And now for something completely different

David Aldous

July 27, 2012

I don't do constants. *Rick Durrett.*

I will describe a topic with the paradoxical features

- Constants matter – it's all about the constants.
- Factors of e^N don't matter – we can just ignore them.

Each of the three words **entropy, complexity, information** has many different meanings; any statement involving two of these words is (at best) true in only some very restricted context.

I will use the word entropy only with its most elementary meaning: for any probability distribution $\mathbf{p} = (p_s)$ on any finite set S, its *entropy* is the number

$$\mathrm{ent}(\mathbf{p}) = -\sum_s p_s \log p_s. \tag{1}$$

Shannon et al had the brilliant idea of combining a simple math lemma and a wild leap of faith.

Lemma

Write $\mathbf{B}$ *for the set of binary strings* $\mathbf{b} = b_1 b_2 \dots b_m$ *and* $\mathrm{len}(\mathbf{b}) = m$ *for the length of a string. Given* $\mathbf{p}$ *as above, there exists a coding (a 1-1 function)* $f_{\mathbf{p}} : S \to \mathbf{B}$ *such that for* X *with distribution* $\mathbf{p}$

$$\mathrm{ent}_2(\mathbf{p}) \leq \mathbb{E}\ \mathrm{len}(f_{\mathbf{p}}(X)) \leq \mathrm{ent}_2(\mathbf{p}) + 1$$

and no coding can improve on the lower bound.

Wild Leap of Faith

Assume that English text can be modeled as a "random" process – random in the specific sense of a stationary random process.

Where does this get us? Skipping over a lot of interesting stuff (will return later) . . . . . .

Data compression

Take English text, length N letters.
Code into binary in most naive way (letter-by-letter, ASCII).
Length of coded text $= c_1 N$ bits.

Code into binary in most sophisticated way (Lempel-Ziv and sequels).
Length of coded text $= c_2 N$ bits.

	uncompressed	compressed
first half of Don Quixote	1109963	444456
second half of Don Quixote	1109901	451336

1. This is one area of "applied probability" that actually works (makes verifiable predictions) even though it *a priori* has nothing to do with probability.
2. Loosely, **entropy rate** of English is (at most) c_2 bits per letter. If you could find an algorithm to compress to less than 40% then .....
3. (Relevant to this talk) **it's all about the constants**.

4. (Somewhat relevant to this talk). One can seek to compress any data (not just sequences). For photos/videos/audio one typically uses lossy compression (.jpg etc). My topic is lossless compression for certain mathematical objects more interesting (to me) than sequences.

Here are some examples to get into the spirit. The uniform distribution on any M-element set has entropy $\log_2 M$, so we should be able to code elements of the set in about $\log_2 M$ bits.

So . . . . . how do we code

- a permutation of $\{1, 2, \ldots, N\}$
- a tree on vertices $\{1, 2, \ldots, N\}$ (there are N^{N-2} of these)

How many bits are required to store one uniform random pick of

- a permutation of $\{1, 2, \ldots, N\}$ (there are $N!$ of these)
- a tree on vertices $\{1, 2, \ldots, N\}$ (there are N^{N-2} of these)

Asymptotically

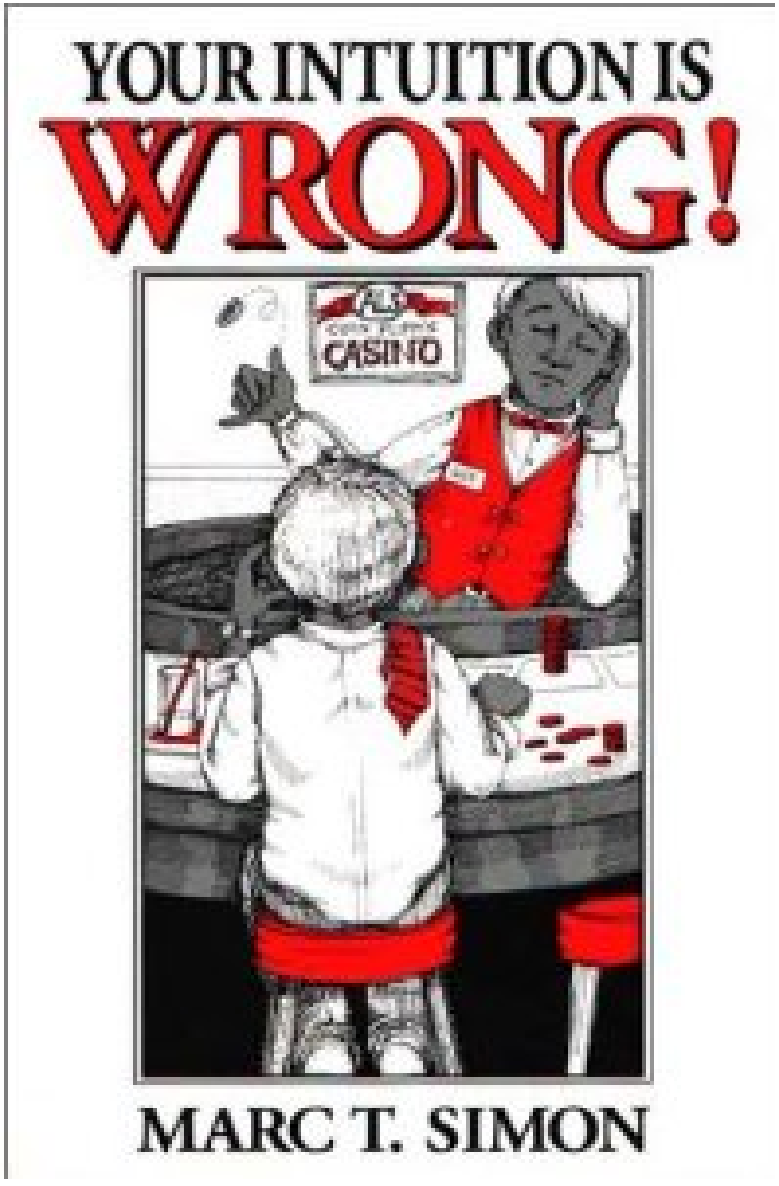
$$\log N^{N-2} \sim 1 \times N \log N$$

$$\log N! \sim \log(N/e)^N \sim 1 \times N \log N.$$

So (relevant to this talk) ...

- constants matter
- both constants $= 1$
- the factor e^N doesn't matter

... welcome to the $N \log N$ entropy world! I will talk about settings where entropy $\sim cN \log N$. This "entropy rate" c is robust to model details, and indeed doesn't depend on the base of logarithms.


YOUR INTUITION IS
WRONG!
CASINO
MARC T. SIMON

Just for fun . . . two examples of (non-uniform) random permutations of a N-card deck. Here "7" is fixed and $N \to \infty$.

Model A: 7 random riffle shuffles.
Model B: Separate into 7 piles, randomize each pile, replace in original order of piles.

Intuition may be hard but calculation is easy:
$\mathrm{ent}(B) \sim 1 \ \cdot N \log N$
$\mathrm{ent}(A) \approx 7N \sim 0 \cdot N \log N$.

In Model B, to reduce entropy to $(1-\varepsilon)N \log N$ we need N^{ε} piles.
In Model A, to increase entropy to $\varepsilon N \log N$ we need $c(\varepsilon) \log N$ shuffles.

What is a network?

- A **graph** is a well-defined mathematical object – vertices and edges etc
- A **network** is a graph with context-dependent extra structure. *David Aldous*

I want to consider some simple-but-interesting notion of "extra structure".

Consider a graph with

- N vertices
- $O(1)$ average degree
- vertices have distinct "names", strings of length $O(\log N)$ from a fixed finite alphabet.

Envisage some association between vertex names and graph structure, as in

- phylogenetic trees on species
- road networks.

I claim this is the "interesting" setting for data compression of sparse graphs, because the "entropy" of both the graph structure and of the names has the same order, $N \log N$. [Will say more later].

What is in the existing literature?

A: "More mathematical". Topic called "graph entropy" studies the number of automorphisms of a N-vertex unlabelled graph. See the 2012 survey by Szpankowski - Choi *Compression of graphical structures: Fundamental limits, algorithms, and experiments*.

B: "More applied".
Paolo Boldi and Sebastiano Vigna (2003). The webgraph framework I: Compression techniques. In Proc. of the Thirteenth International World Wide Web Conference.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, Michael Mitzenmacher, Alessandro Panconesi, and Prabhakar Raghavan (2009). On compressing social networks. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining.

Our formal setting.

Define a finite set $S = S(N, A, \beta, \alpha)$; an element of S is a network with

- N vertices
- ave degree $\leq \alpha$ (at most $\alpha N/2$ edges)
- finite alphabet **A** of size A
- vertices have distinct names – strings from **A** of length $\leq \beta \log_A N$.

Here $A \geq 2$, $0 < \alpha < \infty$, $1 < \beta < \infty$ are fixed and we study as $N \to \infty$.

Easier than you might think to see how big S is:

$$\log |S(N, A, \beta, \alpha)| \sim (\beta - 1 + \tfrac{\alpha}{2}) N \log N.$$

For some given model of random network $\mathcal{G}_N$ we expect

$$\mathrm{ent}(\mathcal{G}_N) \sim cN \log N$$

for some model-dependent entropy rate $0 \leq c \leq (\beta - 1 + \frac{\alpha}{2})$.

In this setting there are $\geq$ two lines of research you might try.

Clearly do-able: Invent probability models and calculate their entropy rate.

Clearly not do-able: Design an algorithm which, given a realization from a probability model, compresses optimally (according to the entropy of the probability model) without knowing what the model is ("universal", like Lempel-Ziv for sequences).

(Repeat previous slide).
For any probability distribution $\mathbf{p} = (p_s)$ on any finite set S, its *entropy* is the number

$$\mathrm{ent}(\mathbf{p}) = -\sum_s p_s \log p_s. \tag{2}$$

Shannon et al had the brilliant idea of combining a simple math lemma and a wild leap of faith.

Lemma

Write $\mathbf{B}$ *for the set of binary strings* $\mathbf{b} = b_1 b_2 \dots b_m$ *and* $\mathrm{len}(\mathbf{b}) = m$ *for the length of a string. Given* $\mathbf{p}$ *as above, there exists a coding (a 1-1 function)* $f_{\mathbf{p}} : S \to \mathbf{B}$ *such that for* X *with distribution* $\mathbf{p}$

$$\mathrm{ent}_2(\mathbf{p}) \leq \mathbb{E}\ \mathrm{len}(f_{\mathbf{p}}(X)) \leq \mathrm{ent}_2(\mathbf{p}) + 1$$

and no coding can improve on the lower bound.

Wild Leap of Faith

Assume that English text can be modeled as a "random" process – random in the specific sense of a stationary random process.

The small print is:
The coding function $f_{\mathbf{p}}$ depends on $\mathbf{p}$
The coding function $f_{\mathbf{p}}$ may be arbitrarily complex.

Example. Suppose, from a library of about one million books, you pick one book uniformly at random and write out the entire text of that book. What is the entropy of what you write?

Answer [according to our definition]: 20 bits.

On the other hand, we saw that the entropy of the text of *Don Quixote* was about 900,000 bits.

(ignoring ergodicty issues . . .)

(Shannon). A stationary random **A**-valued process $(X_1, X_2, \ldots)$ has entropy rate

$$H := \lim_{k \to \infty} k^{-1} \mathrm{ent}(X_1, \ldots, X_k).$$

(Lempel-Ziv). There exists a $1-1$ function
$\mathtt{compress}$: { finite strings from **A**} $\to$ { finite binary strings }
such that, for every stationary $(X_1, X_2, \ldots)$,

$$n^{-1} \mathbb{E} \ \mathrm{len}(\mathtt{compress}(X_1, \ldots, X_n)) \to H.$$

To appreciate this, consider a sequence of uniform random bits derived from
(i) physical randomness
(ii) a standard random number generator.

Alas the "universality" that Lempel-Ziv brings to data compression for sequences seems unavailable in broader contexts, and in particular there are several reasons why universal algorithms are impossible in our context. This is (presumably) why our setting hasn't been studied in IEEE-IT.

A less ambitious project is to invent some heuristic compression algorithm in our setting, and then check by theory/simulation that it does compress optimally for samples from a collection of probability models. So (my sales pitch to EE folks) we have some motivation for

Clearly do-able project: Invent probability models and calculate their entropy rate.

I'll show a few results. Write $\mathcal{E}(p)$ for entropy of Ber(p):

$$\mathcal{E}(p) = -p \log p - (1-p) \log(1-p) \sim p \log(1/p) \text{ as } p \downarrow 0.$$

Sparse Erdős-Rényi, default binary names

Model. N vertices, whose names are the integers $1, \ldots, N$ written as binary strings of length $\lceil \log_2 N \rceil$. Each of the $\binom{N}{2}$ possible edges is present independently with probability α/N, where $0 < \alpha < \infty$.

Entropy rate formula. $c(\alpha) = \frac{\alpha}{2}$.

Proof. The entropy equals $\binom{N}{2}\mathcal{E}(\alpha/N)$; letting $N \to \infty$, this is

$$\sim \frac{N^2}{2} \; \frac{\alpha}{N} \log \frac{N}{\alpha} \sim \frac{\alpha}{2} \; N \log N.$$

Sparse Erdős-Rényi, random A-ary names

Model. As above, N vertices, and each of the $\binom{N}{2}$ possible edges is present independently with probability α/N. Take $L_N \sim \beta \log_A N$ for $1 < \beta < \infty$ and take the vertex names as a uniform random choice of N distinct A-ary strings of length L_N.

Entropy rate formula. $c(\alpha, \beta) = \beta - 1 + \frac{\alpha}{2}$.

Proof. The entropy equals $\log \binom{A^{L_N}}{N} + \binom{N}{2}\mathcal{E}(\alpha/N)$. The first term $\sim (\beta - 1)N \log N$ by (3) and the second term $\sim \frac{\alpha}{2} N \log N$ as in the previous model.

A useful fact, most easily remembered by considering a coding:

$$\log \binom{M}{K} \sim K \log(M/K) \text{ over } 1\text{`} \ll K \ll M. \tag{3}$$

`begin repeat`

Sparse Erdős-Rényi, random A-ary names

Model. As above, N vertices, and each of the $\binom{N}{2}$ possible edges is present independently with probability α/N. Take $L_N \sim \beta \log_A N$ for $1 < \beta < \infty$ and take the vertex names as a uniform random choice of N distinct A-ary strings of length L_N.

Entropy rate formula. $c(\alpha, \beta) = \beta - 1 + \frac{\alpha}{2}$.

Proof. The entropy equals $\log \binom{A^{L_N}}{N} + \binom{N}{2}\mathcal{E}(\alpha/N)$. The first term $\sim (\beta - 1)N \log N$ by (3) and the second term $\sim \frac{\alpha}{2} N \log N$ as in the previous model.

`end repeat`

Remark. One might have naively guessed that the formula would involve β instead of $\beta - 1$, on the grounds that the entropy of the sequence of names is $\sim \beta N \log N$, but this is the rate in a third model where a vertex name is a pair $(i, \mathbf{a})$, where $1 \le i \le N$ and $\mathbf{a}$ is the random string. That is, $N!$ different graphs correspond to the same graph in our model above, and this changes β to $\beta - 1$. This model distinction becomes more substantial for the model to be studied after next digression.

There are rules for manipulating entropy, analogous to rules for manipulating conditional expectation. In particular, writing $\mathrm{ent}(Y|X)$ for the entropy of the conditional distribution,

$$\mathrm{ent}(X, Y) = \mathrm{ent}(X) + \mathbb{E}\mathrm{ent}(Y|X).$$

This means it is often easy to calculate entropy for a random structure which is constructed sequentially.

Nest slide is the first model I can invent whose analysis is not so straightforward.

Bathtub problem: Think of other models – recall we want dependence in sense that the names of adjacent vertices are more similar than non-adjacent vertices.

This reminded me of favorite paper title: *Coevolution to the edge of chaos*.

In outline, the graph structure is again sparse Erdős-Rényi $\mathcal{G}(N, \alpha/N)$, but we construct it inductively over vertices, and make the vertex-names copy parts of the names of previous vertices that the current vertex is linked to. Here are the details.

Model. Take $L_N \sim \beta \log_A N$ for $1 < \beta < \infty$. Vertex 1 is given a uniform random length-L_N A-ary name. For $1 \leq n \leq N-1$:

vertex $n+1$ is given an edge to each vertex $i \leq n$ independently with chance α/N. Write $Q_n \geq 0$ for the number of such edges, and $\mathbf{a}^1, \ldots, \mathbf{a}^{Q_n}$ for the names of the linked vertices. Take an independent uniform random length-L_N A-ary string $\mathbf{a}^0$. Assign to vertex $n+1$ the name obtained by, independently for each coordinate $1 \leq u \leq L_N$, making a uniform random choice from the $Q_n + 1$ letters $a_u^0, a_u^1, \ldots, a_u^{Q_n}$.

This gives a family $(\mathcal{G}_N)$ parametrized by (A, β, α).

Entropy rate formula for $(\mathcal{G}_N^{ord})$.

$$-1 + \frac{\alpha}{2} + \beta \sum_{k \geq 0} \frac{\alpha^k J_k(\alpha) h_A(k)}{k! \log A}$$

where

$$J_k(\alpha) := \int_0^1 x^k e^{-\alpha x} dx$$

and the constants $h_A(k)$ are defined below.

We first study the "ordered" model. The unconditional distribution of each vertex-name is uniform on length-L_N words.

Suppose vertex $n+1$ links to exactly two previous vertices. Conditional on these being a particular pair $1 \le i < j \le n$, with names $\mathbf{a}^i, \mathbf{a}^j$, the contribution to entropy is exactly

$$L_N \sum_{(a,a') \in \mathbf{A} \times \mathbf{A}} g_2(a,a')\ \mu^{i,j}(a,a')$$

where

$$\begin{aligned} g_2(a,a') &= \mathcal{E}_A(\tfrac{A+1}{3A}, \tfrac{A+1}{3A}, \tfrac{1}{3A}, \tfrac{1}{3A}, \ldots\ldots \tfrac{1}{3A}) \text{ if } a' \neq a \\ &= \mathcal{E}_A(\tfrac{2A+1}{3A}, \tfrac{1}{3A}, \tfrac{1}{3A}, \tfrac{1}{3A}, \ldots\ldots \tfrac{1}{3A}) \text{ if } a' = a \end{aligned}$$

and where $\mathcal{E}_A(\mathbf{p})$ is the entropy of a distribution $\mathbf{p} = (p_1, \ldots, p_A)$ and $\mu^{i,j}(a,a')$ is the empirical distribution of the pairs $(a_1^i, a_1^j), \ldots, (a_A^i, a_A^j)$. The technical issue is proving that dependence between names is sufficiently small that the sum above behaves as if they were independent,

$$\sim h_A(2) := A^{-2} \sum_{(a,a') \in \mathbf{A} \times \mathbf{A}} g_2(a,a').$$

Averaging over choices of i, j we might be copying from, the contribution to entropy from the event "link to 2 previous vertices" is

$$\sim \beta \log_A N \times \alpha^2 h_A(2) \frac{\binom{n}{2}}{N^2} \exp(-\alpha n/N).$$

Now we just sum over different numbers k of linked-to vertices and sum over vertices $1 \leq n \leq N-1$. We end up with

$$\frac{\alpha}{2} + \beta \sum_{k \geq 0} \frac{\alpha^k J_k(\alpha) h_A(k)}{k! \log A}$$

which is the original stated formula without the -1 term.

We want entropy rate for the "unordered" model, where we don't see the order $n = 1, 2, \ldots, N$ of vertices. The issue reduces to a question about the sparse Erdős-Rényi graph.

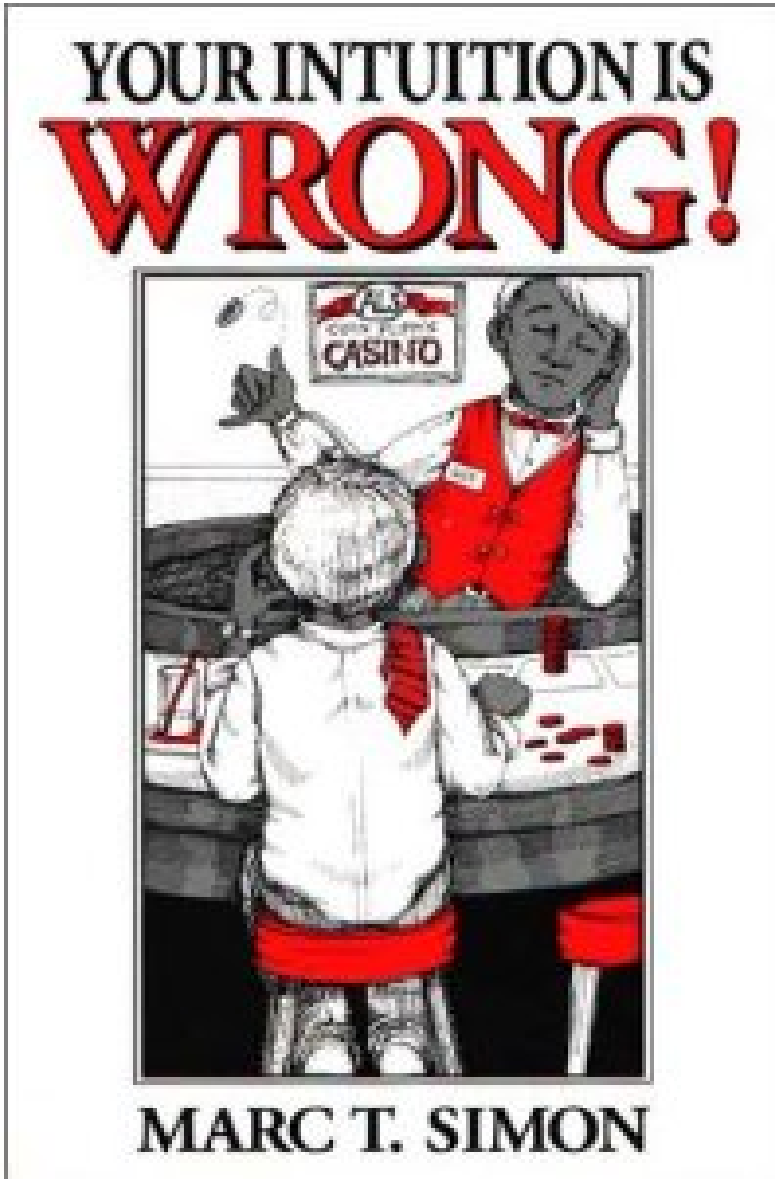
Take a typical realization of $G(N, \alpha/N)$ on vertices $\{1, 2, \ldots, N\}$. Direct the edges $j \to i$ where $i < j$. Remove vertex-labels.

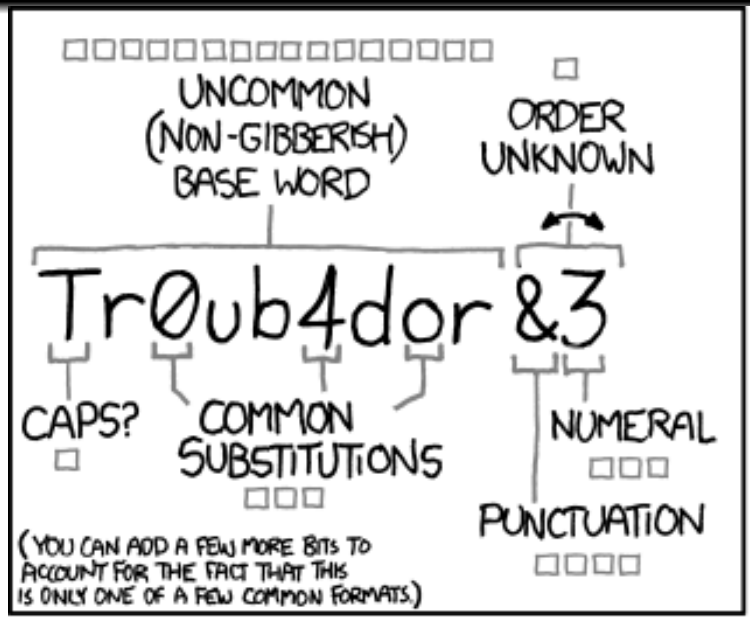
Question: How many of the $N!$ possible re-labelings are consistent with the edge-directions?

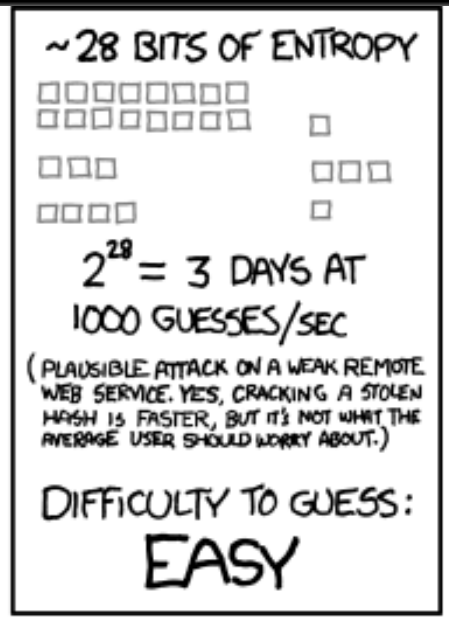
Answer:

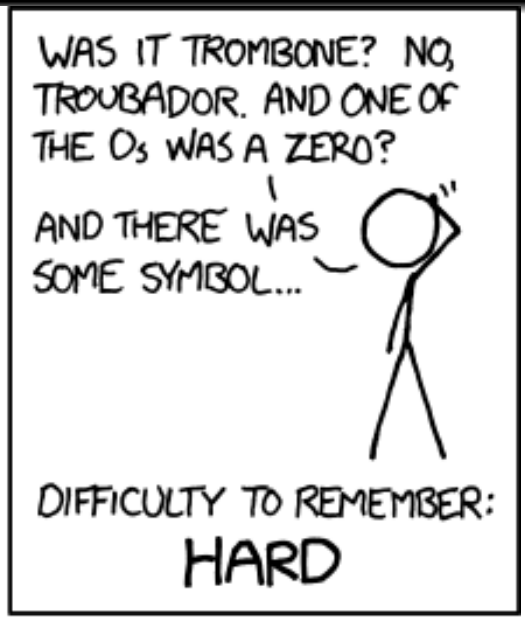
$$\log(\text{number of consistent relabellings}) \sim 1 \cdot N \log N$$

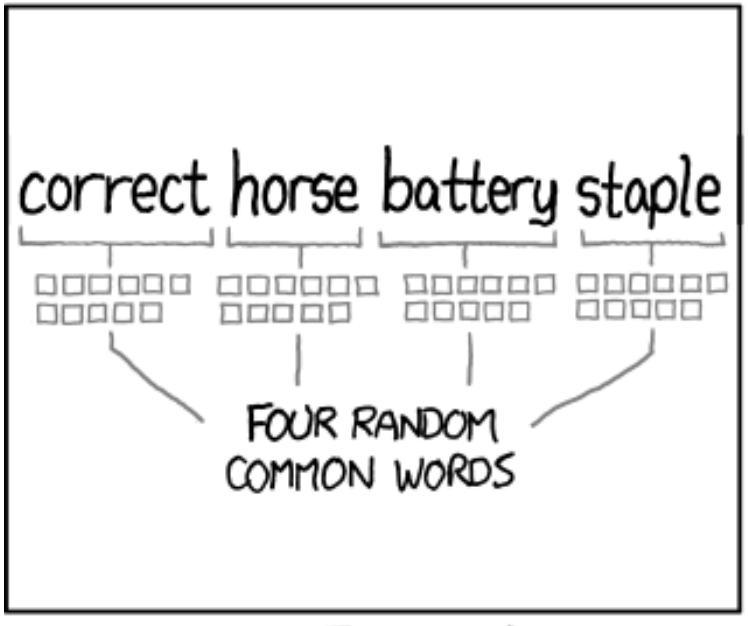
because (informally) the probability that a random relabelling works is **only** exponentially small in N.

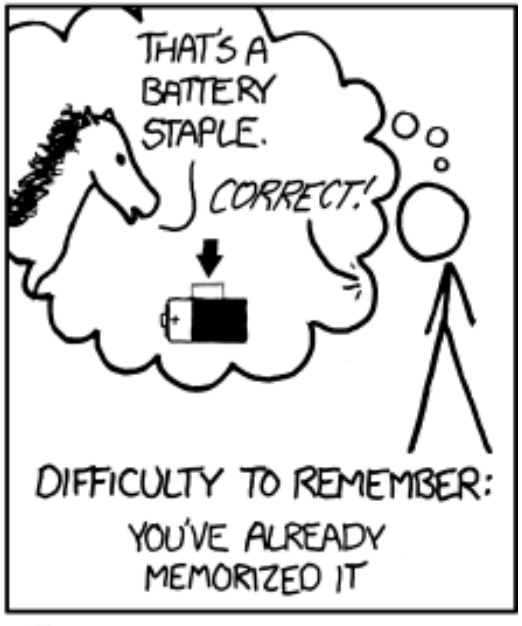

YOUR INTUITION IS
WRONG!
CASINO
MARC T. SIMON











THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

For more about information theory, read Thomas Cover's 1990 Shannon Lecture.