Information Theory

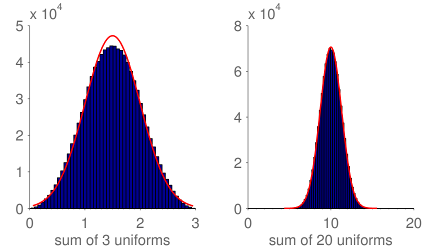
http://www.inf.ed.ac.uk/teaching/courses/it/

Week 2
Information and Entropy

Iain Murray, 2013
School of Informatics, University of Edinburgh

Central Limit theorem

The sum or mean of independent variables with bounded mean and variance tends to a Gaussian (normal) distribution.




```
N=1e6;  hist(sum(rand(3,N),1));  hist(sum(rand(20,N),1));
```

There are a few forms of the Central Limit Theorem (CLT), we are just noting a vague statement as we won't make extensive use of it.

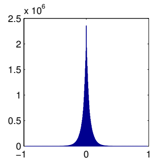
CLT behaviour can occur unreasonably quickly when the assumptions hold. Some old random-number libraries used to use the following method for generating a sample from a unit-variance, zero-mean Gaussian: a) generate 12 samples uniformly between zero and one; b) add them up and subtract 6. It isn't that far off!

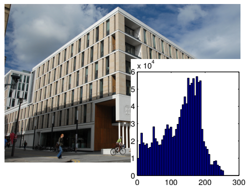
Data from a natural source will usually *not* be Gaussian. The next slide gives examples. Reasons: extreme outliers often occur; there may be lots of strongly dependent variables underlying the data; there may be mixtures of small numbers of effects with very different means or variances.

An example random variable with unbounded mean is given by the payout of the game in the *St. Petersburg Paradox*. A fair coin is tossed repeatedly until it comes up tails. The game pays out $2^{\#\text{heads}}$ pounds. How much would you pay to play? The 'expected' payout is infinite: $1/2 \times 1 + 1/4 \times 2 + 1/8 \times 4 + 1/16 \times 8 + \ldots = 1/2 + 1/2 + 1/2 + 1/2 + \ldots$

Gaussians are not the only fruit

```
xx = importdata('Holst_-_Mars.wav');
hist(double(xx(:)), 400);
```





```
xx = importdata('forum.jpg');
hist(xx(:), 50);
```

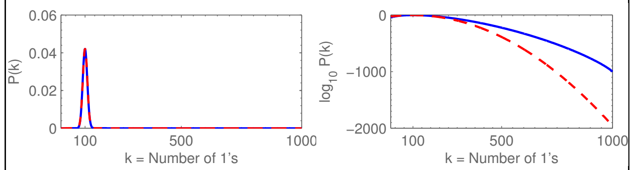
How many 1's will we see?

How many 1's will we see? $P(k) = \binom{N}{k} p^k (1-p)^{N-k}$

Gaussian fit (dashed lines):

$$P(k) \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(k-\mu)^2\right), \quad \mu = Np, \quad \sigma^2 = Np(1-p)$$

(Binomial mean and variance, MacKay p1)



The log-probability plot on the previous slide illustrates how one must be careful with the Central Limit Theorem. Even though the assumptions hold, convergence of the tails is very slow. (The theory gives only "*convergence in distribution*" which makes weak statements out there.) While k, the number of ones, closely follows a Gaussian near the mean, we can't use the Gaussian to make precise statements about the tails.

All that we will use for now is that the mass in the tails further out than a few standard deviations (a few σ) will be small. This is correct, we just can't guarantee that the probability will be quite as small as if the whole distribution actually were Gaussian.

Chebyshev's inequality (MacKay p82, Wikipedia, ...) tells us that:
$P(|k-\mu| \geq m\sigma) \leq \frac{1}{m^2}$,
a loose bound which will be good enough for what follows.

The fact that as $N \to \infty$ all of the probability mass becomes close to the mean is referred to as the *law of large numbers*.

A weighing problem

Find 1 odd ball out of 12

You have a two-pan balance with three outputs:
"left-pan heavier", "right-pan heavier", or "pans equal"

How many weighings do you need to find the odd ball *and* decide whether it is heavier or lighter?

Unclear? See p66 of MacKay's book, but do not look at his answer until you have had a serious attempt to solve it.

Are you sure your answer is right? Can you prove it?
Can you prove it without an extensive search of the solution space?

Weighing problem: bounds

Find 1 odd ball out of 12 with a two-pan balance

There are 24 hypothesis:
ball 1 heavier, ball 1 lighter, ball 2 heavier, ...

For K weighings, there are at most 3^K outcomes:
(left, balance, right), (right, right, left), ...

$3^2 = 9 \Rightarrow$ 2 weighings not enough
$3^3 = 27 \Rightarrow$ 3 weighings *might* be enough

Analogy: sorting (review?)

How much does it cost to sort n items?

There are 2^C outcomes of C binary comparisons

There are $n!$ orderings of the items

To pick out the correct ordering must have:

$C \log 2 \geq \log n! \quad \Rightarrow \quad C \geq \mathcal{O}(n \log n)$ (Stirling's series)

Radix sort is "$\mathcal{O}(n)$", gets more information from the items

Weighing problem: strategy

Find 1 odd ball out of 12 with a two-pan balance

Probability of an outcome is: $\frac{\text{\# hypotheses compatible with outcome}}{\text{\# hypotheses}}$

Experiment	Left	Right	Balance
1 vs. 1	$2/24$	$2/24$	$20/24$
2 vs. 2	$4/24$	$4/24$	$16/24$
3 vs. 3	$6/24$	$6/24$	$12/24$
4 vs. 4	$8/24$	$8/24$	$8/24$
5 vs. 5	$10/24$	$10/24$	$4/24$
6 vs. 6	$12/24$	$12/24$	$0/24$

Weighing problem: strategy

8 hypotheses remain. Find a second weighing where:

3 hypotheses $\Rightarrow$ left pan down
3 hypotheses $\Rightarrow$ right pan down
2 hypotheses $\Rightarrow$ balance

It turns out we can always identify one hypothesis with a third weighing (p69 MacKay for details)

Intuition: outcomes with even probability distributions seem *informative* — useful to identify the correct hypothesis

Measuring information

As we read a file, or do experiments, we get **information**

Very probable outcomes are not informative:
$\Rightarrow$ Information is zero if $P(x)=1$
$\Rightarrow$ Information increases with $1/P(x)$

Information of two independent outcomes add
$\Rightarrow f\big(\frac{1}{P(x)P(y)}\big) = f\big(\frac{1}{P(x)}\big) + f\big(\frac{1}{P(y)}\big)$

Shannon information content: $h(x) = \log \frac{1}{P(x)} = -\log P(x)$

The base of the logarithm scales the information content:
base 2: bits
base e: nats
base 10: bans (used at Bletchley park: MacKay, p265)

$\log \frac{1}{p}$ **is the only 'natural' measure of information based on probability alone.** Derivation non-examinable.

Assume: $f(ab) = f(a) + f(b)$; $f(1) = 0$; f smoothly increases

$$f(a(1+\epsilon)) = f(a) + f(1+\epsilon)$$

Take limit $\epsilon \to 0$ on both sides:

$$f(a) + a\epsilon f'(a) = f(a) + \cancelto{0}{f(1)} + \epsilon f'(1)$$

$$\Rightarrow f'(a) = f'(1)\frac{1}{a}$$

$$\int_1^x f'(a)\,\mathrm{d}a = f'(1)\int_1^x \frac{1}{a}\,\mathrm{d}a$$

$$f(x) = f'(1)\ln x$$

Define $b = e^{1/f'(1)}$, which must be >1 as f is increasing.

$$f(x) = \log_b x$$

We can choose to measure information in any base (>1), as the base is not determined by our assumptions.

Foundations of probability ***(very non-examinable)***

The main step justifying information resulted from $P(a,b) = P(a)\,P(b)$ for independent events. Where did *that* come from?

There are various formulations of probability. Kolmogorov provided a measure-theoretic formalization for frequencies of events.

Cox (1946) provided a very readable rationalization for using the standard rules of probability to express beliefs and to incorporate knowledge: `http://dx.doi.org/10.1119/1.1990764`

There's some (I believe misguided) arguing about the details. A sensible response to some of these has been given by Van Horn (2003) `http://dx.doi.org/10.1016/S0888-613X(03)00051-3`

Ultimately for both information and probability, the main justification for using them is that they have proven to be hugely useful. While one can argue forever about choices of axioms, I don't believe that there are other compelling formalisms to be had for dealing with uncertainty and information.

Information content vs. storage

A 'bit' is a symbol that takes on two values.
The 'bit' is also a unit of information content.

Numbers in 0–63, e.g. $47 = 101111$, need $\log_2 64 = 6$ bits

If numbers 0–63 are equally probable, being told the number has information content $-\log \frac{1}{64} = 6$ bits

The binary digits are the answers to six questions:

1: is $x \geq 32$?
2: is $x \bmod 32 \geq 16$?
3: is $x \bmod 16 \geq 8$?
4: is $x \bmod 8 \geq 4$?
5: is $x \bmod 4 \geq 2$?
6: is $x \bmod 2 = 1$?

Each question has information content $-\log \frac{1}{2} = 1$ bit

---

Fractional information

A dull guessing game: (submarine, MacKay p71)

Q. Is the number 36?
A. $a_1 =$ No.
$h(a_1) = \log \frac{1}{P(x\neq 36)} = \log \frac{64}{63} = 0.0227$ bits

Remember: $\log_2 x = \frac{\ln x}{\ln 2}$

Q. Is the number 42?
A. $a_2 =$ No.
$h(a_2) = \log \frac{1}{P(x\neq 42\,|\,x\neq 36)} = \log \frac{63}{62} = 0.0231$ bits

Q. Is the number 47?
A. $a_3 =$ Yes.
$h(a_3) = \log \frac{1}{P(x=47\,|\,x\neq 42, x\neq 36)} = \log \frac{62}{1} = 5.9542$ bits

Total information: $5.9542 + 0.0231 + 0.0227 = 6$ bits

Entropy

Improbable events are very informative, but don't happen very often! How much information can we *expect*?

Discrete sources:

Ensemble: $X = (x, \mathcal{A}_X, \mathcal{P}_X)$
Outcome: $x \in \mathcal{A}_x$, $\quad p(x=a_i) = p_i$
Alphabet: $\mathcal{A}_X = \{a_1, a_2, \ldots, a_i, \ldots a_I\}$
Probabilities: $\mathcal{P}_X = \{p_1, p_2, \ldots, p_i, \ldots p_I\}$, $\quad p_i > 0$, $\quad \sum_i p_i = 1$

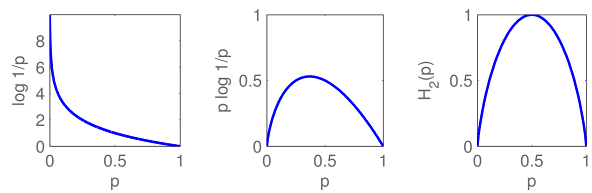
Information content:
$h(x=a_i) = \log \frac{1}{p_i}$, $\qquad h(x) = \log \frac{1}{P(x)}$

Entropy:
$H(X) = \sum_i p_i \log \frac{1}{p_i} = \mathbb{E}_{\mathcal{P}_X}[h(x)]$
average information content of source, also "the uncertainty of X"

Binary Entropy

Entropy of Bernoulli variable:

$$H(X) = H_2(p) = p_1 \log \frac{1}{p_1} + p_2 \log \frac{1}{p_2}$$
$$= -p\log p - (1-p)\log(1-p)$$



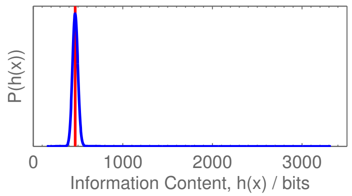
Plots take logs base 2. We define $0\log 0 = 0$

Distribution of Information

Extended Ensemble X^N: N independent draws from X

$\mathbf{x}$ a length-N vector containing a draw from X^N

Bernoulli example: $N = 10^3$, $p = 0.1$, $H(X) = 0.47$ bits



The information content of each element, $h(x_n)$, is a random variable. This variable has mean $H(X)$, and some finite variance σ^2.

Mean and width of the curve: The total information content of a block: $h(\mathbf{x}) = \sum_n h(x_n)$ is another random variable with mean $NH(X)$, shown in red, and variance $N\sigma^2$ or standard deviation $\sqrt{N}\sigma$. (All of the above is true for general extended ensembles, not just binary streams.)

The range of the plot: The block with maximum information content is the most surprising, or least probable block. In the Bernoulli example with $p{=}0.1$, '`1111...111`' is most surprising, with $h(\mathbf{x}){=}Nh(1){=}N \log \frac{1}{0.1}$. Similarly the least informative block, is the most probable. In the example $Nh(0){=}N \log \frac{1}{0.9}$. Remember to take logs base 2 to obtain an answer in bits. Neither of these blocks will *ever* be seen in practice, even though `0000...000` *is* the most probable block.

Only blocks with information contents close to the mean are 'typical'.

Define the *typical set*, T, to be all blocks with information contents a few standard deviations away from the mean:

$h(\mathbf{x}) \in [NH{-}m\sigma\sqrt{N}, NH{+}m\sigma\sqrt{N}]$ for some $m > 0$.

(Actually a family of typical sets for different choices of m.)

We only need to count the typical set: Chebyshev's inequality (see MacKay p82, Wikipedia, ...) bounds the probability that we land outside the typical set.

$$P(|h(\mathbf{x}) - NH| \geq m\sigma\sqrt{N}) \leq \frac{1}{m^2}$$

We can pick m so that the typical set is so large that the probability of landing outside it is negligible. Then we can compress almost every file we see into a number of bits that can index the typical set.

How big is the typical set? Number of elements: $|T|$

Probability of landing in set ≤ 1
Probability of landing in set $\geq |T| p_{\min}$, where $p_{\min} = \min_{\mathbf{x} \in T} p(\mathbf{x})$
Therefore, $|T| < \frac{1}{p_{\min}}$

Block with smallest probability $p_{\min}$ has information $NH + m\sigma\sqrt{N}$.
$p_{\min} = 2^{-NH - m\sigma\sqrt{N}}$
$|T| < 2^{NH + m\sigma\sqrt{N}}$
Number of bits to index typical set is $NH + m\sigma\sqrt{N}$.
Dividing by the block length, N we see we need:

$$H + m\sigma/\sqrt{N} \text{ bits/symbol}, \to H \text{ as } N \to \infty$$

For any choice m, in the limit of large blocks, we can encode the typical set (and for large enough m, any file we will see in practice) with $H(X)$ bits/symbol.

Can we do better?

Motivation: The above result put a loose bound on the probability of being outside T, so we might have made it bigger than necessary. Then we put a loose bound on the number of items, so we assumed it was even bigger than that. *Maybe* we could use many fewer bits per symbol than we calculated? (Amazingly, the answer is that we can't.)

We assume there is a smaller useful set S, which we could encode with only $(1{-}\epsilon)H$ bits/symbol. For example, if $\epsilon{=}0.01$ we would be trying to get a 1% saving in the number of bits for strings in this set.

The size of S is $|S| = 2^{N(1-\epsilon)H}$

Some of S will overlap with T, and some might be outside. But we know that the total probability outside of T is negligible (for large m). The probability mass of elements inside T is less than $|S| p_{\max}$, where $p_{\max}$ is the probability of the largest probability element of T.
$p_{\max} = 2^{-NH + m\sigma\sqrt{N}}$
$p(\mathbf{x} \in S) \leq |S| p_{\max} + \text{tail mass outside } T$

$p(\mathbf{x} \in S) \leq 2^{N(-\epsilon H + m\sigma/\sqrt{N})} + \text{tail mass outside } T$

As $N \to \infty$ the probability of getting a block in S tends to zero for any m. The smaller set is useless.

At least H bits/symbol are required to encode an extended ensemble.

On average, no compressor can use fewer than H bits per symbol (applied to length-N blocks, it wouldn't be using enough bits)

Where now?

A block of variables can be compressed into $H(X)$ bits/symbol, but no less

Where do we get the probabilities from?

How do we *actually* compress the files?
We can't explicitly list 2^{NH} items!
Can we avoid using enormous blocks?

Numerics note: $\log \sum_i \exp(x_i)$

$\binom{N}{k}$ blows up for large N, k; we evaluate $l_{N,k} = \ln \binom{N}{k}$

Common problem: want to find a sum, like $\sum_{k=0}^{t} \binom{N}{k}$

Actually we want its log:

$$\ln \sum_{k=0}^{t} \exp(l_{N,k}) = l_{\max} + \ln \sum_{k=0}^{t} \exp(l_{N,k} - l_{\max})$$

To make it work, set $l_{\max} = \max_k l_{N,k}$. `logsumexp` functions are frequently used

I needed this trick when numerically exploring block codes:

For a range of t we needed to sum up: a) the number of strings with $k = 0..t$; and b) the probability mass associated with those strings. The log of the number of strings says how many bits, C_1 was needed to index them. If the probability mass is close to one, that will also be close to the expected length needed to encode random strings.

For both sums we need the log of the sum of some terms, where each term is available in log form. The next slide demonstrates this for problem a), but the technique readily applies to problem b) too.

The bumps are very well behaved: to what extent can we assume they are Gaussian due to central limit arguments?