so that we can use it to make predictions for new x.

Entropy rate

We've looked at the entropy of a single variable. What about sequences?

The entropy rate for a random process generating sequences X is defined as

$$
\begin{aligned}
H(X) &= \lim_{T \to \infty} -\frac{1}{T} H(x_{1:T}) \\
&= \lim_{T \to \infty} -\frac{1}{T} \sum_{x_{1:T}} P(x_{1:T}) \log_2 P(x_{1:T})
\end{aligned}
$$

We normally don't actually know the real $P(x_{1:T})$. If we have a model $\boldsymbol{\theta}$, then we can calculate the cross entropy rate:

$$
H(\mathbf{p}, \boldsymbol{\theta}) = \lim_{T \to \infty} -\frac{1}{T} \sum_{x_{1:T}} P_{\mathbf{p}}(x_{1:T}) \log_2 P_{\boldsymbol{\theta}}(x_{1:T})
$$

using $\mathbf{p}$ to explicitly denote the real-world (unknown) distribution.

We still can't calculate the above, since we don't have infinite sequences. So we estimate the cross entropy:

$$
H(\mathbf{p}, \boldsymbol{\theta}) \approx -\frac{1}{T} \log_2 P_{\boldsymbol{\theta}}(x_{1:T}) = H(x_{1:T}, \boldsymbol{\theta})
$$

where $x_{1:T}$ is a long sample from $P_{\mathbf{p}}(x_{1:T})$, i.e. $x_{1:T} \sim P_{\mathbf{p}}(x_{1:T})$, and $H(x_{1:T}, \boldsymbol{\theta})$ is the notation we use for the estimated cross entropy of model $\boldsymbol{\theta}$.[1]

---

[1] I've really gone a bit crazy in how I overloaded H to mean different things: $H(x)$ for entropy, $H(X)$ for entropy rate, $H(\mathbf{p}, \mathbf{q})$ for cross entropy, and $H(x_{1:T}, \boldsymbol{\theta})$ for estimated cross entropy. Maybe I should have just written H for all of these and hope that the context is enough. Sometimes the term "entropy" is also used interchangeably for all these things.

This estimated cross entropy is actually what we use to evaluate a language model $\boldsymbol{\theta}$ on some test data $x_{1:T}$. In that case we report the perplexity:

$$\begin{aligned} \mathrm{PP} &= 2^{H(x_{1:T}, \boldsymbol{\theta})} \\ &= 2^{-\frac{1}{T} \log_2 P_{\boldsymbol{\theta}}(x_{1:T})} \\ &= P_{\boldsymbol{\theta}}(x_{1:T})^{-\frac{1}{T}} \end{aligned}$$

Side note: Cross-entropy estimate as a Monte Carlo sample

The jump between the equation for cross entropy $H(\mathbf{p}, \boldsymbol{\theta})$ and its estimate $H(x_{1:T}, \boldsymbol{\theta})$ is similar to how we approximate expected values with Monte Carlo. Expected values can be approximated (Resnik and Hardisty, 2010):

$$\mathbb{E}_{p(x)}\left[f(x)\right] \approx \frac{1}{L} \sum_{l=1}^{L} f(x^{(l)})$$

where $x^{(l)} \sim p(x)$ are samples from $p(x)$. With $L = 1$, i.e. a single sample:

$$\mathbb{E}_{p(x)}\left[f(x)\right] \approx f(x^{(1)})$$

The entropy rate of written English

By using human participants, Shannon (1951) estimated the per-letter entropy rate of written English:

$$0.6 \leq H(x_{1:T}) \leq 1.3$$

Experiment and bound (roughly):

- Subjects are presented with English text and asked to predict the guess of the next letter (out of 27).

- Used letters rather than words, since sometimes a subject had to do an exhaustive (27-character) search.

- Record the number of guesses to get the correct letter.

- Obtained a bound by proving how the number of guesses (a different random variable) is related to entropy of English.

The estimate is probably low because he used a single text.

But still: What do these estimates imply when thinking of entropy as the shortest code in bits (yes/no questions), or perplexity as the weighted branching factor?

Prediction and Entropy of Printed English

By C. E. SHANNON

A new method of estimating the entropy and redundancy of a language is described. This method exploits the knowledge of the language statistics possessed by those who speak the language, and depends on experimental results in prediction of the next letter when the preceding text is known. Results of experiments in prediction are given, and some properties of an ideal predictor are developed.

Further reading

For a formal derivation for why entropy is the average length of the shortest description of a random variable, see Sec. 5.2 and Sec. 5.3 in (Cover and Thomas, 2006). This is also just a very accessible textbook.

Huffman codes (Cover and Thomas, 2006, Sec. 5.6) give a way to construct optimal codewords for a given distribution.

Acknowledgements

References

T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., 2006.

P. Z. Peebles, *Probability, Random Variables and Random Signal Principles*, 4th ed., 2001.

P. Resnik and E. Hardisty, "Gibbs sampling for the uninitiated," *University of Maryland*, 2010.

C. E. Shannon, "Prediction and entropy of printed English," *Bell System Technical Journal*, 1951.