Entropy, which is not to be confused with entropy from physics, comes from **information theory**. Information theory is based on probability and statistics, and deals with the transmission, processing, utilization, and extraction of information. A key concept in information theory is the notion of a **bit of information**. One bit of information is one unit of information.

We can represent a bit of information as a binary number because it either has the value `1` or `0`. Suppose there's an equal probability of tomorrow being sunny (`1`) or not sunny (`0`). If I tell you that it will be sunny, I've given you one bit of information.

We can also think of entropy in terms of information. If we flip a coin where both sides are heads, we know upfront that the result will be heads. We gain no new information by flipping the coin, so entropy is 0. On the other hand, if the coin has a heads side and a tails side, there's a 50% probability that it will land on either. Thus, flipping the coin gives us one bit of information -- which side the coin landed on.

Entropy can be much more complex, especially when we get to cases with more than two possible outcomes, or differential probabilities. A deep understanding of entropy isn't necessary for constructing decision trees, however. If you'd like, you can read more about entropy at Wikipedia.

The formula for entropy looks like this:

$$-\sum_{i=1}^{c} \mathbf{P}(x_i)\,\mathbf{log}_b\,\mathbf{P}(x_i)$$

➔ Vu que l'on calcule l'entropie de la variable **prédite** (colonne « cible »), on pourrait noter plutôt :

$$\mathbf{H}(Y)\ \text{(ou } \mathbf{H}(T) \text{ dans la suite)} = -\sum_{i=1}^{c} \mathbf{P}(y_i)\,\mathbf{log}_b\,\mathbf{P}(y_i)$$

We iterate through each unique value in a single column (in this case, `high_income`), and assign it to ~~i~~ x_i. We then compute the probability of that value occurring in the data (P(x_i)). Next we do some multiplication, and sum all of the values together. b is the base of the logarithm. We commonly use the value `2` for this, but we can also set it to `10` or another value.

Let's say we have this data:

age	high_income
25	1
50	1
30	0
50	0
80	1

We could compute its entropy like this:

$$-\sum_{i=1}^{c} \mathrm{P}(x_i)\log_b \mathrm{P}(x_i) = -\left(\frac{2}{5}\log_2\frac{2}{5}+\frac{3}{5}\log_2\frac{3}{5}\right) \approx 0.97$$

We get less than one "bit" of information -- only `.97` -- because there are slightly more `1`s in the sample data than `0`s. This means that if we were predicting a new value, we could guess that the answer is `1` and be right more often than wrong (because there's a `.6` probability of the answer being `1`). Due to this prior knowledge, we gain less than a full "bit" of information when we observe a new value.

➔

Hi,

On page 8 of the "Introduction to Decision Trees" mission, you write: "If you'd like, you can read more about entropy at Wikipedia."

The URL link references the "Entropy" article (in *physics*!), I think it should rather be:

https://en.wikipedia.org/wiki/Entropy_(information_theory)

Oops! Good catch — I've logged this to be fixed by our content team.

Best,

Dustin

➔ *https://en.wikipedia.org/wiki/Entropy_(information_theory)*

Definition

Named after Boltzmann's H-theorem, Shannon defined the entropy H (Greek capital letter eta) of a discrete random variable X with possible values $\{x_1, ..., x_n\}$ and probability mass function P(X) as:

$$\mathbf{H}(X) = \mathrm{E}[\mathrm{I}(X)] = \mathbf{E}[-\mathbf{ln}(\mathbf{P}(X))].$$

Here E is the expected value operator, and I is the information content of X.[4][5] I(X) is itself a random variable.

The entropy can explicitly be written as

$$\mathbf{H}(X) = \sum_{i=1}^{n} \mathrm{P}(x_i)\,\mathrm{I}(x_i) = -\sum_{i=1}^{n} \mathbf{P}(x_i)\,\mathbf{log}_b\,\mathbf{P}(x_i),$$

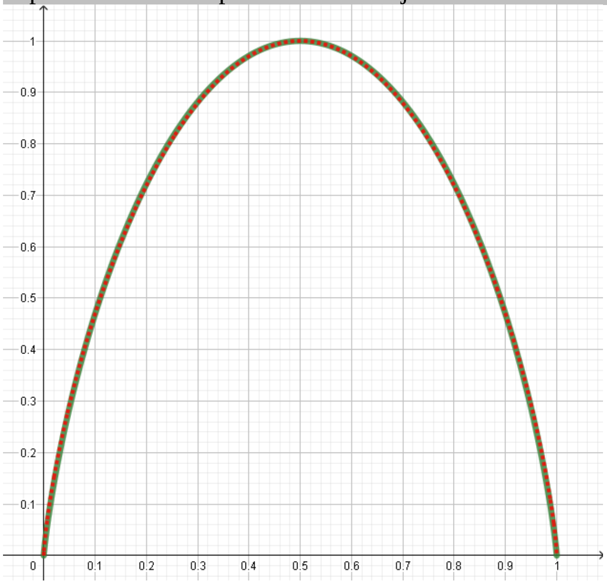
where b is the base of the logarithm used. Common values of b are 2, Euler's number e, and 10, and the corresponding units of entropy are the bits for b = 2, nats for b = e, and bans for b = 10.[6]

One may also define **the conditional entropy** of two events X and Y taking values x_i and y_j respectively, as

$$\mathbf{H}(X|Y) = -\sum_{i,j} p(x_i, y_j)\,\mathbf{log}\frac{p(x_i, y_j)}{p(y_j)}$$

where $p(x_i, y_j)$ is the probability that $X = x_i$ and $Y = y_j$. This quantity should be understood as **the amount of randomness in the random variable X given the event Y**.

➔ Ci-dessous, la courbe de **l'entropie binaire** (ou **entropie de Bernoulli**) $x \longmapsto \mathbf{-(x \log_2 x + (1-x) \log_2(1-x))}$ dont le **maximum**, sans surprise, est atteint pour $x = 1 - x = \frac{1}{2}$.
Pour la petite curiosité, j'ai ajouté en pointillés le remarquable ajustement $x \longmapsto \mathbf{(4x(1-x))^{\alpha}}$ **pour α = 0,74** (après avoir remarqué que la parabole obtenue pour $\alpha = 1$ est déjà une assez bonne approximation).



➔ Notons le *très vague* « air de famille » entre la formule de l'entropie binaire et celle de **la fonction « coût » pour un réseau de neurones à deux couches** :

$$\boldsymbol{J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} (y_i \log(h_\Theta(x_i)) + (1 - y_i) \log(1 - h_\Theta(x_i)))} \text{ où } \boldsymbol{h_\Theta(x_i) = \frac{1}{1 + \exp(-\Theta^T x_i)}}.$$

➔ À ce propos, dans la leçon « Machine Learning in Python - Intermediate » (p. 40), je m'étais posé la question incongrue de la presque aussi vague ressemblance entre chaque terme $y_i \log(h_\Theta(x_i)) + (1 - y_i) \log(1 - h_\Theta(x_i))$ et **le logarithme de la « probabilité binomiale »** $\boldsymbol{p_i = h_\Theta(x_i)^{y_i} (1 - h_\Theta(x_i))^{1 - y_i}}$.
L'ironie est que **la loi binomiale** possède effectivement un **lien**… mais avec **l'entropie binaire**, pas avec la fonction « coût » !
https://en.wikipedia.org/wiki/Bernoulli_process#Law_of_large_numbers,_binomial_distribution_and_central_limit_theorem
Let us assume the canonical process with H ("heads") represented by 1 and T ("tails") represented by 0.
If the probability of flipping heads is given by p, then the total probability of seeing a string of length n with k heads is

$$\mathbf{P}([S_n = k]) = \binom{n}{k} p^k (1-p)^{n-k},$$

where $S_n = \sum_{i=1}^{n} X_i$ and $\binom{n}{k} = \frac{n!k!}{(n-k)!}$. The probability measure thus defined is known as **the Binomial distribution**.
Of particular interest is the question of the value of S_n for a sufficiently long sequences of coin flips, that is, for the limit $n \to \infty$. In this case, one may make use of Stirling's approximation to the factorial, and write

$$n! = \sqrt{2 \pi n}\, n^n e^{-n} \left(1 + \mathrm{O}\left(\frac{1}{n}\right)\right)$$

Inserting this into the expression for $P(k, n)$, one obtains **the Normal distribution**; this is the content of **the central limit theorem**, and this is the simplest example thereof.
The combination of the law of large numbers, together with the central limit theorem, leads to an interesting and perhaps surprising result: the asymptotic equipartition property. Put informally, one notes that, yes, over many coin flips, one will observe H exactly p fraction of the time, and that this corresponds exactly with the peak of the Gaussian. **The asymptotic equipartition property** essentially states that this peak is infinitely sharp, with infinite fall-off on either side. That is, given **the set of all possible infinitely long strings of H and T** occurring in the Bernoulli process, this set is **partitioned into two: those strings that occur with probability 1, and those that occur with probability 0**. This partitioning is known as the Kolmogorov 0-1 law.
➔ Je ne suis pas sûr de ce que signifie précisément l'avant-dernière phrase : doit-on comprendre que,

$$\boldsymbol{\forall\, \varepsilon > 0, \lim_{n\to\infty} \mathrm{P}(\frac{S_n}{n} (= \overline{X}) \in [p - \varepsilon, p + \varepsilon]) = 1} \text{ (et donc } \boldsymbol{\lim_{n\to\infty} \mathrm{P}(|\frac{S_n}{n} - p| > \varepsilon) = 0} \text{) ?}$$

Sauf que cette interprétation n'est qu'*asymptotique* (ne concernant que des suites *finies* de $\{H, T\}^n$) tandis que la phrase semble parler d'une **partition** « fixe » de l'ensemble des suites *infinies* de $\{H, T\}^{\mathbf{N}}$…

La partition en question serait-elle alors justement **le « pic de Dirac » $\{ (X_i)_{i \in \mathbf{N}}$, tq $\lim_{n\to\infty} \frac{S_n}{n} = p \}$, de probabilité 1, et son complémentaire (c'est-à-dire toutes les suites de « moyenne asymptotique » — limite des moyennes de Cesàro — autre que p !), de probabilité 0** ?

The size of this set is interesting, also, and can be explicitly determined: the logarithm of it is exactly the entropy of the Bernoulli process. Once again, consider the set of all strings of length n. The size of this set is 2^{n}. Of these, **only a certain subset are likely; the size of this set is** $\mathbf{2^{nH}}$ for $H \le 1$. By using Stirling's approximation, putting it into the expression for $P(k, n)$, solving for the **location** and **width** of the peak, and finally taking $n \longrightarrow \infty$ one finds that

$$\boldsymbol{H = -\,p \log_2 p - (1-p) \log_2 (1-p)}$$

This value is the Bernoulli entropy of a Bernoulli process. Here, H stands for entropy; do not confuse it with the same symbol H standing for *heads*.

➔ Ce paragraphe elliptique pour ne pas dire lapidaire mérite d'amples explicitations !

« Only a certain subset are likely » : si mon interprétation précédente est correcte, **les suites « *likely* » (disons : « presque certaines ») sont celles pour lesquelles** $\boldsymbol{\frac{S_n}{n} = \frac{k}{n}}$ **est proche de** $\boldsymbol{p}$ **et même tend vers** $\boldsymbol{p}$ **asymptotiquement**. Il s'agirait donc des **suites telles que** $\boldsymbol{S_n = k}$ **= l'entier le plus proche de** $\boldsymbol{p\,n}$. Ces suites sont au nombre de $\binom{n}{k}$.

« The size of this set is 2^{nH} »: ceci définit H comme une « **proportion logarithmique** » :

$$\boldsymbol{H = \frac{n\,H}{n} = \frac{\log_2(\text{nombre de suites « quasi-certaines »})}{\log_2(\text{nombre total de suites})}}.$$

Pour alléger, on notera dans la suite **log = log$_2$**.

Ainsi $\boldsymbol{H = \frac{1}{n} \log \binom{n}{k}}$, c'est déjà une jolie définition synthétique de l'entropie binaire. ➔ Je crains que cette prémisse soit erronée, voir à la fin pourquoi…

Mais ce n'est qu'un début car c'est là que Stirling entre en scène et que ça va devenir remarquable.

$\boldsymbol{n\,! = \sqrt{2\,\pi\,n}\; n^{\,n}\, \mathrm{e}^{\,-n}\, (\,1 + \mathrm{O}\,(\frac{1}{n}\,)\,)}$ donne $\boldsymbol{\log(n\,!) = n \log n - \frac{1}{\ln 2}\, n + \mathrm{O}(\log n)}$

donc, **à O(log $\boldsymbol{n}$) près :**

$$\log \binom{n}{k} \approx n \log n - \frac{1}{\ln 2}\, n - (n-k) \log(n-k) + \frac{1}{\ln 2}\,(n-k) - k \log k + \frac{1}{\ln 2}\, k = n \log n - n \log(n-k) - k \log k + k \log(n-k)$$

Pour faire apparaître des $\log \frac{k}{n}$ et $\log \frac{n-k}{n}$ (objectif de symétrie…), on introduit deux termes en télescopage :

$$\log \binom{n}{k} \approx k \log n - k \log k + n \log n - k \log n - n \log(n-k) + k \log(n-k)$$

$$= -\,k \log \frac{k}{n} + (n-k) \log n - (n-k) \log(n-k) = -\,k \log \frac{k}{n} - (n-k) \log \frac{n-k}{n}\ \boldsymbol{(+\ \mathrm{O}(\log n))}$$

Ainsi, $\boldsymbol{H} = \boldsymbol{\frac{1}{n} \log \binom{n}{k}} \approx \boldsymbol{-\frac{k}{n} \log \frac{k}{n}} - \boldsymbol{(1 - \frac{k}{n}) \log(1 - \frac{k}{n})}\ (+\ \boldsymbol{\mathrm{O}(\frac{\log n}{n})})$

Or $\boldsymbol{\frac{k}{n}} \longrightarrow \boldsymbol{p}$ donc $\boldsymbol{H(p) \approx -\,p \log p - (1-p) \log(1-p)}$: **formule de l'entropie binaire (à** $\boldsymbol{\mathrm{O}(\frac{\log n}{n}) \subset o(1)}$ **près) !!**

On peut aller un peu plus loin en calculant **la probabilité logarithmique asymptotique du pic de Dirac** :

$\boldsymbol{\log(\mathrm{P}(S_n = k))} = \log \binom{n}{k} + k \log p + (n-k) \log(1-p) \approx n\,[\,-\,p \log p - (1-p) \log(1-p) + \mathrm{O}(\frac{\log n}{n}) + p \log p + (1-p) \log(1-p)\,]$ $\boldsymbol{= \mathrm{O}(\log n)}$

Si l'on pouvait **affiner le D.L. de** $\boldsymbol{H}$ (via le D.A. de Stirling) pour obtenir $\boldsymbol{\log(\mathrm{P}(S_n = k)) = o(1)}$, ce serait **cohérent avec l'AEP (asymptotic equipartition property) :** $\boldsymbol{\lim_{n\to\infty} \mathrm{P}(\frac{S_n}{n} \in [p - \varepsilon, p + \varepsilon]\,) = 1}$.

https://fr.wikipedia.org/wiki/Formule_de_Stirling#D%C3%A9veloppement_asymptotique

$$n\,! = \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^{n}\left[1 + \frac{1}{12n} + \frac{1}{288n^2} - \frac{139}{51840n^3} - \frac{571}{2488320n^4} + \frac{163879}{209018880n^5} + O\left(\frac{1}{n^6}\right)\right]$$

$$\ln(n\,!) = n \ln n - n + \frac{1}{2}\ln(2\pi n) + \sum_{k=1}^{K} \frac{(-1)^{k+1} B_{k+1}}{k(k+1)n^k} + O\left(\frac{1}{n^{K+1}}\right)$$

où les B_k sont les nombres de Bernoulli.

$\boldsymbol{n}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
$\boldsymbol{B_n}$	1	$-\frac{1}{2}$	$\frac{1}{6}$	0	$-\frac{1}{30}$	0	$\frac{1}{42}$	0	$-\frac{1}{30}$	0	$\frac{5}{66}$	0	$-\frac{691}{2\,730}$	0	$\frac{7}{6}$

Ici, on a juste besoin d'aller jusqu'à $K = 1$: $\boldsymbol{\log(n\,!) = n \log n - \frac{1}{\ln 2}\, n + \frac{1}{2} \log(2\pi\, n) + \frac{1}{12\,(\ln 2)\, n} + \mathrm{O}(\frac{1}{n^2})}$

➔ Calcul des **D.L. de** $\boldsymbol{\log \binom{n}{k}}$ **puis** $\boldsymbol{\log(\mathrm{P}(S_n = k))}$ à reprendre… Prendre garde que, dans $\log(\mathrm{P}(S_n = k))$, le bloc $\log(p^k\,(1-p)^{n-k}) = k \log p + (n-k) \log(1-p)$ ne pourra peut-être plus être remplacé brutalement par l'approximation au premier ordre $n\,p \log p + n\,(1-p) \log(1-p)$ si l'on doit tenir compte que $\frac{k}{n} = p$ + « O(ε) » (voir plus bas) ?…

…

Notons que pousser le D.L. de H est également nécessaire pour passer de $\boldsymbol{H = \frac{1}{n} \log \binom{n}{k} \approx -\frac{k}{n} \log \frac{k}{n} - (1 - \frac{k}{n}) \log(1 - \frac{k}{n})}$ **à un équivalent de la taille de l'ensemble des suites « presque certaines », qui était initialement** $\boldsymbol{2^{nH}}$ par définition.

Dans « Information Theory, Inference, and Learning Algorithms » (p. 14), on trouve le D.L. obtenu à l'ordre suivant (mais pas encore à o(1) près…) :

$\log \binom{n}{k} \approx n\, H(\frac{k}{n}) - \frac{1}{2} \log[\, 2\,\pi\, n\, \frac{k}{n}\, (1 - \frac{k}{n})\,]$ où ***H* désigne désormais l'entropie binaire**.

Le second terme est en **O(log *n*)** et **négatif**, ce qui fournit donc, non un équivalent, mais une ***majoration*** de la taille **2^(*n*** $\binom{n}{k}$) par $2^{n\,H}$. Si l'on conserve la définition initiale de $2^{n\,H}$, cela signifie que **la taille de l'ensemble des suites « presque certaines » est *strictement supérieure* à** $\binom{n}{k}$, **contrairement à la prémisse faite au début, et donc** $H > \frac{1}{n} \log \binom{n}{k}$.

Autrement dit, **les suites presque certaines ne sont *pas seulement* celles telles** $S_n = k$ **= l'entier le plus proche de *p n***.

Cette « erreur » vient-elle du fait que **le pic de Dirac n'est de largeur nulle qu'*asymptotiquement*** et qu'il est nécessaire, pour tout *n*, de **compter également les *k* tels que** $\frac{k}{n} \in [p - \varepsilon, p + \varepsilon]$ **— mais alors avec quel « tolérance » ε ?!**

Je suis perdu…

Je remarque seulement maintenant la mention du « ***width*** » dans la phrase lapidaire :

« Of these, **only a certain subset are likely; the size of this set is** $2^{n\,H}$ for $H \leq 1$. By using Stirling's approximation, putting it into the expression for $P(k, n)$, solving for the **location** and **width** of the peak, and finally taking $n \to \infty$ one finds that

$H = -\,p \log p - (\,1 - p\,) \log(\,1 - p\,)$ »

ce qui rejoint ma problématique de **la « tolérance »** ε… **Quelle est donc ce « *width* »** à laquelle il est fait allusion dans l'article ?!?...

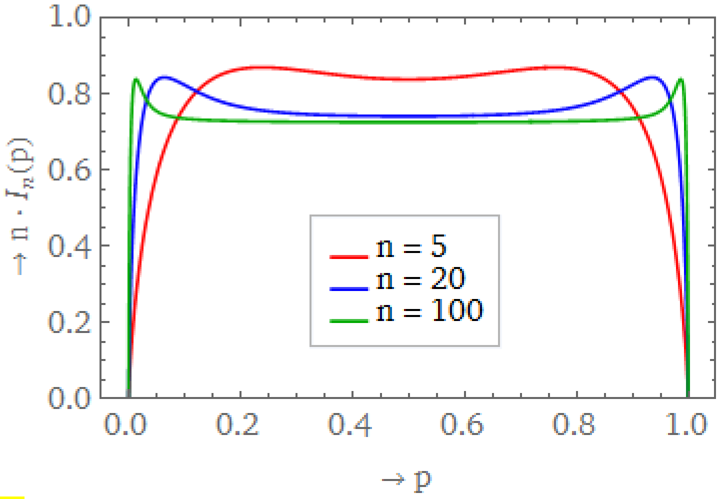
Mon interprétation initiale « ponctuelle » du « *likely* » ($S_n = k$ = l'entier le plus proche de *p n*) serait donc incorrecte et par conséquent le comptage $\binom{n}{k}$ aussi, ce qui expliquerait que l'on obtienne simplement une *minoration* de *H*…

…

➔ Une piste ?

https://mathoverflow.net/questions/200154/expected-centered-entropy-of-the-binomial-distribution

$$I_n(p) = \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} [H(p) - H(\frac{k}{n})] < \frac{1}{n} \text{ où } H(x) = -\,x \log x - (1 - x) \log(1 - x)$$



…

```
import math
# We'll do the same calculation we did above, but in Python
# Passing in 2 as the second parameter to math.log will take a base 2 log
entropy = -(2/5 * math.log(2/5, 2) + 3/5 * math.log(3/5, 2))
print(entropy)
p = [ income["high_income"].value_counts()[x] / len(income["high_income"]) for x in [0,
1] ]
print(p)
income_entropy = - ( p[0] * math.log(p[0], 2) + p[1] * math.log(p[1], 2) )
print(income_entropy)
```

```
Output
0.9709505944546686
[0.75919044255397561, 0.24080955744602439]
0.796383955202
```

We'll need a way to go from computing entropy to figuring out **which variable to split on**. We can do this using ***information gain***, which tells us which split will reduce entropy the most.

Here's the formula for information gain: