

# Information Theory

---

## Outline

Background

Basic Definitions

Information and Entropy

Independent Messages

Codings

# Information Theory *verses* Information Systems

## Information theory:

- originated in electrical engineering
  - ◇ communications
  - ◇ information theory as essential for modern communications as is the transistor
  
- for many years: an estranged cousin of information systems
  
- only recently have substantial connections between information theory and systems been discovered

## Motivation

Why study information theory in an information systems class?

- to see the only quantified model of information
  - ◊ much of philosophy, linguistics, cognitive science ask questions about information
- to think about information
- to avoid confusion of terminology
- to provide an informative example of mathematical modeling
  
- ★ to employ information measures in
  - ◊ database design
  - ◊ data warehousing
  - ◊ data mining

## Common Usage for “Information”

“ • • • lots of **information** • • • ”

implies either

1. meaningful
2. surprising

Information theory deals with:

- *surprise* and leaves meaning to the philosophers
- the quantity, not the value, of information

## Fundamental Assumption

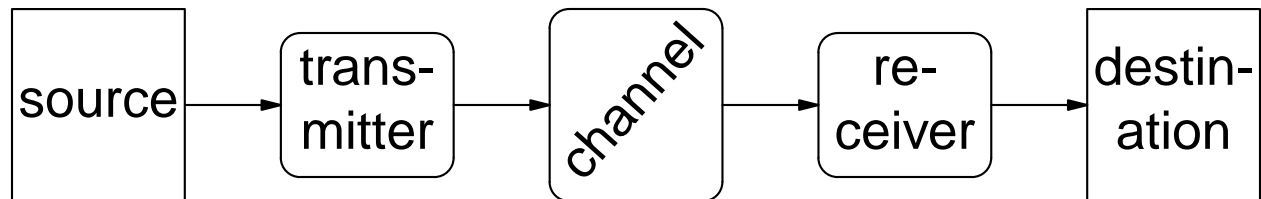
Information is something that *resolves uncertainty*

- If I tell you something you already know, there is no information in that message.
- If I tell you something that is very likely, there is little information in that message.
- If I tell you something surprising, that message has a lot of information.

∴ redundant data contains no information

# Channel

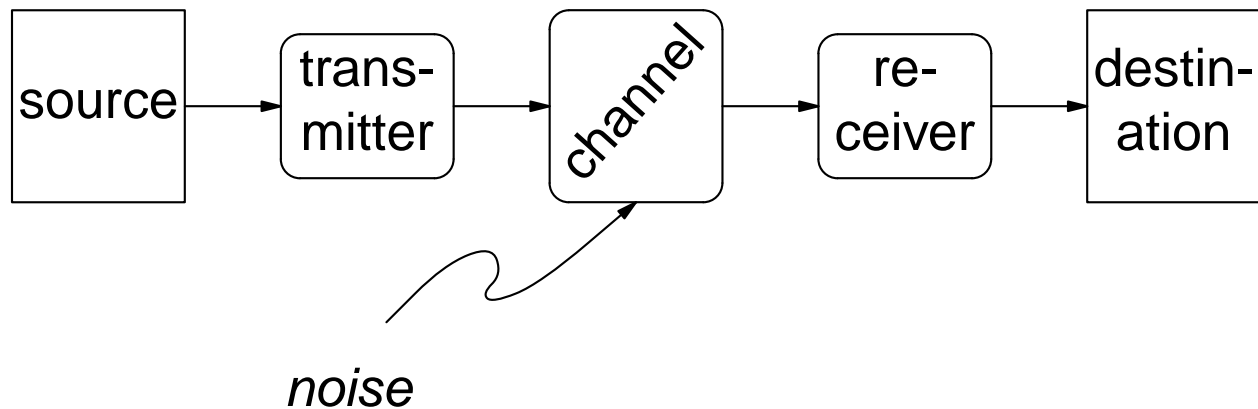
Information is transmitted from a source to a destination through a channel.



- transmitter does encoding  
receiver does decoding
- In databases, a query approximates this model

## Noise

- injected by channel for modeling communication



- receiver filters message from noise
- in databases & other software applications, assume network handles noise
- ability to model *and cope with* noise major strength of information theory
  - ◇ specific impact of errors on bandwidth
- leads to a robust theory of error-correcting codes
  - ◇ Pioneer 10
  - ◇ Ethernet

## Messages

Source can transmit a fixed set of  $n$  messages

e.g. for a weather report,

messages = { *sunny, rain, snow, • • •* }

Identify messages with integers

{ 1, 2, • • •,  $n$  }

Only important aspect of message  $i$  is the probability  $p_i$  that it occurs. Hence, characterize messages by

$$\mathbf{P} = \{ p_1, p_2, \cdot \cdot \cdot, p_n \}$$

Of course,

$$\sum_{i=1}^n p_i = 1$$

e.g. for weather report,  $\mathbf{P}$  depends upon season

$$\mathbf{P}_{\text{January}} = \{ 0.3, 0.05, 0.2, \cdot \cdot \cdot \}$$

$$\mathbf{P}_{\text{July}} = \{ 0.25, 0.25, 10^{-6}, \cdot \cdot \cdot \}$$

Intuitively, the less likely a message, the more information it contains:

- “It is snowing” elicits a *ho-hum* in January but a big response in July.

# Information Value

## Goals:

- formalize “the less likely a message, the more information it contains”
- a function that converts multiplication to addition

- logarithm is the way to do this

$$\log( x \times y ) = \log( x ) + \log( y )$$

all logarithms are to the base 2 in the following

$$\varepsilon \times \log(1/\varepsilon) \rightarrow 0 \text{ as } \varepsilon \rightarrow 0, \text{ so assume } 0 \times \log(1/0) = 0$$

So the information value of receiving message  $i$  is

$$\log( 1 / p_i )$$

or

$$- \log( p_i )$$

## Entropy

The entropy of messages with probability set  $\mathbf{P}$ , denoted  $H(\mathbf{P})$ , is

$$H(\mathbf{P}) = \sum_{i=1}^n p_i \times \log(1/p_i)$$

or, equivalently,

$$H(\mathbf{P}) = - \sum_{i=1}^n p_i \times \log(p_i)$$

Entropy is the expected value of the information gained by receiving one message

“Entropy” is a metaphor based on the mathematical form of the summation

- ★ do not confuse metaphor with actual physical property
- in physics, entropy associated with one state & changes over time
- in communications, entropy associated with set

## Examples of Entropy

Two equally likely messages:

$$\begin{aligned} H(\{0.5, 0.5\}) &= \\ &= \\ &= 1 \end{aligned}$$

- origin of bit (Binary Information Term)

Answer is already known:

$$\begin{aligned} H(\{1, 0\}) &= \\ &= \\ &= 0 \end{aligned}$$

- so no new information

All messages equally likely, so  $p_i = 1/n$  for  $n$  messages:

$$\begin{aligned} H(\{1/n, \dots, 1/n\}) &= \\ &= \\ &= \end{aligned}$$

- when  $n = 2^k$ ,  $k$  bits of information

## Further Examples

Three messages,  $\mathbf{P} = \{ 0.5, 0.25, 0.25 \}$

$H(\mathbf{P}) =$

- what is half a bit?

## Independence

Two sets of messages A and B are independent if knowing a message from one has no influence on the likelihood of a message from the other.

e.g. weather in Bloomington and weather in Perth are independent

weather in Bloomington and weather in Indianapolis are NOT independent

weather now and weather 5 minutes ago are NOT independent

weather today and weather a year ago are independent *provided* that **P** is seasonally adjusted

weather today and weather yesterday ... ?  
this is Indiana, after all

## Messages and Signals

A second weather report contains no information because it has high-level meaning which doesn't change rapidly.

If the "message" is just a symbol (or signal), we presume that each one is independent from the preceeding.

## Independence of Probabilities

Independence of message sets A and B is just the notion of independence from probability theory.

Say

A has probabilities  $\mathbf{P} = \{ p_1, \cdot \cdot \cdot, p_m \}$

B has probabilities  $\mathbf{Q} = \{ q_1, \cdot \cdot \cdot, q_n \}$

then A and B are independent iff probability of message  $i$  from A followed by  $j$  from B is  $p_i \times q_j$

A independent from B implies joint probability set is

$\mathbf{P} \times \mathbf{Q}$

$\times$  is Cartesian product here

## Additivity of Entropy

Fundamental theorem of information theory:

If **P** and **Q** are probabilities of independent sets of messages, then

$$H(\mathbf{P} \times \mathbf{Q}) = H(\mathbf{P}) + H(\mathbf{Q})$$

That is, if messages are truly independent, then their information adds up.

## Proof

$$\mathbf{P} = \{ p_1, \cdot \cdot \cdot, p_m \}$$

$$\mathbf{Q} = \{ q_1, \cdot \cdot \cdot, q_n \}$$

$$\mathbf{P} \times \mathbf{Q} = \{ p_i q_j \}$$

here and below  $i$  ranges over  $1 \cdot \cdot \cdot m$  and  $j$  over  $1 \cdot \cdot \cdot n$

$$H(\mathbf{P} \times \mathbf{Q}) = - \sum_{i,j} p_i q_j \log(p_i q_j)$$

$$=$$

$$=$$

$$=$$

$$=$$

$$\sum p_i = 1$$

$$= H(\mathbf{P}) + H(\mathbf{Q})$$

## Variable Length Encodings

Return to the question “what is half a bit?”

$$\mathbf{P} = \{ 0.5, 0.25, 0.25 \}$$

Use longer codes for less probable messages:

$$\{ \text{“0”}, \text{“1 0”}, \text{“1 1”} \}$$

| <i>msg</i> | <i>prob</i> | <i>code</i> | <i>size</i> | <i>cost</i> |
|------------|-------------|-------------|-------------|-------------|
| A          | 0.5         | 0           | 1           | 0.5         |
| B          | 0.25        | 1 0         | 2           | 0.5         |
| C          | 0.25        | 1 1         | 2           | 0.5         |
| <i>sum</i> |             |             |             | 1.5         |

So “0 1 0 0” encodes “A B A”

Useful?

## Huffman's Algorithm

Specification: Given  $n$  messages with probabilities  $\mathbf{P}$ , assign binary codes to minimize expected bit length.

Input:  $\mathbf{P}$

Output: binary tree with  $n$  leaves, one for each message.

Actual codes given by assigning "0" to each left branch and "1" to each right branch.

Data Structures: nodes of form  $\langle tree, prob \rangle$

set  $S$  of nodes

$tree ::= message \mid ( tree, tree )$

## Code

```
/* initializations */
  S := { <i, p[i]> | 1 <= i <= n }

/* body */
while S has > 1 items do

  pick two items <ta, pa> and <tb, pb>
  in S with lowest probability

  remove <ta, pa> and <tb, pb> from S

  add <( ta, tb ), pa+pb> to S

endwhile

output t, where S = { <t, 1> }
```

## Example

A, B, C, D, E, F, G  
 .25, .125, .0625, .0625, .25, .125, .125

A, B, (C, D), E, F, G  
 .125

A, (B, (C, D)), E, F, G  
 .25

A, (B, (C, D)), E, (F, G)  
 .25

((A, (B, (C, D))), E, (F, G))  
 .5

((A, (B, (C, D))), (E, (F, G)))  
 .5

((A, (B, (C, D))), (E, (F, G)))  
 1.0

Hence code of 0110 maps to C

0 → (A, (B ...))

1 → (B (C, D))

1 → (C D))

0 → C

## Analysis

Message  $i$  with probability  $p_i$  has Huffman code of  $k$  bits, with

$$\lfloor -\log(p) \rfloor \leq k < \lceil -\log(p) \rceil$$

The average cost  $c$  of a Huffman code based on  $\mathbf{P}$  satisfies

$$H(\mathbf{P}) \leq c < H(\mathbf{P}) + 1$$

Easiest to see when each  $p_i = 2^{-k}$ , some  $k$

## Other Applications

To merge files  $F_1, F_2, \dots, F_n$  with sizes  $s_1, s_2, \dots, s_n$

1. associate pseudo-probability

$$p_i = s_i / \sum_j s_j$$

with  $F_i$

2. apply Huffman's algorithm
3. tree indicates merge order

General message: combine smallest first