Entropy and Kolmogorov Complexity

Seungjin Choi

Department of Computer Science and Engineering
Pohang University of Science and Technology
77 Cheongam-ro, Nam-gu, Pohang 790-784, Korea
seungjin@postech.ac.kr
Convex Sets and Functions

Definition (Convex Sets)
Let $C$ be a subset of $\mathbb{R}^m$. $C$ is called a convex set if

$$\alpha x + (1 - \alpha)y \in C, \quad \forall x, y \in C, \forall \alpha \in [0, 1]$$

Definition (Convex Function)
Let $C$ be a convex subset of $\mathbb{R}^m$. A function $f : C \mapsto \mathbb{R}$ is called a convex function if

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad \forall x, y \in C, \forall \alpha \in [0, 1]$$
Jensen’s Inequality

Theorem (Jensen’s Inequality)

If \( f(x) \) is a \textit{convex function} and \( x \) is a random vector, then

\[
E\{f(x)\} \geq f(E\{x\}).
\]

Note: Jensen’s inequality can also be rewritten for a \textit{concave function}, with the \textit{direction of the inequality} reversed.
Proof of Jensen’s Inequality

Need to show that \( \sum_{i=1}^{N} p_i f(x_i) \geq f \left( \sum_{i=1}^{N} p_i x_i \right) \). The proof is based on the recursion, working from the right-hand side of this equation.

\[
f \left( \sum_{i=1}^{N} p_i x_i \right) = f \left( p_1 x_1 + \sum_{i=2}^{N} p_i x_i \right)
\]

\[
\leq p_1 f(x_1) + \left[ \sum_{i=2}^{N} p_i \right] f \left( \frac{\sum_{i=2}^{N} p_i x_i}{\sum_{i=2}^{N} p_i} \right) \quad \left( \text{choose } \alpha = \frac{p_1}{\sum_{i=1}^{N} p_i} \right)
\]

\[
\leq p_1 f(x_1) + \left[ \sum_{i=2}^{N} p_i \right] \left\{ \alpha f(x_2) + (1 - \alpha) f \left( \frac{\sum_{i=3}^{N} p_i x_i}{\sum_{i=3}^{N} p_i} \right) \right\}
\]

\[
\left( \text{choose } \alpha = \frac{p_2}{\sum_{i=2}^{N} p_i} \right)
\]

\[
= p_1 f(x_1) + p_2 f(x_2) + \sum_{i=3}^{N} p_i f \left( \frac{\sum_{i=3}^{N} p_i x_i}{\sum_{i=3}^{N} p_i} \right)
\]

and so forth.
Information Theory

- Information theory answers two fundamental questions in communication theory
  - What is the ultimate data compression? $\rightarrow$ entropy $H$.
  - What is the ultimate transmission rate of communication? $\rightarrow$ channel capacity $C$.

- In the early 1940’s, it was thought that increasing the transmission rate of information over a communication channel increased the probability of error $\rightarrow$ ”This is not true.”
  Shannon surprised the communication theory community by proving that this was not true as long as the communication rate was below the channel capacity.

- Although information theory was developed for communications, it is also important to explain ecological theory of sensory processing. Information theory plays a key role in elucidating the goal of unsupervised learning.
Information and Entropy

- **Information** can be thought of as surprise, uncertainty, or unexpectedness. Mathematically it is defined by

\[ I = - \log P(i), \]

where \( P(i) \) is the probability that the event labelled \( i \) occurs. The rare event gives large information and frequent event produces small information.

- **Entropy** is average information, i.e.,

\[ H = - \sum_{i=1}^{N} P(i) \log P(i). \]
Example: Horse Race

Suppose we have a horse race with eight horses taking part. Assume that the probabilities of winning for the eight horses are

\[
\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\right).
\]

Suppose that we wish to send a message to another person indicating which horse won the race.

How many bits are required to describe this for each of the horses?

3 bits for any of the horses?
No! The win probabilities are not uniform. It makes sense to use shorter descriptions for the more probable horses and longer descriptions for the less probable ones so that we achieve a lower average description length. For example, we can use the following strings to represent the eight horses:

0, 10, 110, 1110, 111100, 111101, 111110, 111111.

The average description length in this case is 2 bits as opposed to 3 bits for the uniform code.

We calculate the entropy:

\[
H(X) = -\frac{1}{2} \log \frac{1}{2} - \frac{1}{4} \log \frac{1}{4} - \frac{1}{8} \log \frac{1}{8} - \frac{1}{16} \log \frac{1}{16} - 4 \frac{1}{64} \log \frac{1}{64}
\]

\[
= 2 \text{ bits.}
\]

The entropy of a random variable is a lower bound on the average number of bits required to represent the random variables and also on the average number of questions needed to identify the variable in a game of "twenty questions".
Entropy and Relative Entropy

- **Entropy** is the average information (a measure of uncertainty) that is defined by

\[
H(x) = -\sum_{x \in X} p(x) \log p(x)
\]

\[
= -E_p\{\log p(x)\}.
\]

- **Relative entropy** (Kullback-Leibler divergence) is a measure of distance between two distributions and is defined by

\[
KL[p\|q] = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}
\]

\[
= E_p \left\{ \log \frac{p(x)}{q(x)} \right\}.
\]
Mutual Information

Mutual information is the relative entropy between the joint distribution and the product of marginal distributions,

\[ I(x, y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right) \]

\[ = D[p(x, y) \parallel p(x)p(y)] \]

\[ = E_{p(x,y)} \left\{ \log \left( \frac{p(x, y)}{p(x)p(y)} \right) \right\}. \]

Mutual information can be interpreted as the reduction in the uncertainty of \( x \) due to the knowledge of \( y \), i.e.,

\[ I(x, y) = H(x) - H(x|y), \]

where \( H(x|y) = -E_{p(x,y)} \{\log p(x|y)\} \) is the conditional entropy.
Gibb's Inequality

Theorem

\[ KL[p||q] \geq 0 \text{ with equality iff } p = q. \]

Proof: Consider the Kullback-Leibler divergence for discrete distributions:

\[
KL[p||q] = \sum_i p_i \log \frac{p_i}{q_i}
= - \sum_i p_i \log \frac{q_i}{p_i}
\geq - \log \left[ \sum_i p_i \frac{q_i}{p_i} \right] \quad \text{(by Jensen’s inequality)}
= - \log \left[ \sum_i q_i \right]
= 0.
\]
More on Gibb’s Inequality

In order to find the distribution $p$ which minimizes $KL[p||q]$, we consider a Lagrangian

$$
E = KL[p||q] + \lambda \left(1 - \sum_i p_i \right) = \sum_i p_i \frac{p_i}{q_i} + \lambda \left(1 - \sum_i p_i \right).
$$

Compute the partial derivative $\frac{\partial E}{\partial p_k}$ and set to zero,

$$
\frac{\partial E}{\partial p_k} = \log p_k - \log q_k + 1 - \lambda = 0,
$$

which leads to $p_k = q_k e^{\lambda - 1}$. It follows from $\sum_i p_i = 1$ that $\sum_i q_i e^{\lambda - 1} = 1$, which leads to $\lambda = 1$. Therefore $p_i = q_i$.

The Hessian, $\frac{\partial^2 E}{\partial p_i^2} = \frac{1}{p_i}$, $\frac{\partial^2 E}{\partial p_i \partial p_j} = 0$, is positive definite, which shows that $p_i = q_i$ is a genuine minimum.
Universal Turing Machines

- A Turing machine that can simulate an arbitrary Turing machine on arbitrary input.
- Reprogrammable TM (the origin of stored program computer).
- A universal TM, $M_u$, is an automaton that given as input the description of any TM $M$ and a string $w$, can simulate the computation of $M$ on $w$.
- Consists of three tapes:
  - tape 1: description of $M$ (encoded definition of $M$);
  - tape 2: tape contents of $M$;
  - tape 3: internal state of $M$.
- $M_u$ looks first at the contents of tape 2 and tape 3 to determine the configuration of $M$. Then, $M_u$ consults tape 1 to see what $M$ would do in this configuration. Finally, tape 2 and tape 3 will be modified to reflect the result of the move.
In 1936, Turing was obsessed with the question of whether the thoughts in a living brain could be held equally well by a collection of inanimate parts. In short, could a machine think?

Human computational processes: A human thinks, writes, thinks some more, writes, and so on.

Consider a computer as a finite-state machine operating on a finite symbol set. A program tape, on which a binary program is written, is fed left to right into this finite-state machine. At each unit of time, the machine inspects the program tape, writes some symbols on a work tape, changes its state according to its transition table, and calls for more program. The operations of such a machine can be described by a finite list of transitions. Turing argued that this machine could mimic the computational ability of a human being.
Kolmogorov Complexity

- Also known as descriptive complexity, algorithmic entropy, program-size complexity.
- A measure of computability resources needed to specify the object.
- If $p$ is a program which outputs string $x$, then $p$ is a description of $x$.

**Definition**

The Kolmogorov complexity $K_U(x)$ of a string $x$ with respect to a universal computer $U$ is defined as

$$K_U(w) = \min_{p : U(p) = w} l(p),$$

the minimum length over all programs that print $w$ and halt. Thus, $K_U(w)$ is the shortest description length of $w$ over all descriptions interpreted by computer $U$. 


1. 01010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101}  

01 27 times

2. 011010100000100111100110011001111111001110111100110010

initial segments of binary expansion of $\sqrt{2} - 1$

3. 01111010011011000010100001100010000010000011111001

random
Universality of Kolmogorov Complexity

**Theorem**

If $\mathcal{U}$ is a universal computer, for any other computer $\mathcal{A}$, there exists a constant $c_{\mathcal{A}}$ such that

$$K_{\mathcal{U}}(x) \leq K_{\mathcal{A}}(x) + c_{\mathcal{A}},$$

for all strings $x \in \{0, 1\}^*$ and the constant $c_{\mathcal{A}}$ does not depend on $x$. 
Kolmogorov Complexity vs Shannon Entropy

Theorem
Let $P(x)$ be a recursive probability distribution such that $H(P) < \infty$. Then,

$$0 \leq \sum_x P(x)K(x) - H(P) \leq K(P),$$

where $K(x)$ is the Kolmogorov complexity of $x$, $H(P)$ is Shannon entropy of the distribution $P$, and $K(P)$ is the length of the shortest program that describes the distribution $P$.

The value of Shannon entropy equals the expected value of Kolmogorov complexity, up to a term that only depends on the distribution.