A flavor of Information Theory

Guillaume Tochon

guillaume.tochon@lrde.epita.fr

LRDE, EPITA

On data compressibility

Given some data, what kind of information can be compressed, and to which extent?

On data compressibility

Given some data, what kind of information can be compressed, and to which extent?

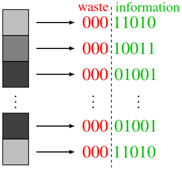
→ **Redundancy** : when the same information is repeated throughout the data.

Ex: Bunch of adjacent pixels with the same value in an image.

Keywords (`if`, `while`, `struct`,...) in a programming language.

→ **Waste** : when some information is given too much encoding support w.r.t. what would be necessary.

Ex: Grayscale image with only 32 different gray levels encoded on 1 byte ⇒ waste.



On data compressibility

Given some data, what kind of information can be compressed, and to which extent?

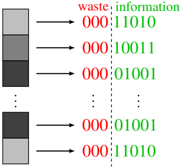
→ **Redundancy** : when the same information is repeated throughout the data.

Ex: Bunch of adjacent pixels with the same value in an image.

Keywords (`if`, `while`, `struct`,...) in a programming language.

→ **Waste** : when some information is given too much encoding support w.r.t. what would be necessary.

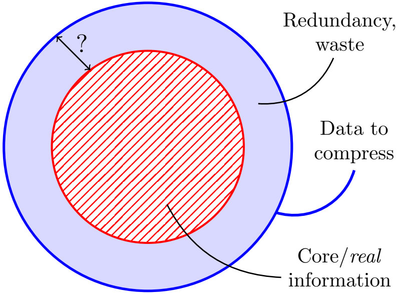
Ex: Grayscale image with only 32 different gray levels encoded on 1 byte ⇒ waste.



Compression ⇔ hunt for waste and redundancy.

Schematic representation of data

Data = Core information (incompressible) + redundancy/waste (compressible).



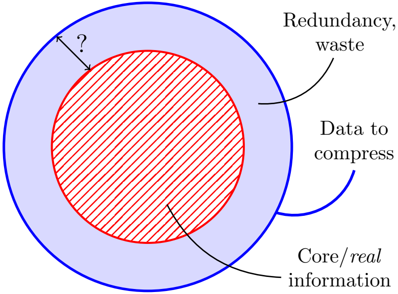
What can be compressed/removed :

$\rightarrow$ is *a priori* unknown.

$\rightarrow$ depends on the file itself.

Schematic representation of data

Data = Core information (incompressible) + redundancy/waste (compressible).



What can be compressed/removed :

→ is *a priori* unknown.

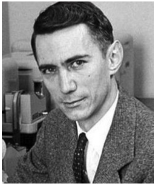
→ depends on the file itself.

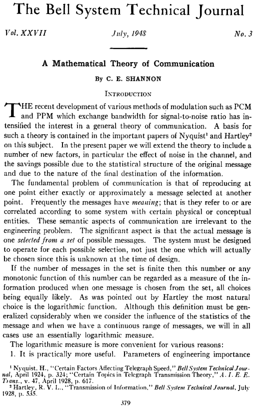
⇒ Need for an appropriate tool/compass to indicate to which degree a data can be compressed.

Shannon's Information theory

The tool we are looking for

Information theory provides the *entropy*, the tool we need to quantify compressibility of a given file.





The Bell System Technical Journal

Vol. XXVII *July, 1948* *No. 3*

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance

[1] Nyquist, H., "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal*, April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," *A. I. E. E. Trans.*, v. 47, April 1928, p. 617.
[2] Hartley, R. V. L., "Transmission of Information," *Bell System Technical Journal*, July 1928, p. 535.

379

- → Proposed by Claude E. Shannon in 1948.
- → Studies the quantification, storage and communication of *information*.
- → Based on the notion of *uncertainty* of some given event.
- → Has found applications in a countless number of (seemingly unrelated) fields, such as cryptography, natural langages processing, quantum computing or bioinformatics.

Information theory provides the *entropy*, the tool we need to quantify compressibility of a given file.

The Bell System Technical Journal

Vol. XXVII *July, 1948* *No. 3*

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance

[1] Nyquist, H., "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal*, April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," *A. I. E. E. Trans.*, v. 47, April 1928, p. 617.
[2] Hartley, R. V. L., "Transmission of Information," *Bell System Technical Journal*, July 1928, p. 535.

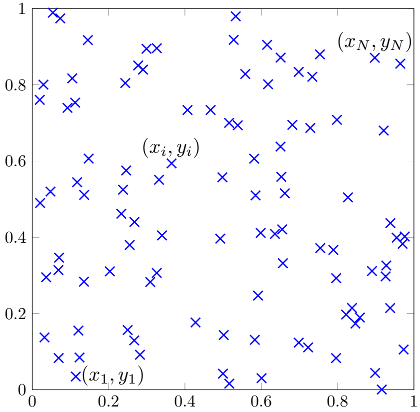
379

- → Proposed by Claude E. Shannon in 1948.
- → Studies the quantification, storage and communication of *information*.
- → Based on the notion of *uncertainty* of some given event.
- → Has found applications in a countless number of (seemingly unrelated) fields, such as cryptography, natural langages processing, quantum computing or bioinformatics.

A (if not THE) cornerstone of today's digital era.

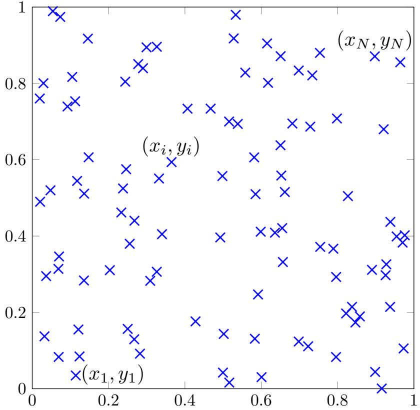
Compressibility and randomness (1/2)

How many values are necessary to store all coordinates $\{(x_i, y_i)\}_{i=1}^{N}$ of N points randomly generated and uniformly distributed in $[0,1] \times [0,1]$?



Compressibility and randomness (1/2)

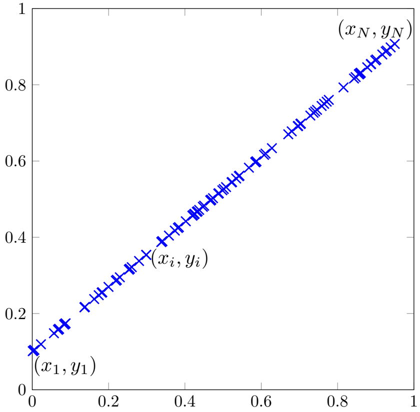
How many values are necessary to store all coordinates $\{(x_i, y_i)\}_{i=1}^{N}$ of N points randomly generated and uniformly distributed in $[0,1] \times [0,1]$?



All N couples $(x_1, y_1), \ldots, (x_N, y_N)$ must be stored $\rightarrow$ $\boxed{2N}$ values are necessary.

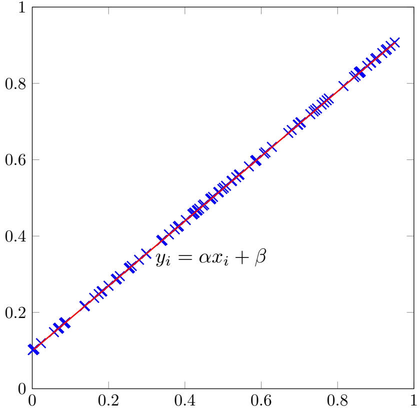
Compressibility and randomness (1/2)

How many values are necessary to store all coordinates $\{(x_i, y_i)\}_{i=1}^{N}$ of N points that are perfectly aligned?



Compressibility and randomness (1/2)

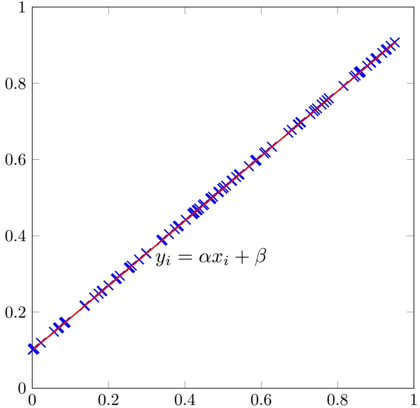
How many values are necessary to store all coordinates $\{(x_i, y_i)\}_{i=1}^{N}$ of N points that are perfectly aligned?



Each y-coordinate y_i can be deduced from x_i following $y_i = \alpha x_i + \beta$.

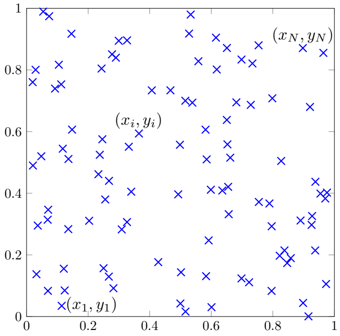
Compressibility and randomness (1/2)

How many values are necessary to store all coordinates $\{(x_i, y_i)\}_{i=1}^{N}$ of N points that are perfectly aligned?

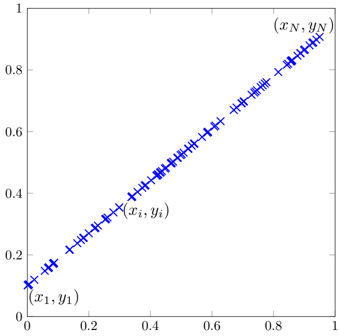


$(x_1, \ldots, x_N, \alpha, \beta)$ describes the whole data $\rightarrow$ $\boxed{N+2}$ values are sufficient.

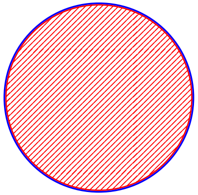
Compressibility and randomness (2/2)

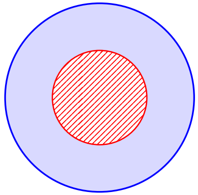


Random data $\Leftrightarrow$ not compressible ✗.



Ordered data $\Leftrightarrow$ compressible ✓.





Compression likes order

Compression is posssible whenever there is an underlying order structuring the data.

Ex: $y_i = \alpha x_i + \beta \rightarrow$ no need to store y_i.

Compression likes order

Compression is posssible whenever there is an underlying order structuring the data.

Ex: $y_i = \alpha x_i + \beta \rightarrow$ no need to store y_i.

But the relation structuring the order does not even need to be explicitely known.

Ex: Could you guess which letters have been hidden in the following texts?

1) H■re i■ a bu■ch of ■ords wit■ so■e let■ers hidde■ b■hind litt■e bla■k sq■ares.
2) Kg■ ■g f■ehk ■mlajd■i wpd■ib q mpzo■f az lg■j r■utv ■azni ghf■osdaq■ f■sn.

Compression likes order

Compression is posssible whenever there is an underlying order structuring the data.

Ex: $y_i = \alpha x_i + \beta \rightarrow$ no need to store y_i.

But the relation structuring the order does not even need to be explicitely known.

Ex: Could you guess which letters have been hidden in the following texts?

1) H■re i■ a bu■ch of ■ords wit■ so■e let■ers hidde■ b■hind litt■e bla■k sq■ares.
2) Kg■ ■g f■ehk ■mlajd■i wpd■ib q mpzo■f az lg■j r■utv ■azni ghf■osdaq■ f■sn.

1) Here is a bunch of words with some letters hidden behind little black squares.
   - $\rightarrow$ The order is implicitely set by the langage structure. Therefore, the previous piece of text is *ordered*, and should thus be compressible.

Compression likes order

Compression is posssible whenever there is an underlying order structuring the data.

Ex: $y_i = \alpha x_i + \beta \rightarrow$ no need to store y_i.

But the relation structuring the order does not even need to be explicitely known.

Ex: Could you guess which letters have been hidden in the following texts?

1) H■re i■ a bu■ch of ■ords wit■ so■e let■ers hidde■ b■hind litt■e bla■k sq■ares.
2) Kg■ ■g f■ehk ■mlajd■i wpd■ib q mpzo■f az lg■j r■utv ■azni ghf■osdaq■ f■sn.

1) Here is a bunch of words with some letters hidden behind little black squares.
   - $\rightarrow$ The order is implicitely set by the langage structure. Therefore, the previous piece of text is *ordered*, and should thus be compressible.
2) Kg? ?g f?ehk ?mlajd?i wpd?ib q mpzo?f az lg?j r?utv ?azni ghf?osdaq? f?sn.
   - $\rightarrow$ Letters were drawn at random, the message is meaningless and the missing bits are impossible to guess.

Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25^{th}.

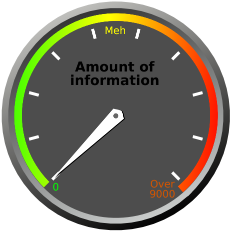
Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25^{th}.



Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25^{th}.
- ⋆ This year, it will be 15°C on average on Bastille day.

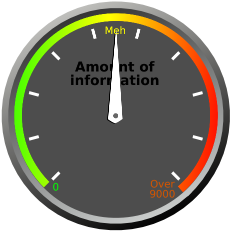
Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25th.
- ⋆ This year, it will be 15°C on average on Bastille day.



Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25^{th}.
- ⋆ This year, it will be 15°C on average on Bastille day.
- ⋆ Next year, PSG will win the UEFA Champions League.

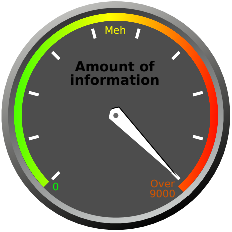
Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25th.
- ⋆ This year, it will be 15°C on average on Bastille day.
- ⋆ Next year, PSG will win the UEFA Champions League.



Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25^{th}.
- ⋆ This year, it will be 15°C on average on Bastille day.
- ⋆ Next year, PSG will win the UEFA Champions League.

Toward the notion of self-information (1/2)

To build its mathematical theory of Information, Shannon asks himself the question:

What amount of information carries a given message?

As an example, rate the amount of information contained in the following sentences:

- ⋆ This year, Christmas will be celebrated on December, 25^{th}.
- ⋆ This year, it will be 15°C on average on Bastille day.
- ⋆ Next year, PSG will win the UEFA Champions League.

Information $\equiv$ Unlikely event $\equiv$ Scoop.

Toward the notion of self-information (2/2)

- → From Shannon's point of view, the more unlikely/surprising a message is, the higher the *self-information* (*a.k.a*, the amount of information) this message contains.
- → From a probabilistic point of view, the likeliness of a message is defined as the probability that this message occurs (among the set of all considered messages).

Toward the notion of self-information (2/2)

→ From Shannon's point of view, the more unlikely/surprising a message is, the higher the *self-information* (*a.k.a*, the amount of information) this message contains.

→ From a probabilistic point of view, the likeliness of a message is defined as the probability that this message occurs (among the set of all considered messages).

⇒ A message m that happens with certainty does not contain any information. Contrarily, a message m that is very unlikely has a high self-information.

$$q(m) = f\left(\frac{1}{\mathbb{p}(m)}\right)$$

Toward the notion of self-information (2/2)

→ From Shannon's point of view, the more unlikely/surprising a message is, the higher the *self-information* (*a.k.a*, the amount of information) this message contains.

→ From a probabilistic point of view, the likeliness of a message is defined as the probability that this message occurs (among the set of all considered messages).

⇒ A message m that happens with certainty does not contain any information. Contrarily, a message m that is very unlikely has a high self-information.

$$\underbrace{q(m)}_{\text{self-information of message } m} = \underbrace{f}_{\text{unknown mapping}}\left(\frac{1}{\underbrace{\mathrm{p}(m)}_{\text{probability of occurrence of message } m}}\right)$$

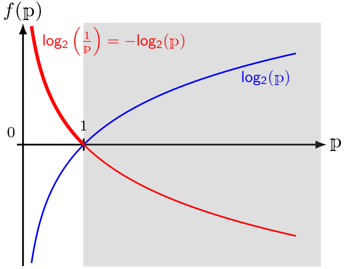
Toward the notion of self-information (2/2)

→ From Shannon's point of view, the more unlikely/surprising a message is, the higher the *self-information* (*a.k.a*, the amount of information) this message contains.

→ From a probabilistic point of view, the likeliness of a message is defined as the probability that this message occurs (among the set of all considered messages).

⇒ A message *m* that happens with certainty does not contain any information. Contrarily, a message *m* that is very unlikely has a high self-information.

$$\boxed{q(m) = \log_2\left(\frac{1}{\mathbb{p}(m)}\right)} = -\log_2(\mathbb{p}(m))$$



Toward the notion of self-information (2/2)

→ From Shannon's point of view, the more unlikely/surprising a message is, the higher the *self-information* (*a.k.a*, the amount of information) this message contains.

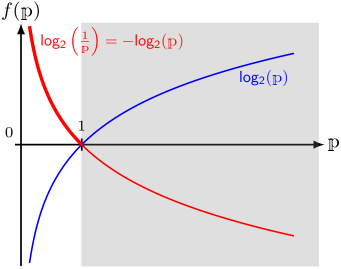
→ From a probabilistic point of view, the likeliness of a message is defined as the probability that this message occurs (among the set of all considered messages).

⇒ A message m that happens with certainty does not contain any information. Contrarily, a message m that is very unlikely has a high self-information.

$$\boxed{q(m) = \log_2\left(\frac{1}{\mathbb{p}(m)}\right)} = -\log_2(\mathbb{p}(m))$$

→ The unit of $q(m)$ is the Shannon (abbrv. *Sh*)

→ The log function is particularly convenient thanks to its property that $\log(a \times b) = \log(a) + \log(b)$



Introducing the entropy (1/2)

Let us consider some alphabet Σ composed of N_Σ symbols $\{s_1, s_2, \dots, s_{N_\Sigma}\}$, where each symbol s_i has a probability of occurrence being $\mathbb{p}(s_i) = \mathbb{p}_i$ (with $\sum_{i=1}^{N_\Sigma} \mathbb{p}_i = 1$).

Ex: Classical latin alphabet $\Rightarrow N_\Sigma = 26$, $s_1 = a$, $s_2 = b$, and so on...

$\rightarrow$ The self-information of a symbol s_i is $q(s_i) = -\log_2(\mathbb{p}_i)$ Sh.

Introducing the entropy (1/2)

Let us consider some alphabet Σ composed of N_Σ symbols $\{s_1, s_2, \ldots, s_{N_\Sigma}\}$, where each symbol s_i has a probability of occurrence being $\mathbb{p}(s_i) = \mathbb{p}_i$ (with $\sum_{i=1}^{N_\Sigma} \mathbb{p}_i = 1$).

Ex: Classical latin alphabet $\Rightarrow N_\Sigma = 26$, $s_1 = a$, $s_2 = b$, and so on...

$\rightarrow$ The self-information of a symbol s_i is $q(s_i) = -\log_2(\mathbb{p}_i)$ Sh.

Consider also some text file F composed of N_F symbols (*e.g.* a page of text).

$\rightarrow$ The symbol s_i is statistically present $N_F \times \mathbb{p}_i$ times in the file F.

$\rightarrow$ Thus, the total self-information of s_i in F is $Q_{tot}(s_i) = -N_F \mathbb{p}_i \log_2(\mathbb{p}_i)$ (with convention that $\mathbb{p}_i \log_2(\mathbb{p}_i) = 0$ if $\mathbb{p}_i = 0$).

$\rightarrow$ And the total self-information of F is

$$Q_{\text{tot}}(F) = \sum_{i=1}^{N_\Sigma} Q_{tot}(s_i) = -N_F \sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i) \; Sh.$$

Introducing the entropy (1/2)

Let us consider some alphabet Σ composed of N_Σ symbols $\{s_1, s_2, \ldots, s_{N_\Sigma}\}$, where each symbol s_i has a probability of occurrence being $\mathbb{p}(s_i) = \mathbb{p}_i$ (with $\sum_{i=1}^{N_\Sigma} \mathbb{p}_i = 1$).

Ex: Classical latin alphabet $\Rightarrow N_\Sigma = 26$, $s_1 = a$, $s_2 = b$, and so on...

$\rightarrow$ The self-information of a symbol s_i is $q(s_i) = -\log_2(\mathbb{p}_i)$ *Sh*.

Consider also some text file F composed of N_F symbols (*e.g.* a page of text).

$\rightarrow$ The symbol s_i is statistically present $N_F \times \mathbb{p}_i$ times in the file F.

$\rightarrow$ Thus, the total self-information of s_i in F is $Q_{tot}(s_i) = -N_F \mathbb{p}_i \log_2(\mathbb{p}_i)$ (with convention that $\mathbb{p}_i \log_2(\mathbb{p}_i) = 0$ if $\mathbb{p}_i = 0$).

$\rightarrow$ And the total self-information of F is

$$Q_{\text{tot}}(F) = \sum_{i=1}^{N_\Sigma} Q_{tot}(s_i) = -N_F \sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i) \; Sh.$$

But defined as such, $Q_{\text{tot}}(F)$ can be arbitrarily large, and makes pointless the comparison of self-information of two files of uneven sizes.

Introducing the entropy (2/2)

Solution: normalizing $Q_{\text{tot}}(F)$ by the size of the file F yields the definition of the *entropy* of F.

Entropy

The (Shannon) entropy of the N_Σ symbols $\{s_1, s_2, \ldots, s_{N_\Sigma}\}$ is defined as

$$\boxed{H = -\sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i)}$$

Introducing the entropy (2/2)

Solution: normalizing $Q_{\text{tot}}(F)$ by the size of the file F yields the definition of the *entropy* of F.

Entropy

The (Shannon) entropy of the N_Σ symbols $\{s_1, s_2, \ldots, s_{N_\Sigma}\}$ is defined as

$$\boxed{H = -\sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i)}$$

Remarks:

- It no longer depends on the considered file F, but only on the probability distribution $\mathbb{p}_i, i = 1, \ldots, N_\Sigma$ of the symbols composing the alphabet Σ.

Introducing the entropy (2/2)

Solution: normalizing $Q_{\text{tot}}(F)$ by the size of the file F yields the definition of the *entropy* of F.

Entropy

The (Shannon) entropy of the N_Σ symbols $\{s_1, s_2, \ldots, s_{N_\Sigma}\}$ is defined as

$$\boxed{H = -\sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i)}$$

Remarks:

- It no longer depends on the considered file F, but only on the probability distribution $\mathbb{p}_i, i = 1, \ldots, N_\Sigma$ of the symbols composing the alphabet Σ.
- It is expressed in Shannon/symbol (abbrv. *Sh/symb*).

Introducing the entropy (2/2)

Solution: normalizing $Q_{\text{tot}}(F)$ by the size of the file F yields the definition of the *entropy* of F.

Entropy

The (Shannon) entropy of the N_Σ symbols $\{s_1, s_2, \ldots, s_{N_\Sigma}\}$ is defined as

$$\boxed{H = -\sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i)}$$

Remarks:

- It no longer depends on the considered file F, but only on the probability distribution $\mathbb{p}_i, i = 1, \ldots, N_\Sigma$ of the symbols composing the alphabet Σ.
- It is expressed in Shannon/symbol (abbrv. *Sh/symb*).
- It can be written as $H = \mathbb{E}\left[q(s_i)\right]$, where $\mathbb{E}[\cdot]$ is the expected value operator and $q(s_i) = -\log_2(\mathbb{p}_i)$ is the self-information of symbol $s_i \rightarrow$ the entropy H is the average of the self-information of all symbols s_i in Σ.

Binary entropy function

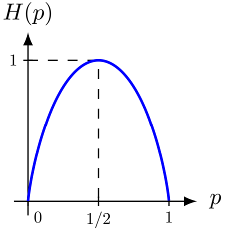
Consider a file F built upon a binary alphabet $\Sigma = \{0, 1\}$, where 0 has a probability of occurrence being $\mathbb{p}_0 = p \in [0, 1]$ (thus 1 having a probability $\mathbb{p}_1 = 1 - p$).

Binary entropy function

Consider a file F built upon a binary alphabet $\Sigma = \{0, 1\}$, where 0 has a probability of occurrence being $\mathbb{p}_0 = p \in [0, 1]$ (thus 1 having a probability $\mathbb{p}_1 = 1 - p$).

By definition,

$$\begin{aligned} H &= -\mathbb{p}_0 \log_2(\mathbb{p}_0) - \mathbb{p}_1 \log_2(\mathbb{p}_1) \\ &= -p \log_2(p) - (1 - p) \log_2(1 - p) \end{aligned}$$

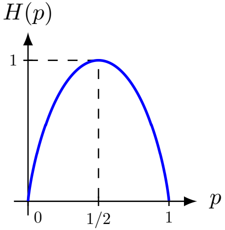


Binary entropy function

Consider a file F built upon a binary alphabet $\Sigma = \{0, 1\}$, where 0 has a probability of occurrence being $\mathbb{p}_0 = p \in [0, 1]$ (thus 1 having a probability $\mathbb{p}_1 = 1 - p$).

By definition,

$$\begin{aligned} H &= -\mathbb{p}_0 \log_2(\mathbb{p}_0) - \mathbb{p}_1 \log_2(\mathbb{p}_1) \\ &= -p \log_2(p) - (1-p) \log_2(1-p) \end{aligned}$$



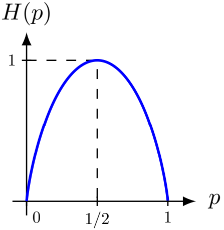
$\Rightarrow$ The entropy is maximal for $p = 1/2$ and decreases to 0 both for $p \to 0$ and $p \to 1$.

Binary entropy function

Consider a file F built upon a binary alphabet $\Sigma = \{0, 1\}$, where 0 has a probability of occurrence being $\mathbb{p}_0 = p \in [0, 1]$ (thus 1 having a probability $\mathbb{p}_1 = 1 - p$).

By definition,

$$\begin{aligned} H &= -\mathbb{p}_0 \log_2(\mathbb{p}_0) - \mathbb{p}_1 \log_2(\mathbb{p}_1) \\ &= -p \log_2(p) - (1-p) \log_2(1-p) \end{aligned}$$



$\Rightarrow$ The entropy is maximal for $p = 1/2$ and decreases to 0 both for $p \to 0$ and $p \to 1$.

What does it mean?

- When $p = 1/2$, both symbols 0 and 1 are equiprobable. The file F is completely random, thus incompressible, and $\boxed{\text{the entropy is maximal}}$.
- When $p \neq 1/2$, one symbol is more likely than the other. Some underlying order appears in the file F, which becomes compressible, and $\boxed{\text{the entropy decreases}}$.

Equiprobability $\equiv$ maximal disorder

Say that the file F contains the outcomes of dice rolls ($N_\Sigma = \{1, 2, 3, 4, 5, 6\}$).

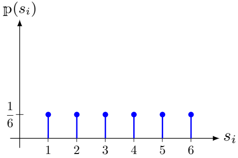
Equiprobability $\equiv$ maximal disorder

Say that the file F contains the outcomes of dice rolls ($N_\Sigma = \{1, 2, 3, 4, 5, 6\}$).

Fair dice $\Rightarrow$ outcomes $1, 2, \ldots, 6$ are equiprobable.

$\rightarrow$ $F = \{26435416542216\ldots\}$ is totally disordered.

$\rightarrow$ H is maximal (but how much?).



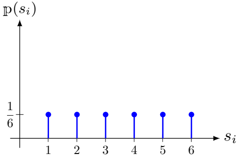
Equiprobability $\equiv$ maximal disorder

Say that the file F contains the outcomes of dice rolls ($N_\Sigma = \{1, 2, 3, 4, 5, 6\}$).

Fair dice $\Rightarrow$ outcomes $1, 2, \ldots, 6$ are equiprobable.

$\rightarrow$ $F = \{26435416542216\ldots\}$ is totally disordered.

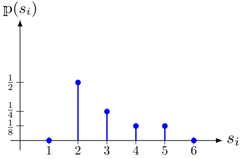
$\rightarrow$ H is maximal (but how much?).



Loaded dice $\Rightarrow$ some outcomes are more probable.

$\rightarrow$ $F = \{3225243252232\ldots\}$ is somewhat ordered.

$\rightarrow$ H is neither maximal nor minimal.



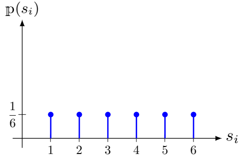
Equiprobability $\equiv$ maximal disorder

Say that the file F contains the outcomes of dice rolls ($N_\Sigma = \{1,2,3,4,5,6\}$).

Fair dice $\Rightarrow$ outcomes $1, 2, \ldots, 6$ are equiprobable.

$\rightarrow$ $F = \{26435416542216\ldots\}$ is totally disordered.

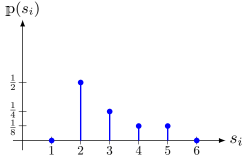
$\rightarrow$ H is maximal (but how much?).



Loaded dice $\Rightarrow$ some outcomes are more probable.

$\rightarrow$ $F = \{3225243252232\ldots\}$ is somewhat ordered.

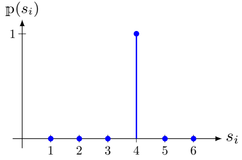
$\rightarrow$ H is neither maximal nor minimal.



Totally loaded dice (is that even possible?)

$\rightarrow$ $F = \{44444444444444\ldots\}$ is totally ordered.

$\rightarrow$ $H = 0$ is minimal.



Entropy as a compressibility gauge (1/2)

The binary case showed that the entropy H is maximal when all symbols $\{s_i\}_{i=1}^{N_\Sigma}$ in Σ are equiprobables ($\mathbb{p}_i = \frac{1}{N_\Sigma}$).

Assuming that $N_\Sigma = 2^m$. Then

$$H = -\sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i) = -\sum_{i=1}^{2^m} 2^{-m} \log_2(2^{-m}) = -2^m \times 2^{-m} \times (-m) = \boxed{m}$$

Therefore, $H < m$ if all symbols are not equiprobable (in general $H < \log_2(N_\Sigma)$ if $N_\Sigma \neq 2^m$).

Entropy as a compressibility gauge (1/2)

The binary case showed that the entropy H is maximal when all symbols $\{s_i\}_{i=1}^{N_\Sigma}$ in Σ are equiprobables ($\mathbb{p}_i = \frac{1}{N_\Sigma}$).

Assuming that $N_\Sigma = 2^m$. Then

$$H = -\sum_{i=1}^{N_\Sigma} \mathbb{p}_i \log_2(\mathbb{p}_i) = -\sum_{i=1}^{2^m} 2^{-m} \log_2(2^{-m}) = -2^m \times 2^{-m} \times (-m) = \boxed{m}$$

Therefore, $H < m$ if all symbols are not equiprobable (in general $H < \log_2(N_\Sigma)$ if $N_\Sigma \neq 2^m$).

The entropy of a file F gives an idea of "how compressible" is the file F:

→ Maximum entropy ⇔ complete randomness ⇔ incompressible file.

→ Lower entropy ⇔ underlying order ⇔ compressible file.

A random file has maximum information according to Shannon.

$\Rightarrow \underbrace{\text{Information}}_{\text{Mathematics}} \neq \underbrace{\text{Signification}}_{\text{semantic meaning}}$

Entropy as a compressibility gauge (2/2)

Example

Take $\Sigma = \{A, B, C, D\}$ (hence $N_\Sigma = 4$) with $\mathbb{p}_A = \mathbb{p}_B = \mathbb{p}_C = \mathbb{p}_D = \frac{1}{4}$.

$\Rightarrow$ Equiprobability $\Leftrightarrow$ maximum entropy $H = 2\ Sh/symb$

$\Leftrightarrow$ each symbol has to be encoded on 2 bits.

$\Leftrightarrow$ incompressible file (a priori).

Entropy as a compressibility gauge (2/2)

Example

Take $\Sigma = \{A, B, C, D\}$ (hence $N_\Sigma = 4$) with $\mathbb{p}_A = \mathbb{p}_B = \mathbb{p}_C = \mathbb{p}_D = \frac{1}{4}$.

$\Rightarrow$ Equiprobability $\Leftrightarrow$ maximum entropy $H = 2$ *Sh/symb*

$\Leftrightarrow$ each symbol has to be encoded on 2 bits.

$\Leftrightarrow$ incompressible file (a priori).

Now keep the same alphabet Σ, but with $\mathbb{p}_A = \frac{1}{2}, \mathbb{p}_B = \mathbb{p}_C = \frac{1}{4}, \mathbb{p}_D = 0$.

$H = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - 2 \times \frac{1}{4}\log_2\left(\frac{1}{4}\right) = \frac{3}{2}\ Sh/symb < 2$

$\Rightarrow$ Lower entropy $\Leftrightarrow$ compressible file.

$\Leftrightarrow$ each symbol can be encoded on 3/2 bits (on average).

Entropy as a compressibility gauge (2/2)

Example

Take $\Sigma = \{A, B, C, D\}$ (hence $N_\Sigma = 4$) with $\mathbb{p}_A = \mathbb{p}_B = \mathbb{p}_C = \mathbb{p}_D = \frac{1}{4}$.

$\Rightarrow$ Equiprobability $\Leftrightarrow$ maximum entropy $H = 2$ *Sh/symb*

$\Leftrightarrow$ each symbol has to be encoded on 2 bits.

$\Leftrightarrow$ incompressible file (a priori).

Now keep the same alphabet Σ, but with $\mathbb{p}_A = \frac{1}{2}, \mathbb{p}_B = \mathbb{p}_C = \frac{1}{4}, \mathbb{p}_D = 0$.

$H = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - 2 \times \frac{1}{4}\log_2\left(\frac{1}{4}\right) = \frac{3}{2}$ *Sh/symb* < 2

$\Rightarrow$ Lower entropy $\Leftrightarrow$ compressible file.

$\Leftrightarrow$ each symbol can be encoded on $3/2$ bits (on average).

Take some file F with $N_F = 1000$ symbols drawn from Σ.

- $\rightarrow$ 2000 bits are necessary to encode F with the first probability distribution.
- $\rightarrow$ But it can be encoded on $1000 \times \frac{3}{2} = 1500$ bits with the second distribution.

⚠ The value of H does not say anything on the most efficient way to attain this bound.

A quick link with thermodynamics and statistical physics

Carnot entropy

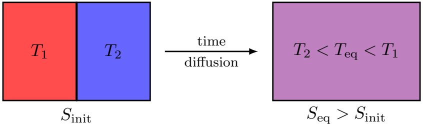
Thermodynamics (macroscopic point of view)

incremental entropy → $\boxed{dS = \frac{\delta Q}{T}}$ ← heat transfers

← temperature

Second law of thermodynamics

$dS \geq 0 \Rightarrow$ the entropy of a system always increases.



Boltzmann entropy

Statistical physics (microscopic point of view)

$$\boxed{S = k_B \ln(\Omega)}$$

k_B: Boltzmann constant ($1.38 \times 10^{-23} J.K^{-1}$)

Ω: Number of microscopic configurations yielding the current macroscopic one.

Boltzmann entropy is actually a particular case of Gibbs entropy

$$S = -k_B \sum_i \mathbb{p}_i \ln(\mathbb{p}_i)$$

when all microstates i of the system have the same probability $\mathbb{p}_i$.

The entropy of a thermodynamic system is a measure of the disorder of this system.

Exercise

How many bits are necessary to encode the file $F = \left\{ \text{ACABBDDBAAABCAAA} \right\}$ uncompressed?

Assuming that the probability of occurrence of the symbols in F is equal to their relative frequency, what would be the smallest compressed size of the file F?

Exercise

How many bits are necessary to encode the file $F = \Big\{$ ACABBDDBAAABCAAA $\Big\}$ uncompressed?
Assuming that the probability of occurrence of the symbols in F is equal to their relative frequency, what would be the smallest compressed size of the file F?

$\rightarrow$ F is composed of 4 different symbols $\{A, B, C, D\}$, so a support of 2 bits/symbols is necessary to encode the 4 symbols. In addition, F is composed of $N_F = 16$ symbols, hence a total uncompressed size of $\boxed{16 \times 2 = 32 \text{ bits}}$.

Exercise

How many bits are necessary to encode the file $F = \left\{ \text{ACABBDDBAAABCAAA} \right\}$ uncompressed?
Assuming that the probability of occurrence of the symbols in F is equal to their relative frequency, what would be the smallest compressed size of the file F?

→ F is composed of 4 different symbols $\{A, B, C, D\}$, so a support of 2 bits/symbols is necessary to encode the 4 symbols. In addition, F is composed of $N_F = 16$ symbols, hence a total uncompressed size of $\boxed{16 \times 2 = 32 \text{ bits}}$.

→ Let's first retrieve the probability of occurence of the symbols: $\mathbb{p}_A = \frac{8}{16} = \frac{1}{2}$, $\mathbb{p}_B = \frac{4}{16} = \frac{1}{4}$, $\mathbb{p}_C = \frac{2}{16} = \frac{1}{8}$ and $\mathbb{p}_D = \frac{2}{16} = \frac{1}{8}$.

Exercise

How many bits are necessary to encode the file $F = \left\{ \text{ACABBDDBAAABCAAA} \right\}$ uncompressed?
Assuming that the probability of occurrence of the symbols in F is equal to their relative frequency, what would be the smallest compressed size of the file F?

- $\rightarrow$ F is composed of 4 different symbols $\{A, B, C, D\}$, so a support of 2 bits/symbols is necessary to encode the 4 symbols. In addition, F is composed of $N_F = 16$ symbols, hence a total uncompressed size of $\boxed{16 \times 2 = 32 \text{ bits}}$.
- $\rightarrow$ Let's first retrieve the probability of occurence of the symbols: $\mathbb{p}_A = \frac{8}{16} = \frac{1}{2}$, $\mathbb{p}_B = \frac{4}{16} = \frac{1}{4}$, $\mathbb{p}_C = \frac{2}{16} = \frac{1}{8}$ and $\mathbb{p}_D = \frac{2}{16} = \frac{1}{8}$.
  It allows to compute the entropy:

$$\begin{aligned} H &= -\tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right) - \tfrac{1}{4}\log_2\left(\tfrac{1}{4}\right) - \tfrac{1}{8}\log_2\left(\tfrac{1}{8}\right) - \tfrac{1}{8}\log_2\left(\tfrac{1}{8}\right) \\ &= \tfrac{1}{2} + \tfrac{1}{2} + \tfrac{3}{8} + \tfrac{3}{8} = \tfrac{7}{4}\ Sh/symb. \end{aligned}$$

Exercise

How many bits are necessary to encode the file $F = \left\{ \text{ACABBDDBAAABCAAA} \right\}$ uncompressed?
Assuming that the probability of occurrence of the symbols in F is equal to their relative frequency, what would be the smallest compressed size of the file F?

→ F is composed of 4 different symbols $\{A, B, C, D\}$, so a support of 2 bits/symbols is necessary to encode the 4 symbols. In addition, F is composed of $N_F = 16$ symbols, hence a total uncompressed size of $\boxed{16 \times 2 = 32 \text{ bits}}$.

→ Let's first retrieve the probability of occurence of the symbols: $\mathbb{p}_A = \frac{8}{16} = \frac{1}{2}$, $\mathbb{p}_B = \frac{4}{16} = \frac{1}{4}$, $\mathbb{p}_C = \frac{2}{16} = \frac{1}{8}$ and $\mathbb{p}_D = \frac{2}{16} = \frac{1}{8}$.
It allows to compute the entropy:

$$\begin{aligned} H &= -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) \\ &= \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8} = \frac{7}{4}\ Sh/symb. \end{aligned}$$

Thus a minimal compressed size of $\boxed{16 \times \frac{7}{4} = 28 \text{ bits}}$.

But again, it does not say anything on the optimal encoding scheme...

N-gram entropy for text compression

Consider two files $F_1 = \left\{ \text{ABCDABCD} \right\}$ and $F_2 = \left\{ \text{AABCBADA} \right\}$.

N-gram entropy for text compression

Consider two files $F_1 = \left\{ \text{ABCDABCD} \right\}$ and $F_2 = \left\{ \text{AABCBADA} \right\}$.

You can easily check that $H_1 = 2\ Sh/symb$ and $H_2 = 1.75\ Sh/symb$. Thus, $H_1 > H_2$ even though F_1 appears more ordered than F_2.

Paradox?

Consider two files $F_1 = \left\{ \text{ABCDABCD} \right\}$ and $F_2 = \left\{ \text{AABCBADA} \right\}$.

You can easily check that $H_1 = 2\ Sh/symb$ and $H_2 = 1.75\ Sh/symb$. Thus, $H_1 > H_2$ even though F_1 appears more ordered than F_2.

Paradox? Actually, no!

The problem comes from a notion of scale: in its classical definition, the entropy considers symbols to be single characters (1-gram entropy).

But if you define the metasymbol ABCD, composed of 4 characters, then the 4-gram entropy of F_1 drops to 0, which is more in line with the intuition we have of it being a totally ordered file.

N-gram entropy for text compression

Consider two files $F_1 = \left\{ \text{ABCDABCD} \right\}$ and $F_2 = \left\{ \text{AABCBADA} \right\}$.

You can easily check that $H_1 = 2\ Sh/symb$ and $H_2 = 1.75\ Sh/symb$. Thus, $H_1 > H_2$ even though F_1 appears more ordered than F_2.

Paradox? Actually, no!

The problem comes from a notion of scale: in its classical definition, the entropy considers symbols to be single characters (1-gram entropy).

But if you define the metasymbol ABCD, composed of 4 characters, then the 4-gram entropy of F_1 drops to 0, which is more in line with the intuition we have of it being a totally ordered file.

However, two limitations naturally arise for text compression purposes:

1) The best order to consider depends on the langage.
2) If the alphabet has size N_Σ (hence, there are N_Σ 1-grams), there are $\binom{N_\Sigma}{k}$ different k-grams $\Rightarrow$ combinatorial explosion.