# Minimum description length

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

A model that fits the data well gives the data a short code, but a model that is itself long to describe costs bits of its own. This page counts both costs in bits, and chooses the model that makes the sum shortest.

## Description length

**Definition 0.1** (Two-part description length)**.** Let $\underset{\sim}{x}$ be data, and let $p_1, \dots, p_m$ be models, each of which gives $\underset{\sim}{x}$ a probability $p_j(\underset{\sim}{x})$. Let a prefix code on the models give each model $p_j$ a codeword of $\ell(p_j)$ bits.
The **two-part description length** of $\underset{\sim}{x}$ under a model $p_j$ is

$$\ell(p_j) + \mathtt{s}_{p_j}(\underset{\sim}{x}).$$

The first part is the length of the codeword that names the model. The second part is the surprisal $\log_2 \frac{1}{p_j(\underset{\sim}{x})}$ of $\underset{\sim}{x}$ under the model, which is the length in bits of a code for $\underset{\sim}{x}$ built from the model.

**Example 0.1** (The two-part description lengths of coin flips)**.**
A coin is flipped 100 times, and it lands heads 70 times. The chunk below follows Definition 0.1 with nine models, the Bernoulli models with the probability of heads $0.1, 0.2, \dots, 0.9$. It gives each model a codeword of the same length, and it prints the two parts and their sum for each model.

```r
n_flips <- 100
n_heads <- 70
success_prob <- seq(0.1, 0.9, by = 0.1)
model_bits <- log2(length(success_prob))
heads_bits <- n_heads * log2(success_prob)
tails_bits <- (n_flips - n_heads) * log2(1 - success_prob)
data_bits <- -(heads_bits + tails_bits)
total_bits <- model_bits + data_bits
data.frame(success_prob, model_bits, data_bits, total_bits)
#>   success_prob model_bits data_bits total_bits
#> 1          0.1   3.169925 237.09506  240.26498
#> 2          0.2   3.169925 172.19281  175.36273
#> 3          0.3   3.169925 137.02479  140.19471
#> 4          0.4   3.169925 114.64393  117.81386
#> 5          0.5   3.169925 100.00000  103.16993
#> 6          0.6   3.169925  91.24543   94.41536
#> 7          0.7   3.169925  88.12909   91.29901
#> 8          0.8   3.169925  92.19281   95.36273
#> 9          0.9   3.169925 110.29806  113.46798
```

Each model takes 3.17 bits to name, because nine equally long codewords need $\log_2 9$ bits. The data part is smallest, at 88.1 bits, for the model with probability of heads 0.7.

**Definition 0.2** (Minimum description length principle)**.** Given data $\underset{\sim}{x}$ and models $p_1, \ldots, p_m$, the **minimum description length principle** chooses the model with the smallest two-part description length of $\underset{\sim}{x}$. The models may have codewords of different lengths.

**Example 0.2** (The minimum description length model for coin flips)**.**
The chunk below applies Definition 0.2 to the two-part description lengths of Example 0.1. It prints the model with the smallest two-part description length, and it prints the length of writing the 100 flips, with 70 heads, out as plain heads and tails.

```r
chosen <- which.min(total_bits)
success_prob[chosen]
#> [1] 0.7
total_bits[chosen]
#> [1] 91.29901
n_flips
#> [1] 100
```

The principle chooses the model with the probability of heads 0.7, with a two-part length of 91.3 bits. That is shorter than the 100 bits of the flips themselves, because the flips are far from fair.

**Theorem 0.1** (Minimum description length and maximum likelihood)**.** *Let $\underset{\sim}{x}$ be data, and let $p_1, \ldots, p_m$ be models that give $\underset{\sim}{x}$ the probabilities $p_j(\underset{\sim}{x}) > 0$, with the same codeword length $\ell(p_j) = \ell_0$ for every model. Then a model has the smallest two-part description length of $\underset{\sim}{x}$ exactly when it is a maximum likelihood estimate of the model given $\underset{\sim}{x}$.*

Proof

*Proof.* Every model has the first part $\ell_0$, so the models differ only in the second part $\mathbb{s}_{p_j}(\underset{\sim}{x})$. The surprisal $\log_2 \frac{1}{u}$ decreases as the probability $u$ increases. So the second part is smallest for the model with the largest $p_j(\underset{\sim}{x})$, which is a maximum likelihood estimate. □

**Example 0.3** (Equal codeword lengths for the coin flips)**.**
The 9 models of Example 0.1 have codewords of equal length, so Theorem 0.1 applies to the data of 70 heads in 100 flips. The chunk below compares the model with the shortest two-part description length to the model that gives the flips the highest probability.

```
likelihood <- success_prob^n_heads * (1 - success_prob)^(n_flips - n_heads)
success_prob[which.max(likelihood)]
#> [1] 0.7
success_prob[which.min(total_bits)]
#> [1] 0.7
```

Both models have the probability of heads 0.7, as the theorem says.

*Remark* 0.1 (Where the minimum description length principle comes from). Cover and Thomas (2006, sec. 14.13) state the principle as the choice of the model such that the model's length plus the length of the data given the model is as short as possible. They give the model's length as its Kolmogorov complexity, not as the length of a codeword from a prefix code.

## References

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