

**ENTROPY**

DSTA

# MOTIVATIONS

# INFORMATION ENTROPY TODAY

- clearly, the data we work on impacts the quality of our Data Science
- opinion: quantity of data is the main driver of quality of predictions etc.  
[Halevy, Norvig and Pereira, IEEE Int. Sys., 2009]
- but what about quality?
  - which dimensions (columns) are informative?
  - given two suitable datasets, which carries the most valuable information?
  - can the valuable parts be extracted/compressed into a smaller representation?

# FROM INFORMATION CONTENT TO PREDICTIONS

One use of entropy is to identify informative attributes

[Provost-Fawcett, ch. 3]: in a supervised scenario, informative attributes lead to data segmentation thus to the ability to make predictions based on similarity.

# ENTROPY AND DIVERGENCE

# INFORMATION ENTROPY [SHANNON, 1948]

Information channels: to communicate  $n$  distinct signals/commands, how many lamps/semaphores are needed?



It depends on the informative content (surprise) of the signals.

Data compression: how many bits are needed to store a text? Can we compress it?

It depends on frequency of the letters: are they equally likely?

# WEATHER NEWS: LONDON VS. WADI HALFA

Weather forecasts for London are frequent and **nuanced**

Not so in [Wadi Halfa \(Sudan\)](#), one of the driest cities on Earth



A light rain may be surprising in Wadi Halfa but in London?

What if we want to add Weather information at the bus stop?

Weather in Wadi Halfa has **low entropy** thus needs a *small communication channel*: few signals are needed.

London needs a high-capacity communication channel.



# NOTATIONS

# DISTRIBUTIONS

A set of  $n=31$  observations, e.g., [London Weather](#):

*{sunny, sunny, rain, cloudy, sunny, rain ... }*

Count them:

*{sunny: 25, cloudy:2, rain:4}*

Drop the labels then normalize:

divide each value by  $n$ : values will sum to 1:

*{0.8065, 0.0645, 0.1290}*

Mind numerical issues w. rounding etc.

# RAND. VARIABLES

Let  $X$  be a *numerical random variable* and  $x_1, \dots, x_n$  its possible *outcomes*.

Example: throw an unbiased die.

$X_{die}$  will take values over  $1 \dots 6$

$$Pr[X_{die} = x_i] = \frac{1}{6}$$

$$Pr[X_{wheater} = cloudy] = 0.0645$$

# EXPECTATION

$$E[X] = \sum_{i=1}^n x_i \cdot Pr[X = x_i]$$

For numerical outcomes,  $E[X]$  predicts the cumulative effect of repeating obs. on  $X$

$$E[X_{die}] = 3.5$$

For  $n$  throws of a dice expect a cumulative score  $n \cdot 3.5$

# UNDERSTANDING THE DEFINITION

# INFORMATION CONTENT

Captures surprise: the least likely signal carries an important information (e.g., snow alert in London)

$$\frac{1}{Pr[X=x_i]}$$

To smooth the parabolic effect, we ‘log:’

$$I[x_i] = \log_2 \left( \frac{1}{Pr[X=x_i]} \right)$$

The information content of a message is the log-distribution of its **surprise**.

# INFORMATIVE ENTROPY (ETA)

The expectation to receive information

$$H[X] = \sum Pr[X = x_i] \cdot I[x_i]$$

where

$$I[x_i] = \log_2 \left( \frac{1}{Pr[X=x_i]} \right)$$

# FINAL DEFINITION

$$H[X] = - \sum Pr[X = x_i] \cdot \log_2 Pr[X = x_i]$$

Min:  $H[X]=0$ , the system is deterministic, no information in knowing about.

Max:  $H[X] = \log_2 n$  all messages have the same probability.



# IMPLEMENTATION

```
1 def H(distribution):
2     '''computes Shannon's entropy of a distribution: a numpy array'''
3
4     ent = 0.0
5
6     for dim in distribution:
7         if dim == 0.0:
8             ent += 0.0
9         else:
10            ent += dim*math.log(dim, 2)
11
12     return -ent
```

# APPLICATIONS

1. Data compression: we need only  $\lceil H(Dist) \rceil$  bits.
2. How informative a dataset is?
3. Approximation: what is the model distribution that approximates the observed data while **losing as little information as possible?**