Information Theory

Data Intelligence and Learning (DIAL) Lab
Prof. Jongwuk Lee

Information Entropy

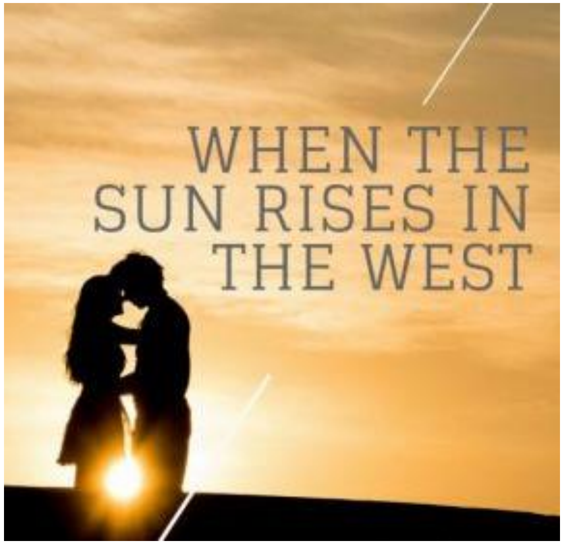
What is Information?

- **A low-probability event carries more information ("surprisal") than a high-probability event.**
- **A lower-probability outcome yields surprising information.**

- **Example: which is more surprising?**
  - When the sun rises in the **east**

    vs.

  - When the sun rises in the **west**

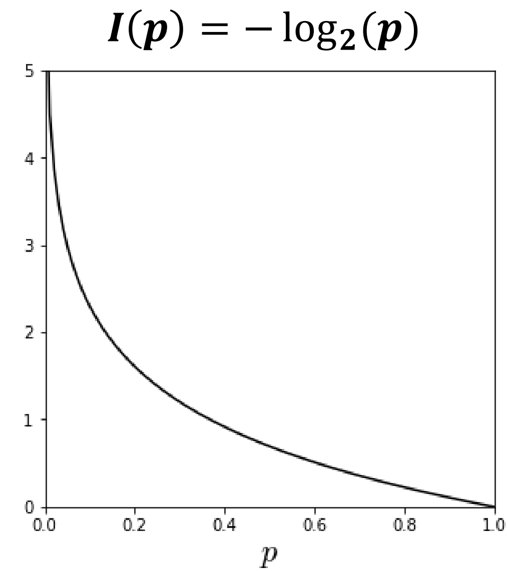


Measuring Information

➢ Definition

$$I(X = x_i) = \log_2\left(\frac{1}{p(x_i)}\right) = -\log_2(p(x_i))$$

- $p(x_i)$ is the probability of the event $X = x_i$.



➢ The unit of measurement is the binary information unit.

- **1 bit of information** corresponds to 0.5.

➢ Note: it is not the same as binary bit.

- For a fair coin toss, we have received 1 bit of information.

What is Information Entropy?

- Consider a discrete random variable $X = \{x_1, x_2, \dots, x_n\}$ with respective probabilities $p(x_1), p(x_2), \dots, p(x_n)$.

- The information in **independent events** is additive.

- The information entropy $H(X)$ of X is the **expected value of the information** produced by a stochastic outcome of X.

$$H(X) = \sum_{i=1}^{n} p(x_i) I(X = x_i) = \sum_{i=1}^{n} p(x_i) \log_2\left(\frac{1}{p(x_i)}\right)$$

Probability Information

Example: Measuring Information

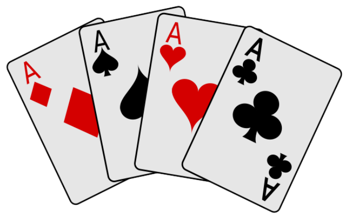
- How to measure information in terms of bits?

= ? bits

= ? bits

Example: Measuring Information

- **Draw cards at random from a standard 52-card deck.**
  - Because elementary outcome is probability **1/52**, information is $\log_2(52/1) = 5.7$ bits.



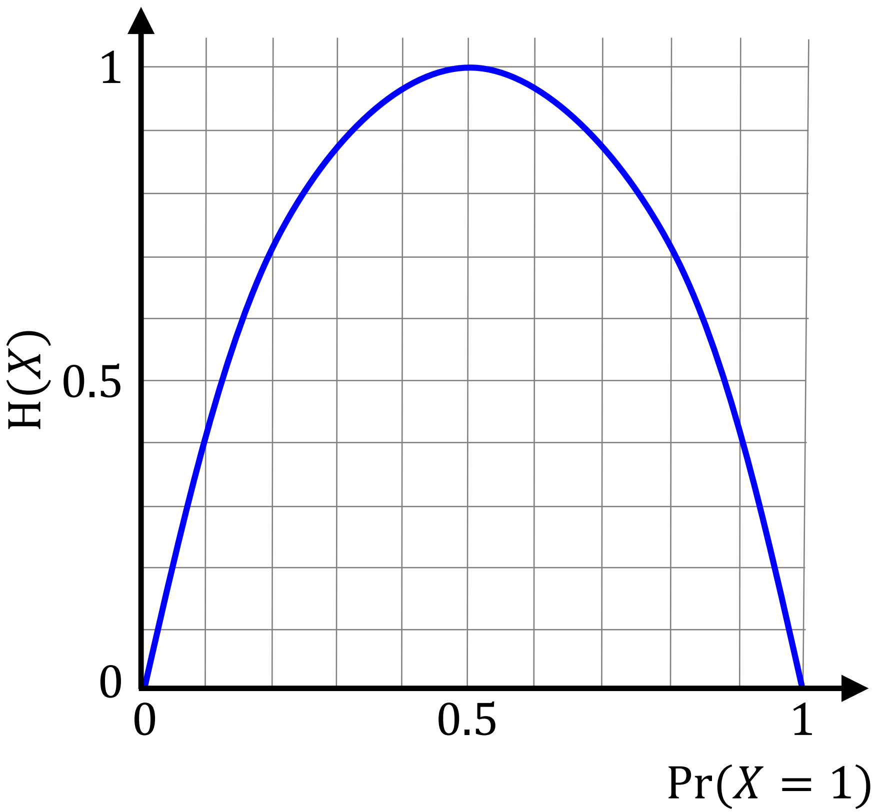
- **Q: If I tell you the card is a spade ♠, how many bits of information have you received?**
- **A: The probability of drawing a spade is 13/52, and the amount of information received is $\log_2(52/13) = 2$ bits.**

- **Q: If I tell you the card is a seven, how much information?**
- **A: $\log_2(52/4) = \log_2(13) = 3.7$ bits**

Binary Entropy Function

- Heads (or $C = 1$) with probability p
- Tails (or $C = 0$) with probability $1 - p$
- The entropy is **maximized** at 1 bit when two possible outcomes are **equally probable**, as in an unbiased coin toss.

$$H(C) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$



Example: Binary Entropy Function

- **Suppose that p = 1/1024, i.e., very small probability of getting a head in 1024 trials. Then,**

$$H(X) = -\left(\frac{1}{1024}\log_2\frac{1}{1024} + \frac{1023}{1024}\log_2\frac{1}{1024}\right) \approx 0.112$$

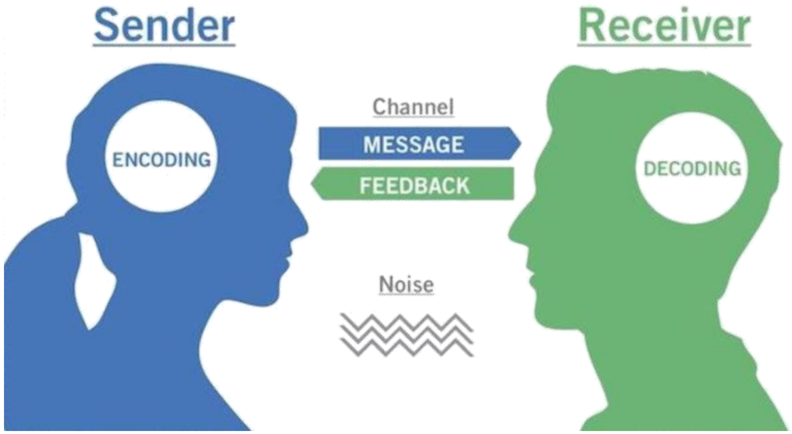
bits of uncertainty of information per trial on average

- **Because most trials are not surprised, using 1024 binary digits to code the results of 1024 tosses seems wasteful.**

Significance of Entropy

- **Entropy (in bits) tells us the average amount of information (in bits) that must be delivered.**
  - This is a **lower bound on the number of binary digits** that must, on average, be used to encode our messages.

- **Achieving the entropy lower bound is the gold standard for an encoding (from the view of information compression).**



Example: Code Compression

- The expected information in a choice is given by the entropy.

choice_i	p_i	$\log_2(1/p_i)$
A	1/2	1.00
B	1/3	1.58
C	1/12	3.58
D	1/12	3.58

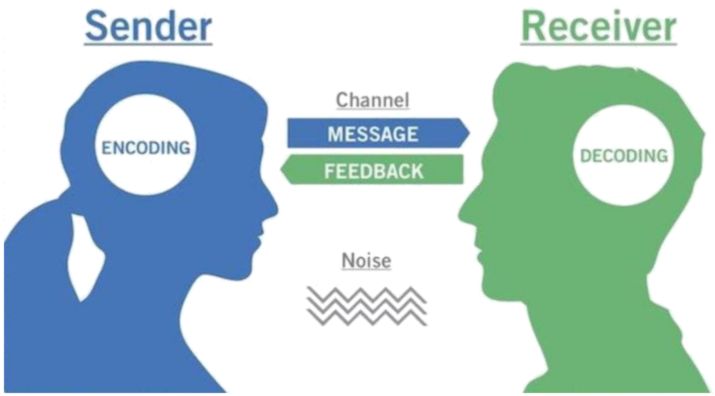
$$H(X) = 0.5 \times 1.00 + 0.333 \times 1.58 + 2 \times 0.083 \times 3.58 = 1.626 \text{ bits}$$

- Can we find a better encoding where transmitting 1000 choices requires 1626 binary digits on average?

Example: Code Compression

- The expected information in a choice is given by the entropy.

$choice_i$	p_i	$encoding$
A	1/2	00
B	1/3	01
C	1/12	10
D	1/12	11

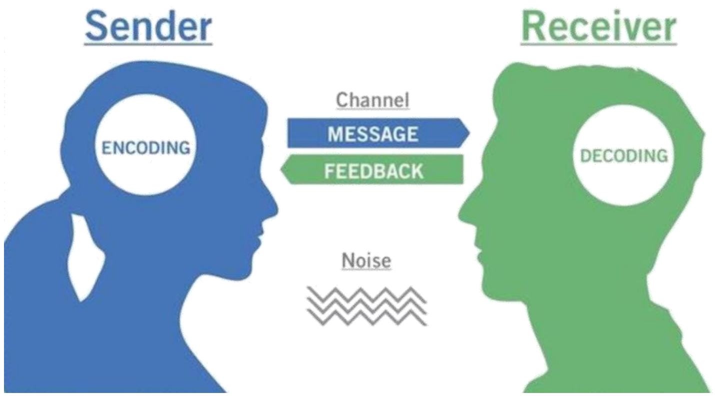


- The natural fixed-length encoding uses two binary digits for each choice. How many bits?
- It requires 2,000 binary digits for 1,000 choices.

Example: Code Compression

- The expected information in a choice is given by the entropy.

$choice_i$	p_i	$encoding$
A	1/2	**0**
B	1/3	**10**
C	1/12	**110**
D	1/12	**111**



$$0.5 \times 1 + 0.333 \times 2 + 2 \times 0.83 \times 3 = 1.666$$

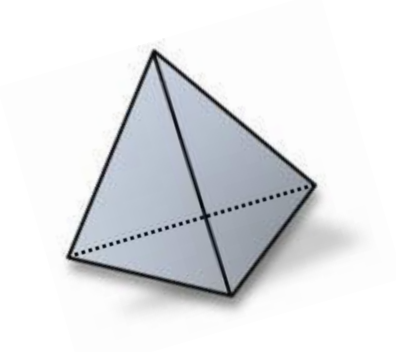
- The various-length encoding requires **1,666 binary digits** for **1,000 choices.** However, it is **NOT optimal**.
- Theoretically, **the optimal value is 1,626 binary digits**.

Cross-Entropy

- Measuring the **average number of bits** needed if a coding scheme is used on a **probability distribution Q**, rather than **true distribution P**.

$$H(\boldsymbol{P}, \boldsymbol{Q}) = -\sum_{i=1}^{n} \boldsymbol{p}(\boldsymbol{x}_i) \log_2 \boldsymbol{q}(\boldsymbol{x}_i)$$

- **Example: four-side dice**



true distribution observed distribution

- $P = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right), Q = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right)$

$$H(P, Q) = -\frac{1}{4}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{8} - \frac{1}{4}\log_2\frac{1}{8} = 2.25$$

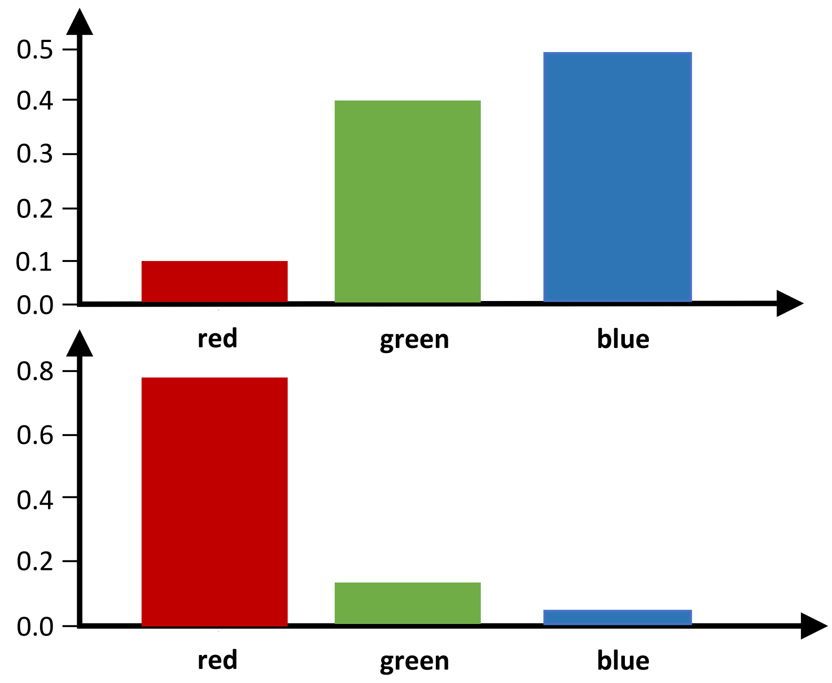
Example: Cross-Entropy

- **Given $P = (0.1, 0.4, 0.5)$ and $Q = (0.8, 0.15, 0.05)$**

$$H(P,Q) = -0.1\log_2 0.8 - 0.4\log_2 0.15 - 0.5\log_2 0.05 = 3.288$$

$$H(Q,P) = -0.8\log_2 0.1 - 0.15\log_2 0.4 - 0.05\log_2 0.5 = 2.906$$

- **It is asymmetric.**



https://machinelearningmastery.com/cross-entropy-for-machine-learning/

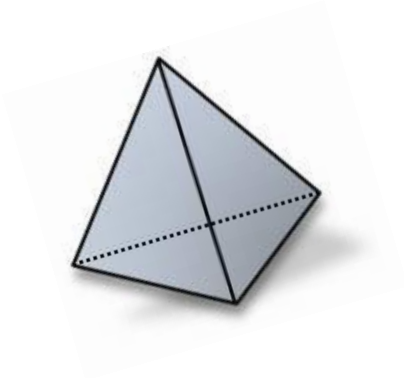
Relative Entropy

- Given $P = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$ and $Q = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right)$,

- We compare $H(\boldsymbol{P}, \boldsymbol{Q})$ with $H(\boldsymbol{P}, \boldsymbol{P})$.

$$H(P, Q) = -\frac{1}{4}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{8} - \frac{1}{4}\log_2\frac{1}{8} = 2.25$$

$$H(P, P) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} = 2.0$$



- The additional bits using Q are 2.25 – 2.00 = 0.25.

Relative Entropy

- **The relative entropy between two probability distributions $\boldsymbol{p(X)}$ and $\boldsymbol{q(X)}$**
  - Also, called **Kullback-Leibler (KL) divergence**

$$KL(P\,||\,Q) = H(P,Q) - H(P)$$

$$= \left(-\sum_{i=1}^{n} p(x_i)\log_2 q(x_i)\right) - \left(-\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)\right)$$

$$= -\sum_{i=1}^{n} p(x_i)\log_2 \frac{q(x_i)}{p(x_i)}$$

- **Because $\boldsymbol{H(P)}$ is fixed, minimizing the KL divergence of $\boldsymbol{Q}$ from $\boldsymbol{P}$ with respect to $\boldsymbol{Q}$ is equivalent to minimizing the cross entropy of $\boldsymbol{P}$ and $\boldsymbol{Q}$.**

Properties of KL Divergence

- $KL(P \,||\, Q) \geq 0$
- When $P = Q$, $KL(P \,||\, Q) = 0$
- Asymmetric: $KL(P \,||\, Q) \neq KL(Q \,||\, P)$
- KL divergence is not a distance.
- However, it has been widely used as the measure for the distance between two probability distributions.
- Usually, the true distribution P is given.

Proof: Non-Negativity of KL Divergence

- **If X is random variable, and $f(X)$ is convex function,**

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$$

- **Assuming $f(X) = -\log_2 \frac{q(x_i)}{p(x_i)}$,**

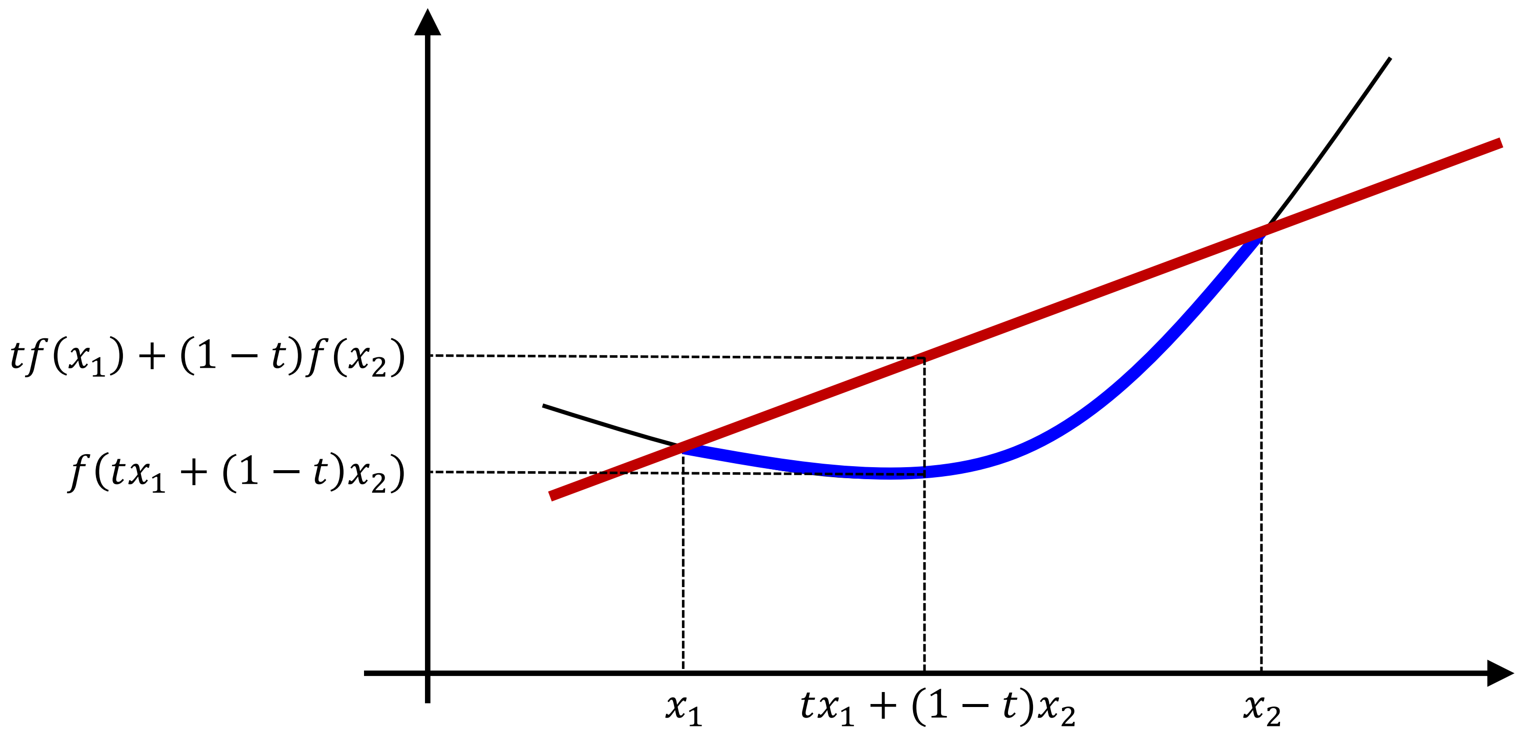
$$KL(P\,||\,Q) = -\sum_{i=1}^{n} p(x_i)\log_2 \frac{q(x_i)}{p(x_i)} \geq -\log_2 \sum_{i=1}^{n} p(x_i)\frac{q(x_i)}{p(x_i)}$$

$$\geq -\log_2 \sum_{i=1}^{n} q(x_i) = -\log_2 1 = 0$$

Jensen's Inequality

- A function f is convex $\Leftrightarrow$ the function f is **below** any line segment between two points on f.

$$f(tx + (1-t)x') \leq tf(x) + (1-t)f(x') \text{ for any } x, x' \in X \text{ and } t \in [0,1]$$



Jensen-Shannon Divergence (JSD)

- **Forward KL**

$$KL(P \,||\, Q) = -\sum_{i=1}^{n} p(x_i) \log_2 \frac{q(x_i)}{p(x_i)}$$

- **Backward KL**

$$KL(Q \,||\, P) = -\sum_{i=1}^{n} q(x_i) \log_2 \frac{p(x_i)}{q(x_i)}$$

- **It is a symmetrized version of the KL divergence.**

$$JSD(P \,||\, Q) = \frac{1}{2} KL\left(P \,||\, \frac{P+Q}{2}\right) + \frac{1}{2} KL\left(Q \,||\, \frac{P+Q}{2}\right)$$

Information Theory used in ML

- **Cross-entropy is commonly used as the loss function for the classification problem.**
  - E.g., logistic regression adopts the cross entropy as the loss function.
- **KD divergence is used to measure the similarity between two probability distributions.**
- **JSD is used to prove the training process for generative adversarial networks (GANs).**

Q&A