Information Theory I: Entropy, Cross-Entropy, KL

Surprisal · Source Coding · Cross-Entropy · KL Divergence

Why Information Theory?

Founded by **Claude Shannon** (1948):
"A Mathematical Theory of Communication"

Core question: how do we **measure** information?

In ML, information theory gives us:

loss functions · model selection · feature selection · compression

We'll develop three key concepts step by step:

Entropy $\rightarrow$ **Cross-Entropy** $\rightarrow$ **KL Divergence**

Entropy

How much **surprise** does a random variable carry?

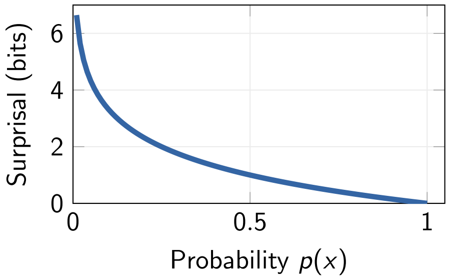
Can we quantify **uncertainty**?

Surprisal: Rare Events Are More Informative

Observing a **rare** event is more "surprising" than observing a common one.

Surprisal (self-information) of outcome x:

$$I(x) = -\log_2 p(x) = \log_2 \frac{1}{p(x)} \qquad \text{(measured in \textbf{bits})}$$



Properties:

- $I(x) \geq 0$ always
- $p(x) = 1 \Rightarrow I(x) = 0$ (certain = no surprise)
- $p(x) \to 0 \Rightarrow I(x) \to \infty$ (rare = very surprising)
- **Additive:** $I(x, y) = I(x) + I(y)$ for independent events

Shannon Entropy = Expected Surprisal

We can't predict which outcome we'll see, so we take the **average** surprisal:

Shannon Entropy:

$$H(X) \quad := \quad \mathbb{E}[-\log_2 p(X)] \quad = \quad -\sum_x p(x)\, \log_2 p(x)$$

Convention: $0 \cdot \log_2 0 \;=\; 0$ (justified by $\lim_{t\to 0^+} t \log t \;=\; 0$)

$\log_2 \;\to$ **bits**; $\ln \to$ **nats** (1 nat $= \log_2 e \approx 1.44$ bits). ML often uses nats.

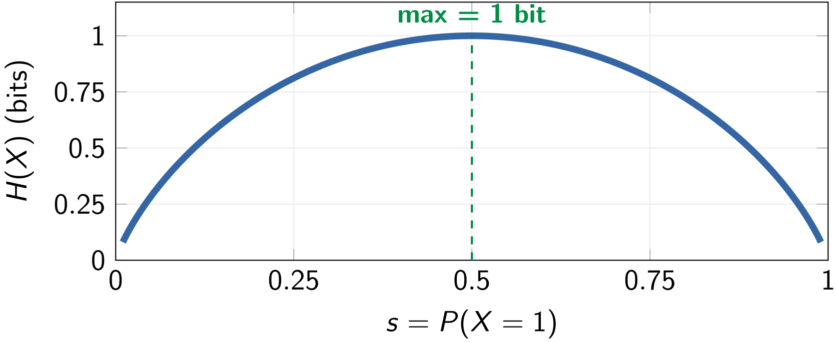
Worked example: X takes values $\{a, b, c, d\}$ with $p = \left(\frac{1}{2},\ \frac{1}{4},\ \frac{1}{8},\ \frac{1}{8}\right)$.

$$\begin{aligned} H(X) &= -\left[\tfrac{1}{2}\log_2 \tfrac{1}{2} + \tfrac{1}{4}\log_2 \tfrac{1}{4} + \tfrac{1}{8}\log_2 \tfrac{1}{8} + \tfrac{1}{8}\log_2 \tfrac{1}{8}\right] \\ &= \tfrac{1}{2}\cdot 1 + \tfrac{1}{4}\cdot 2 + \tfrac{1}{8}\cdot 3 + \tfrac{1}{8}\cdot 3 = \boxed{\tfrac{7}{4} = 1.75 \text{ bits}} \end{aligned}$$

Entropy of the Bernoulli Distribution

For $X \sim \mathrm{Bern}(s)$: $P(X{=}1) = s$, $P(X{=}0) = 1 - s$.

$$H(X) = -s\log_2 s - (1-s)\log_2(1-s)$$



$H(0) = H(1) = 0$ (deterministic). $H(0.5) = 1$ bit (fair coin: maximum uncertainty).

Properties of Entropy

#	Property	Formula
1	**Non-negative**	$H(X) \geq 0$
2	**Zero for deterministic**	$H(X) = 0$ iff one $p(x) = 1$
3	**Continuous** in probabilities	small $\Delta p \Rightarrow$ small ΔH
4	**Symmetric** in p values	relabeling outcomes doesn't change H
5	**Additive** for independent RVs	$H(X, Y) = H(X) + H(Y)$
6	**Maximal for uniform**	$H(X) \leq \log_2 g \quad (g = \#\text{outcomes})$

Uniqueness (Khinchin, 1957): Shannon entropy is the **only** function satisfying properties 1–5 (up to a constant). There is no other sensible measure of uncertainty!

Entropy Is Maximal for Uniform Distributions

Claim: Among all distributions on g outcomes, the uniform maximizes entropy.

Proof (Lagrange multipliers): Maximize $H = -\sum_{i=1}^{g} p_i \log_2 p_i$ subject to $\sum p_i = 1$.

$$\mathcal{L} = -\sum_{i=1}^{g} p_i \log_2 p_i \; - \; \lambda\left(\sum_{i=1}^{g} p_i - 1\right)$$

Entropy Is Maximal for Uniform Distributions

Claim: Among all distributions on g outcomes, the uniform maximizes entropy.

Proof (Lagrange multipliers): Maximize $H = -\sum_{i=1}^{g} p_i \log_2 p_i$ subject to $\sum p_i = 1$.

$$\mathcal{L} = -\sum_{i=1}^{g} p_i \log_2 p_i \;-\; \lambda\left(\sum_{i=1}^{g} p_i - 1\right)$$

$$\frac{\partial \mathcal{L}}{\partial p_i} = -\log_2 p_i - \frac{1}{\ln 2} - \lambda = 0 \quad\Rightarrow\quad \log_2 p_i = \text{const} \quad\Rightarrow\quad \boxed{p_i = \frac{1}{g} \;\text{ for all } i}$$

$$H_{\max} = -\sum_{i=1}^{g} \frac{1}{g} \log_2 \frac{1}{g} = \log_2 g$$

Intuition: The uniform distribution is the "most uncertain" — it makes no assumptions about which outcome is more likely.

More outcomes $\Rightarrow$ higher maximum entropy, but with diminishing returns.

Source Coding

Entropy measures uncertainty. But what IS information, **physically**?

Source coding gives a concrete answer: **bits**.

The Coding Problem

A source produces symbols from $\mathcal{X} = \{\text{dog, cat, fish, bird}\}$. We want to encode them as **binary strings** for transmission.

Fixed-length code: Each symbol gets a codeword of the same length.

Symbol	Probability	Codeword	Length
dog	1/2	00	2 bits
cat	1/4	01	2 bits
fish	1/8	10	2 bits
bird	1/8	11	2 bits

$$\mathbb{E}[L] = \tfrac{1}{2} \cdot 2 + \tfrac{1}{4} \cdot 2 + \tfrac{1}{8} \cdot 2 + \tfrac{1}{8} \cdot 2 = \textbf{2 bits per symbol}$$

But "dog" appears **half** the time — shouldn't it get a **shorter** code?

Variable-Length Codes and the Prefix Property

Idea: Shorter codes for more probable symbols, longer for less probable.

Problem: Ambiguity! If dog→0, cat→1, fish→01, bird→11, then 01 could be "dog, cat" or "fish" — we can't tell!

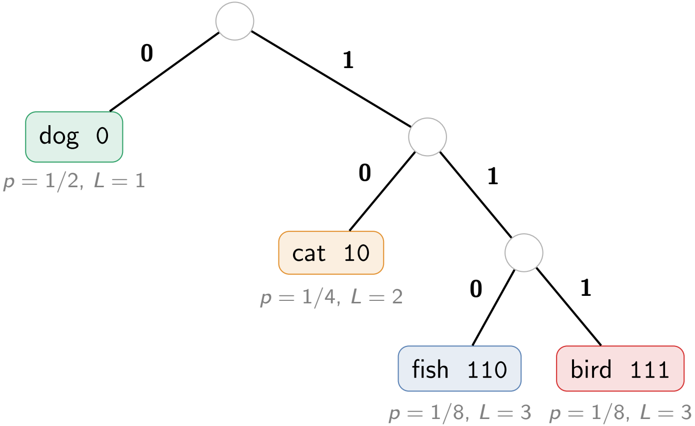
Prefix property: No codeword is a prefix of another codeword.

Guarantees **unambiguous decoding** — read left-to-right, always know where each codeword ends.

Solution — a valid prefix code:

Symbol	**Prob.** $p(x)$	**Codeword**	**Length** $L(x)$	**Surprisal** $-\log_2 p(x)$
dog	$1/2$	0	1	1
cat	$1/4$	10	2	2
fish	$1/8$	110	3	3
bird	$1/8$	111	3	3

Prefix Code as a Binary Tree



Shorter paths (fewer bits) for more probable symbols.
Each leaf is a codeword; the prefix property is guaranteed by the tree structure.

Optimal Code Length Equals Entropy

Key observation: In our prefix code, the code length of each symbol equals its surprisal!

$$L(x) = -\log_2 p(x) \qquad \text{for every symbol } x$$

Optimal Code Length Equals Entropy

Key observation: In our prefix code, the code length of each symbol equals its surprisal!

$$L(x) = -\log_2 p(x) \qquad \text{for every symbol } x$$

The **expected code length**:

$$\mathbb{E}[L(X)] = \sum_x p(x)\, L(x) = \sum_x p(x) \cdot \left(-\log_2 p(x)\right)$$

$$= \tfrac{1}{2} \cdot 1 + \tfrac{1}{4} \cdot 2 + \tfrac{1}{8} \cdot 3 + \tfrac{1}{8} \cdot 3 = \mathbf{1.75\ bits}$$

$$= H(X) \quad \checkmark$$

The optimal prefix code assigns $-\log_2 p(x)$ bits to symbol x.

The **average code length** of this optimal code is exactly **the entropy** $H(X)$.

Compared to fixed-length (2 bits): we save $2 - 1.75 = 0.25$ bits per symbol!

Shannon's Source Coding Theorem

Noiseless Coding Theorem (Shannon, 1948):

For any source X with entropy $H(X)$:

(1) No prefix code can achieve $\mathbb{E}[L] \quad < \quad H(X)$.

(2) There exists a prefix code with $\mathbb{E}[L] \quad < \quad H(X) + 1$.

Entropy = the fundamental limit of lossless compression.

If you try to use fewer bits on average, you **must** lose information.

In practice: **Huffman coding** achieves near-optimal code lengths.

This gives entropy a physical meaning: $H(X) =$ minimum average bits needed to describe X.

Cross-Entropy

What happens when we use the **wrong** code?

Using the Wrong Codebook

The true distribution is p, but we **think** the distribution is q (and design our code for q).

Symbol	$p(x)$	$q(x)$	$L_p(x) = -\log_2 p$	$L_q(x) = -\log_2 q$	**Waste**
dog	1/2	1/4	1	2	$+1$
cat	1/4	1/4	2	2	0
fish	1/8	1/4	3	2	-1
bird	1/8	1/4	3	2	-1

Expected length with the **right** code (for p): $\mathbb{E}_p[L_p] = H(p) = 1.75$ bits.
Expected length with the **wrong** code (for q): $\mathbb{E}_p[L_q] = \frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 2 = 2$ bits.

Using the wrong code wastes $2 - 1.75 = 0.25$ bits per symbol on average.
The wrong code is **always at least as long** as the right one.

Cross-Entropy: Definition

Cross-Entropy of p relative to q:

$$H(p\|q) \quad = \quad -\sum_{x} p(x)\,\log_2 q(x) \quad = \quad \mathbb{E}_{X\sim p}[-\log_2 q(X)]$$

"Average code length when data comes
from p but we use the optimal code for q."

$H(p\|p) = H(p)$ (right code = entropy)

$H(p\|q) \geq H(p)$ (wrong code always wastes bits)

$H(p\|q) \neq H(q\|p)$ (**not symmetric!**)

KL Divergence

How many bits do we **waste** by using the wrong code?

From Cross-Entropy to KL Divergence

The **gap** between cross-entropy and entropy measures the wasted bits:

$$
\begin{aligned}
\underbrace{H(p\|q)}_{\text{wrong code}} - \underbrace{H(p)}_{\text{right code}} &= -\sum_x p(x)\log_2 q(x) - \left(-\sum_x p(x)\log_2 p(x)\right) \\
&= \sum_x p(x)\left[\log_2 p(x) - \log_2 q(x)\right] \\
&= \sum_x p(x)\ \log_2 \frac{p(x)}{q(x)}
\end{aligned}
$$

From Cross-Entropy to KL Divergence

The **gap** between cross-entropy and entropy measures the wasted bits:

$$\underbrace{H(p\|q)}_{\text{wrong code}} - \underbrace{H(p)}_{\text{right code}} = -\sum_x p(x)\log_2 q(x) - \left(-\sum_x p(x)\log_2 p(x)\right)$$

$$= \sum_x p(x)\left[\log_2 p(x) - \log_2 q(x)\right]$$

$$= \sum_x p(x)\,\log_2 \frac{p(x)}{q(x)}$$

Kullback–Leibler Divergence:

$$D_{\mathrm{KL}}(p\|q) \quad = \quad \sum_x p(x)\,\log_2 \frac{p(x)}{q(x)} \quad = \quad \mathbb{E}_{X\sim p}\left[\log_2 \frac{p(X)}{q(X)}\right]$$

"Average number of **extra bits** when using q instead of p."

The Fundamental Identity

$$\boxed{H(p\|q) \;=\; H(p) \;+\; D_{\mathrm{KL}}(p\|q)}$$

$H(p\|q)$ = **cross-entropy**

$$\overbrace{\underbrace{H(p)}_{\text{entropy (optimal bits)}} \quad \underbrace{D_{\mathbf{KL}}(p\|q)}_{\text{extra bits (waste)}}}^{}$$

Since $D_{\mathrm{KL}}(p\|q) \geq 0$ (we'll prove this next), cross-entropy **always** exceeds entropy:

$H(p\|q) \geq H(p)$, with equality iff $p = q$.

Information Inequality: $D_{\mathrm{KL}} \geq 0$

Gibbs' Inequality: $D_{\mathrm{KL}}(p\|q) \geq 0$, with equality iff $p = q$.

Proof (via Jensen's inequality, since log is concave):

$$
\begin{aligned}
-D_{\mathrm{KL}}(p\|q) &= \sum_x p(x) \log \frac{q(x)}{p(x)} && \text{(flip the ratio)} \\
&\leq \log\left(\sum_x p(x) \cdot \frac{q(x)}{p(x)}\right) && (\text{Jensen: } \mathbb{E}[\log Z] \leq \log \mathbb{E}[Z]) \\
&= \log\left(\sum_x q(x)\right) = \log 1 = 0 \\
&\Rightarrow \quad D_{\mathrm{KL}}(p\|q) \geq 0
\end{aligned}
$$

Information Inequality: $D_{\mathrm{KL}} \geq 0$

Gibbs' Inequality: $D_{\mathrm{KL}}(p\|q) \geq 0$, with equality iff $p = q$.

Proof (via Jensen's inequality, since log is concave):

$$
\begin{aligned}
-D_{\mathrm{KL}}(p\|q) &= \sum_x p(x) \log \frac{q(x)}{p(x)} && \text{(flip the ratio)} \\
&\leq \log\left(\sum_x p(x) \cdot \frac{q(x)}{p(x)}\right) && \text{(Jensen: } \mathbb{E}[\log Z] \leq \log \mathbb{E}[Z]\text{)} \\
&= \log\left(\sum_x q(x)\right) = \log 1 = 0
\end{aligned}
$$

$$\Rightarrow \quad D_{\mathrm{KL}}(p\|q) \geq 0$$

Equality iff $q(x)/p(x)$ is constant $\forall\, x$ (strict concavity of log), i.e., $p = q$.

The most fundamental inequality in information theory.

You can never do better than the optimal code. Using any other distribution wastes bits.

KL Divergence Is Not a Distance

Despite measuring “closeness,” D_{KL} is **not a distance**:

Property	**True distance?**	**KL?**
Non-negativity: $d(p, q) \geq 0$	✓	✓
Identity: $d(p, q) = 0 \Leftrightarrow p = q$	✓	✓
Symmetry: $d(p, q) = d(q, p)$	✓	×
Triangle inequality	✓	×

Example: $p = \mathrm{Bern}(0.1)$, $q = \mathrm{Bern}(0.5)$.

$$D_{\mathrm{KL}}(p\|q) \approx 0.53 \text{ bits} \quad \neq \quad D_{\mathrm{KL}}(q\|p) \approx 0.74 \text{ bits}$$

KL is a **divergence**, not a distance. The asymmetry will matter a lot in ML!

Three Interpretations of KL Divergence

1. Extra bits: Average number of extra bits when coding data from p using the optimal code for q instead of p.

2. Expected log-ratio: $D_{\mathrm{KL}}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right]$. How "distinguishable" are p and q on average, when data comes from p?

3. Expected evidence (likelihood ratio): In hypothesis testing, $H_0 : q$ vs $H_1 : p$, each observation provides $D_{\mathrm{KL}}(p\|q)$ nats of evidence on average in favor of the truth p.

All three interpretations say the same thing: D_{KL} measures **how different q is from p, as seen by p.**

Differential Entropy (Brief)

For **continuous** RVs with density $f(x)$, entropy generalizes to:

$$h(X) = -\int f(x) \log f(x)\, dx$$

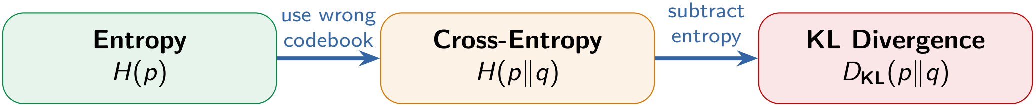
Distribution	**Differential entropy $h(X)$**	**Depends on**
Uniform$[0, a]$	$\log a$	support width
$N(\mu, \sigma^2)$	$\frac{1}{2}\log(2\pi e\, \sigma^2)$	variance only

Warning: Differential entropy can be **negative**!
E.g., Uniform$[0, 1/2]$: $h(X) = \log(1/2) = -1$ bit.

It is **not** invariant to coordinate changes. Use with care.

The KL divergence $D_{\mathrm{KL}}(p\|q) = \int p \log \frac{p}{q}$ remains well-behaved for continuous distributions.

The Big Picture



$$\underbrace{H(p\|q)}_{\text{cross-entropy}} = \underbrace{H(p)}_{\text{entropy}} + \underbrace{D_{\mathrm{KL}}(p\|q)}_{\text{extra bits}}$$

Next lecture: these three concepts give us loss functions, MLE, and more.

Homework

1. Compute $H(X)$ for $X \sim \text{Bernoulli}(1/3)$. Express the answer in bits.
2. Use the information inequality ($D_{\text{KL}} \geq 0$) to give a one-line proof that $H(X) \leq \log_2 g$ for any distribution on g outcomes.
   Hint: Let q be the uniform distribution on g outcomes.
3. Let $p = (1/4,\ 1/4,\ 1/4,\ 1/4)$ and $q = (1/2,\ 1/4,\ 1/8,\ 1/8)$.
   (a) Compute $H(p)$, $H(q)$, $H(p\|q)$, $H(q\|p)$, $D_{\text{KL}}(p\|q)$, $D_{\text{KL}}(q\|p)$.
   (b) Verify $H(p\|q) = H(p) + D_{\text{KL}}(p\|q)$ for both directions.
4. Design a Huffman code for the source $\{A, B, C, D, E\}$ with probabilities $(0.4,\ 0.2,\ 0.2,\ 0.1,\ 0.1)$. Compute the expected code length and compare with $H(X)$.

Questions?