5 Information theory and Decision trees

5.1 Information theory

Information theory studies the transmission of messages. The information transmitted could be binary (as in digital communication), alphanumeric (as in a book) or genetic. Uncertainty, information and probability are all terms associated with information transfer. These are terms relevant to an event when one outcome must be selected out of many. e.g. a binary stream of data is being transmitted over a wire. Then it can be said that there is uncertainty about the next symbol/bit (0 or 1) that is expected.

'Information' does not mean 'meaning'.

Once the outcome is known, uncertainty is gone (or in general reduced), and information is gained. The meaning of the information contained in the communicated message is a totally different funda: it cannot be represented mathematically. The question relevant here is how much information was gained after a message was communicated? Note that in general when an outcome becomes known, uncertainty does not become zero: the reason is noise. For example: a binary string is being transmitted. The next bit is received and is observed to be 1. However, due to the possibility of noise, there exists a small and nonzero probability that the true value could be 0. The actual information gained must be the *difference* between the uncertainty before and after the event.

i.e. by how much did the uncertainty reduce?

$$\text{Information} = R = H_{\text{before}} - H_{\text{after}} \tag{5.1}$$

In the absence of noise, the uncertainty after the event = 0 and hence information gained = uncertainty before the communication.

Therefore 'Information' does not mean 'uncertainty', but the *reduction* in uncertainty after the message is received. This definition of information as a difference is very important. If information were directly equated to the absolute value of uncertainty before the message is received, then the more random the message, the higher its entropy and (paradoxically) the higher its information content.

Uncertainty is also called Shannon entropy H.

How should information content be measured? Given an event with many possible outcomes, the uncertainty of an outcome is small when the probability of its occurence is high. When a pair of dice are rolled, the possible totals range from 2 to 12. However, a 7 is more probable than 2 or 12 because 7 may be obtained in different ways (1+6, 2+5 etc.). Then the uncertainty of rolling a 7 is less than that of rolling a 2 or 12. The uncertainty of the outcome of an event will be largest when probabilities of all outcomes are equal. Conversely, when the probabilities of all outcomes except one is small or zero (i.e. one outcome is extremely likely), the uncertainty is least. Uncertainty is greatest when there are many outcomes to an event.

From the above observations, an estimate of uncertainty and information (H) depends on the probabilities of various outcomes (p_i). H = maximum when all probabilities are equal ($p_i = 1/n$) [assuming n outcomes]. Note that H must be a monotonic increasing function in n. $H = 0$ when all but one of the $p_i = 0$ [one outcome is expected]. H must be continuous: small changes in few of the p_i must result in small changes in H.

The information in a sentence in English (26 alphabets + punctuation) can be converted to Morse code (4 alphabets: dot, dash, letter space, word space), transmitted, received, and then

decoded back to English. Since information was not lost, this implies that the information content of the Morse message is the same as the English message. This process has multiple events: encoding, transmission, reception and decoding. These are conditional events: decoding can happen only after encoding, transmission etc. The measure of uncertainty of the overall event is the weighted sum of the expectation values of the uncertainty of the events of which it is composed.

Shannon's definition

The above criteria are satisfied by Claude Shannon's measure of information H as defined below. Given that p_i is the probability of the ith symbol, H (called Shannon entropy) is

$$H = -\sum_i p_i \log(p_i) = \mathcal{E}\left[-\log(p)\right] \tag{5.2}$$

H is a measure of the average uncertainty of an outcome. H says nothing about the meaning of that info. Note that estimation of H requires that the probability distributions are known. In practice they are unknown, and event frequencies are used instead.

The units of H are bits per symbol (if $\log_2$), digits (if $\log_{10}$) and nats (if $\log_e$).

To understand how Shannon came up with the definition of H, consider the case of equiprobable outcomes. Assume a simple $\log_2(n)$ as a formula for uncertainty. But $\log_2(n)$ = -$\log_2(p_i)$ since $p_i = 1/n$. This term (-$\log_2(p_i)$) is called the 'surprisal' u_i, in the sense that if p_i is almost 0, then you would be very surprised to see outcome i actually occur next. On the other hand, if the outcome i is guaranteed, then $p_i = 1$ and $u_i = 0$. In the general case where outcomes are not equiprobable, for a sequence of n letters, where the ith symbol (in an alphabet of m symbols) appears n_i times, then

From the definition, the surprisal would approach ∞.

$$n = \sum_{i=1}^{m} n_i \tag{5.3}$$

There are n_i cases which have surprisal u_i and hence the average surprisal (uncertainty) is

$$\begin{aligned}\mathcal{E}\left[u\right] &= \frac{\sum_{i=1}^m n_i u_i}{\sum_{i=1}^m n_i} = \frac{\sum_{i=1}^m (n_i/n) u_i}{\sum_{i=1}^m (n_i/n)} = \frac{\sum_{i=1}^m p_i u_i}{\sum_{i=1}^m p_i} \\ &= \sum_{i=1}^{m} p_i u_i = -\sum_{i=1}^{m} p_i \log_2(p_i) = H\end{aligned} \tag{5.4}$$

Note that the law of large numbers has been used: the probability p_i is assumed to be equal to the normalized event frequency n_i/n for large n. Then in the context of DNA/protein sequence analysis, can an accurate model be inferred from a small database?

From four tosses of a possibly biased coin, can it be inferred that the probability of getting a heads is 0.5?

A plot of H vs. p_i for an alphabet of two symbols would be symmetric about $p_i = 0.5$, as expected where the two symbols would be equally likely. This is convenient because of the yes/no nature of binary logic: Shannon entropy can now be thought of as the number of yes/no questions required to identify the outcome. Note that while a binary digit (0 or 1) is called a bit, here it refers to a measure of information or uncertainty. Also, very conveniently, logs are additive, probabilities by themselves are not. If two events A and B can occur, then the probability that both occur is *p(A)p(B)* but the log of this is log(*p(A)p(B)*)= log(*p(A)*) + log(*p(B)*).

$\log_2$ is commonly used. It doesn't matter if you use $\log_{10}$ or $\log_e$ or $\log_2$ for the logarithm: they are all proportional to each other, which is a nice thing about logs.

Note that if $p_i \to 0$, then from L'Hôpital's rule, (hint: use $y = 1/x$ and then apply the rule)

$$\lim_{x \to 0} x \log(x) = 0 \tag{5.5}$$

This is a little bit more appealing intuitively: we prefer additive measures of information. For example, 3 books have ∼3 times more information than one.

This implies that if one outcome is certain, $p_k = 1$ and $p_i = 0$ for $i \neq k$, then

$$H = -\sum_i p_i \log(p_i) = 0 \tag{5.6}$$

When guessing an integer from 1 to 8, $H = \log_2 8 = 3$ bits. The integer can be uniquely identified by asking 3 yes/no questions.

Q1 = AT or GC? If answer = AT, then Q2 = A or T etc.

When all the outcomes are equally probable, for n outcomes, $p_i = 1/n$, then

$$H = -\sum_i \frac{1}{n} \log_2 \left(\frac{1}{n}\right) = \frac{1}{n} \sum_i \log(n) = \log_2(n) \tag{5.7}$$

If each symbol of DNA {A, T, G, C} is equally likely, then $n = 4$ and $H = 2$ bits per DNA symbol. Hence 2 questions are needed to identify a DNA symbol. Assume that a DNA sequence which was originally random mutates such that a particular position shows conservation with A or G occuring at $p_A = 0.7$ and $p_G = 0.3$. Then $H_{before} = 2$ bits/symbol (since $p_i = 1/4$). H after the mutations is

$$H_{after} = -0.7 \log_2(0.7) - 0.3 \log_2(0.3) = 0.88$$

$$R = H_{before} - H_{after} = 1.12$$

Obviously, the more conserved the position, the smaller the value of H_{after} and the higher the information content after the substitution.

In biological systems, note that H_{after} is known, but H_{before} is unknown. Also, as the previous example indicates, natural selection implies that substitution of nucleotides reduces the information content of DNA. This may seem contradictory: a gene evolves and yet has less information content. That's the result of nature identifying useful subsequences (coding for active sites, helices etc.) and then keeping them around during evolution using selective pressure. Also remember the distinction between mutations and substitutions: mutations are random, selection results in random mutations settling into a conserved substitution pattern.

Sequence Logos

Sequence logos are visual representations of Position Specific Scoring Matrices (PSSMs) and are used to visually detect conserved DNA/protein sequence motifs. This is a powerful way of detecting genes or gene elements.

The consensus sequence of an *E. coli* TATA box is `TATAAT`. In matrix form, this could be represented as the first matrix below. The frequencies of each nucleotide at a position could also be calculated (matrix on the right).

	1	2	3	4	5	6
A	0	1	0	1	1	0
C	0	0	0	0	0	0
G	0	0	0	0	0	0
T	1	0	1	0	0	1

	1	2	3	4	5	6
A	2	94	26	59	50	1
C	9	2	14	13	20	2
G	9	1	16	15	14	2
T	80	3	44	13	16	95

A set of sequences which are believed to be related are aligned at some position (column) based on previous insight and the frequencies $f(b, l)$ of base b at position l are computed. A set of aligned sequences are displayed as a stack of characters at a position, with the height of each character proportional to its frequency, and the letters sorted so that the most common one is on top. The height of the entire stack is then adjusted to represent the information content of the sequences at that position. The average information (or uncertainty at a position l is (in bits)

$$H(l) = -\sum_{b=a}^{t} f(b, l) \log_2 f(b, l)$$

The decrease in uncertainty at the position is the total information and is

$$R_{seq}(l) = 2 - (H(l) + e(n))$$

where $e(n)$ is a correction factor that is required when n is small (only a few sample sequences

For example, alignment at `ATG`, the translation start codon.

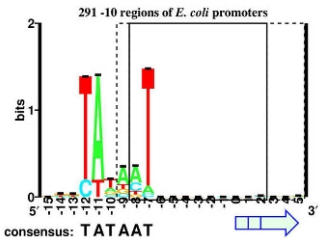


Fig. 5.1: The `TATA` box in *E. coli*.

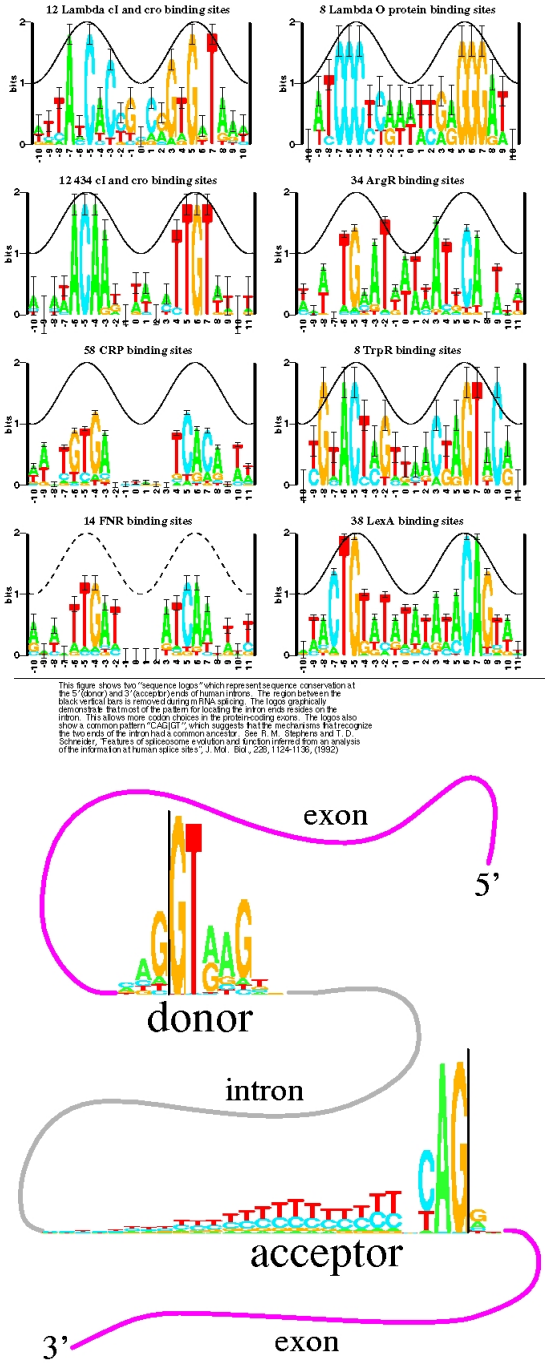


Fig. 5.2 A gallery of transcription factor binding sites is shown on the left. Differences between human donor and acceptor splice junctions are evident on the right.

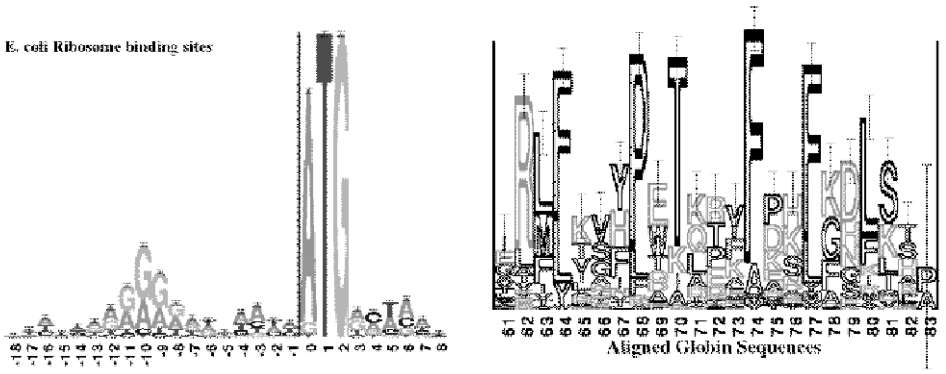
are available). $R_{seq}(l)$ versus l then forms a curve with height = $R_{seq}(l)$. The height of each base at that position is $f(b,l)R_{seq}(l)$. In the sequence logo for the `TATA` box, notice that positions 1, 2 and 6 are highly conserved.

In Fig. 5.2a, the periodicity that results when proteins bind helically twisting DNA is obvious. In figure 5.2b, both human donor or acceptor splice junctions have the same consensus `CAG|GT`. However, from the extent of information per position, these are clearly qualitatively different.

The examples below show conservation of *E. coli* ribosome binding sites (left) and the aligned globin binding sites.

For protein sequences, assuming that uncertainty is highest if the 20 amino acids are equiprobable,

$$R_{seq}(l) = \log_2 20 - (H(l) + e(n))$$



The *E. coli* ribosome binding site example is particularly interesting: *E. coli* has 4.7×10^6 bases and approximately 2600 genes. The information required by a ribosome to touch down onto one of these 2600 binding sites is

$$R_{frequency} = \log_2 4.7 \times 10^6 - \log_2 2600 = 10.8 \text{ bits/site}$$

A postman needs sufficient information to correctly deliver a letter.

The information provided by the binding site (the area under the sequence logo) is $R_{sequence} = 11.0 \pm 0.4$ bits/site!

Similarly, human splice acceptor sites contain ∼9.4 bits of info. The average distance between acceptor sites = average size of introns + exons ∼812 bases, which implies that

$$R_{frequency} = \log_2 812 = 9.7 \text{ bits}$$

Bacteriophage T7 promoters have ∼35 bits of info surrounding the transcriptional start. T7 polymerase needs $\sim 18 \pm 2$ bits. A possible hypothesis is that two molecules may wish to bind to the same site. However not every bit has to contribute the same binding energy. Having excess information does not hurt.

$k_B T \ln 2 \leq -q/R$

Randomness

Consider the two sequences 010101010101010101... and 011011001101111000.... The first sequence is obviously (01) repeated many times. However, the second sequence has no obvious pattern. If these two patterns represent tosses of a coin (with heads = 1),then both sequences are equally likely from 'random' tosses of a coin. Yet sequence 1 is more orderly. The key observation here is that sequence 1 may be represented by a short algorithm which says '*repeat* 01 *n times*'. If both strings were to be extended to infinity, then the infinitely long first sequence would be represented by an algorithm of finite length. If it had to be transmitted elsewhere, it would be obviously easier to transmit the algorithm than the actual string itself.

Also known as Kolmogorov complexity.

This, roughly, is how computer files are compressed.

Thus, the information present in the sequence is compressed and transmitted with the number of bits in the algorithm being a small fraction of the number of bits in the sequence. The second sequence, however, cannot be easily compressed. *A sequence is random if the smallest algorithm capable of specifying it to a computer has ∼ the same number of bits of information as the sequence itself.* This concept can be extended to a choice of model, when several are available to describe a phenomenon: we choose the model with the *minimum description length* (MDL), which is the most compressed description of the model. A related philosophical concept is *Occam's razor*: Given different theories of apparently equal merit, the simplest is to be preferred. Look up Occam's razor [sometimes spelt Ockham].

KISS: Keep It Simple Stupid.

Entropy

The equivalent concept to Shannon entropy is that of Maxwell-Boltzmann-Gibbs entropy from statistical thermodynamics. Let N atoms be arranged in M_i infinitesimal cells in W ways. Then entropy S is defined as

$$S = \ln(W) \tag{5.8}$$

Information in statistical thermo is also measured using a log form: $\log_e$ instead of $\log_2$.

But

$$W = \frac{N!}{\prod_i M_i!} \quad \text{and} \quad S = \ln(N!) - \sum_i \ln(M_i!) \tag{5.9}$$

Using Stirling's approximation for a factorial, and assuming that for a large number of atoms that $p_i = M_i/N$, then

$$N! \approx \sqrt{2\pi N}\left(\frac{N}{e}\right)^n$$

$$S \approx -\sum_i p_i \ln(p_i) = -k_B \sum_i p_i \ln(p_i) \tag{5.10}$$

k_B is the Boltzmann constant ($= 1.38 \times 10^{-23}$ J/°K).

Shannon entropy and S are not the same thing in general, despite the similarity in formulae. They are similar only if their probability spaces are isomorphic (each probability has a one to one mapping between the info theory and thermo formulations). This would be like trying to compare two different sequences generated by repeatedly tossing a coin and rolling a die. These two sequences *would* be isomorphic however, if the numbers on the die are noted down as odd or even (hence only 2 outcomes instead of 6). Also, in thermo there is a conservation of energy requirement which is not present in information theory.

Since $R = H_{before} - H_{after}$ and $\Delta S = S_{before} - S_{after}$, assuming that the sequences are isomorphic, then

$$\Delta S = -k_B \ln(2) R \tag{5.11}$$

But from the second law of thermo, $\Delta S \geq q/T$ or $k_B T \ln(2) \leq -q/R$ which implies that for every bit of information gained ($R = 1$) the heat that must be dissipated into the surroundings must be more than $\varepsilon_{min} = k_B T \ln(2)$ Joules/bit.

Note that the ln(2) comes in because we choose a bit as a unit of choice. If digits (decimal) had been chosen instead, the minimal energy dissipation ε_{min} would be $k_B T \ln(10)$ Joules/*digit*.

This is a different way of looking at the second law: think of how Maxwell's demon was supposed to allow fast molecules to pass into a separate part of a chamber initially at equilibrium. Then the temperatures of the two sections of the chamber would have changed resulting in a violation of the second law. The resolution of this paradox is that for every decision that the demon makes (i.e. which section to let a given molecule enter), it must dissipate at least $k_B T \ln(2)$. Assuming that there are three steps for each molecule (1. see the molecule, 2. decide whether it a fast or slow molecule and 3. choose the proper chamber) then each molecule requires $3\varepsilon_{min}$. The paradox can still pose a problem if most of the molecules have energies which are more than $3\varepsilon_{min}$ above the background (average) energy. However, most of the molecules would have less than $3\varepsilon_{min}$ energy above background, with the Boltzmann distribution stating that the probability p of a molecule with energy ΔE above background is

Thus the demon has to make $\log_2(8) = 3$ bits of choice and must *itself* spend at least $3\varepsilon_{min}$ joules of energy.

$$p = e^{-\Delta E/k_B T} \tag{5.12}$$

Then letting $\Delta E = 3\varepsilon_{min} = 3k_B T \ln(2)$, $p = 1/8$ which implies that the demon will reject 7 molecules for every one allowed to pass.

Cross entropy:

Given two probability distributions, p and q, the cross entropy is

$$H(p, q) = \mathcal{E}_p\left[-\log q(x)\right] = -\sum p(x) \log q(x) \tag{5.13}$$

Jensen's inequality: when f is a convex function and $\sum p_i = 1$.

$$\sum p_i f(x_i) \geq f\left(\sum p_i x_i\right)$$

Intuitively, $H(p,q) \geq H(p)$. Using Jensen's inequality,

$$H(p,q) - H(p) = \sum p(x) \left[-\log \frac{q(x)}{p(x)} \right] \geq -\log \sum p(x) \frac{q(x)}{p(x)} = 0 \tag{5.14}$$

Kullback-Liebler divergence:
The Kullback-Liebler divergence $D(p||q)$ is a measure comparing two distributions p and q.

$$D(p||q) = H(p,q) - H(p) = \sum p(x) \log \frac{p(x)}{q(x)} \tag{5.15}$$

If p is fixed, $D(p||q)$ and $H(p,q)$ vary the same way. If p is some empirical distribution, then minimizing $D(p||q)$ or $H(p,q)$ is equivalent to performing a maximum likelihood estimation.

$D(p||q) > 0$
$D(p||q) \neq D(q||p)$
$D(p||q) = 0$ iff $p = q$

Proof that $D(p||q) \geq 0$: Note that $e^x \geq 1 + x$ implies $e^{x-1} \geq x$ and on taking the log, $\ln(x) \leq x - 1$. Then $-\sum p_i \ln \frac{p_i}{q_i} \geq -\sum p_i \left(\frac{q_i}{p_i} - 1 \right)$ (The inequality changes direction because of the minus sign. Also a shift from log to ln involves a constant conversion factor which may be ignored). The RHS is $\sum(p_i - q_i) = 0$. There is also a proof using Jensen's inequality. Check wikipedia for both Jensen's and Gibb's inequalities.

Cross and conditional entropy:
Given two random variables X and Y, and their joint pdf $p(x,y)$, then

$$H(x) = -\sum_{x,y} p(x,y) \log p(x) \tag{5.16}$$

$$H(y) = -\sum_{x,y} p(x,y) \log p(y) \tag{5.17}$$

$$H(x|y) = -\sum_{x,y} p(x,y) \log p(x|y) = -\sum_{y} p(y) \sum_{x} p(x|y) \log p(x|y) \tag{5.18}$$

$$H(x,y) = -\sum_{x,y} p(x,y) \log p(x,y) \tag{5.19}$$

Mutual information:
The simultaneous existence of information at two locations can be often important (the simultaneous existence of various informative sites on a stretch of DNA could imply that it is a gene). This is a measure of correlation between two random variables. Mutual information (entropy) is the sum of individual entropies minus the entropy of co-occurences.

$$\begin{aligned} I(x;y) &= H(x) - H(x|y) \\ &= H(y) - H(y|x) \\ &= H(x) + H(y) - H(x,y) \end{aligned} \tag{5.20}$$

$I(x;y) \geq 0$
$I(x;y) = I(y;x)$
$I(x;y) = 0$ iff X and Y are independent.

Obviously, $I(x;y)$ is a measure of the reduction of uncertainty in X given some information about Y.

$I(x;y)$ can serve as a metric. $D(x||y)$ cannot. Why?

Example 5.1. Assume that I have chosen a number ζ out of the set $N = 1, 2, 3, 4$. What is the minimum number of yes/no questions required to identify the number?

Soln:

Let $p(n)$ = probability that ζ equals n. Then

$$H(N) = \sum_{i=1}^{4} -p(n) \log_2 p(n) = \sum_{i=1}^{4} -0.25 \log_2 0.25 = 2 \,\text{bits}$$

Ask the question "Is $\zeta = 3$ or 4?". If the answer is yes, then conditional information is now

$$H(N|\text{yes}) = \sum_{i=1}^{4} -p(n|\text{yes}) \log_2 p(n|\text{yes}) = 0 + 0 + 0.5 + 0.5 = 1 \,\text{bit}$$

Similarly, $H(N|\text{no}) = 1\,\text{bit}$. Then the expectation of the conditional probability $= H(N|Q)$ where $Q = \{\text{yes,no}\}$ is

$$\begin{aligned} H(N|Q) &= \sum_{q\in\{\text{yes,no}\}} \sum_{i=1}^{4} P(q)P(n|q)\log_2 P(n|q) \\ &= \sum_{n=1}^{4} -P(\text{yes})P(n|\text{yes})\log_2 P(n|\text{yes}) + + \sum_{n=1}^{4} -P(\text{no})P(n|\text{no})\log_2 P(n|\text{no}) \\ &= 0.5\times 1 + 0.5\times 1 = 1\,\text{bit} \end{aligned}$$

The reduction in uncertainty equals 1 bit. Contrast the question "is $\zeta = 3$ or 4" with "is $\zeta = 4$". The expected information gain with "is $\zeta = 4$" is

$$\begin{aligned} I(N;Q) &= H(N) - H(N|Q) \\ &= 2 - \sum_{n=1}^{4} -P(\text{yes})P(n|\text{yes})\log_2 P(n|\text{yes}) + \sum_{n=1}^{4} -P(\text{no})P(n|\text{no})\log_2 P(n|\text{no}) \\ &= 2 - (0 + 1.19) = 0.81\,\text{bit} \end{aligned}$$

The information gain with "is $\zeta = 3$ or 4" was 1 bit, and hence is the better question.

This is a useful strategy in inducing decision tree rules.

5.2 Decision Trees

The more popular Decision Tree algorithms include ID3, C4.5 and CART. A simplified version of ID3 is given below.

Consider the medical data given below:

Patient	Heart rate	B.P.	Class
1	irregular	normal	ill
2	regular	normal	healthy
3	irregular	abnormal	ill
4	irregular	normal	ill
5	regular	normal	healthy
6	regular	abnormal	ill
7	regular	normal	healthy
8	regular	normal	healthy

The uncertainty before any questions are asked is

$$\begin{aligned} H(\text{class}) &= -P(\text{healthy})\log_2 P(\text{healthy}) - P(\text{ill})\log_2 P(\text{ill}) \\ &= 0.5 + 0.5 = 1\text{ bit} \end{aligned}$$

There are different ways to partition, based on the question asked first:

$$\begin{aligned} H(\text{class}|\text{heart rate}) &= -P(\text{irreg})P(\text{healthy}|\text{irreg})\log_2 P(\text{healthy}|\text{irreg}) \\ &\quad -P(\text{reg})P(\text{healthy}|\text{reg})\log_2 P(\text{healthy}|\text{reg}) + \\ &\quad -P(\text{irreg})P(\text{ill}|\text{irreg})\log_2 P(\text{ill}|\text{irreg}) - P(\text{reg})P(\text{ill}|\text{reg})\log_2 P(\text{ill}|\text{reg}) \\ &= -0.375\times 0 \; - 0.625\times 0.8\log_2 0.8 - 0.375\times 0 \; - 0.625\times 0.2\log_2 0.2 \\ &= 0.45\text{ bits} \\ H(\text{class}|\text{B.P.}) &= 0.25\times 0 \; + 0.75\times 0.66\log_2 0.66 + 0.25\times 0 \; + 0.75\times 0.33\log_2 0.33 \\ &= 0.69\text{ bits} \end{aligned}$$

$$\begin{aligned} I(\text{class};\text{heart rate}) &= 1 - 0.45 &&= 0.55\text{ bit} \\ I(\text{class};\text{B.P.}) &= 1 - 0.69 &&= 0.31\text{ bit} \end{aligned}$$

Obviously, one should partition first using heart rate.

Notice that irregular heart beat patients are all ill (convenient). Of 5 people with reg. heart beat, one is ill (inconvenient). This node has to be further subdivided into two groups using B.P.

Real problems include noise: overfitting can happen.

By walking along the branches, the folowing rules may be induced:

1a If heart rate = irreg, then ill.
2a If heart rate = reg and BP = abnormal, patient = ill.
3a If heart rate = reg and BP = normal, then healthy.

However consider an alternative set of rules, which works just as well:

1b If heart beat = irreg, then ill.
2b If BP = abnormal, then ill
3b If heart rate = normal and BP = normal, then healthy

Rules 1a and 2a overlap.

Clearly, 2a is more specific than 2b. The second set of rules permits overlapping subclasses, and in general, this is desirable. More complicated systems result in very complicated rules: overlapping rules are better in such systems. When very complicated trees are induced, it may be necessary to go through a pruning stage where least essential branches are removed while ensuring that the performance of the decision tree does not deteriorate substantially.

Gini impurity

Nodes are expected to be pure: i.e. represent points belonging to the same class.

Information gain works as a measure of node impurity. An alternate to information gain is the Gini criterion (used in CART)

$$I = \sum_{i \neq j} p(\omega_i)p(\omega_j) = \frac{1}{2}\left[1 - \sum_j p^2(\omega_j)\right] \tag{5.21}$$

This is equivalent to the expected error rate if the classification was picked according to the class distribution. We can also use the risk concept from Bayesian classification: we can involve weights λ_{ij} which represent the risk of misclassifying a point from class i to class j.

$$I = \sum_{j \neq i} \lambda_{ij} p(\omega_i)p(\omega_j) \tag{5.22}$$

5.3 Codes and Channel capacity

Examples of codes for transmitting information include any language (English!), Morse code, binary code, the genetic code etc. The 4 DNA nucleotides specify the 20 amino acids in proteins using the genetic code.

In general, a code is the mapping of letters of alphabet A onto the letters of alphabet B. The code need not be isomorphic(i.e. bijective i.e. a unique one-to-one mapping both ways. DNA codon triplets map uniquely to amino acids but not vice versa). Hence information is conserved from DNA to mRNA (the same Shannon entropies) but some information is lost in going from mRNA to protein.

Note the redundancy in the genetic code: 64 codons map to 20 amino acids. Also note that all words using this code have the same length (codon = 3 letters long). This is an example of a *block code* where all the words (sequences) are of the same length and use a certain alphabet. Each code word is distinguishable from the other and hence the code is *distinct*. Every code word is identifiable instantaneously when contained in a sequence of other words (a sentence). This code is *uniquely decodable*. For instance if UU were mapped to Ala and UUU to Leu, then on receiving UUUAA, is it actually UU+UAA or UUU+AA? This code would not be instantaneously decodable. The code would be *optimal* if its words had a minimum average

Another way to state this is that the code has been maximally compressed.

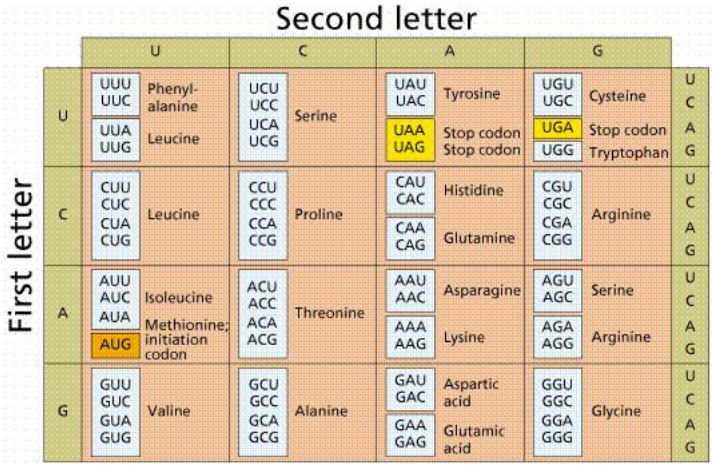


Fig. 5.3 The 'Universal' genetic code. Alternate codes exist but are rarely used. The layout of amino acids in this code is not random. Consider what might happen when substitutions occur in the third or second position.

length, while remaining uniquely decodable. The genetic code has been shown to have these properties.

Shannon's channel capacity theorem

A code exists such that an original message may be recovered by the use of redundancy, from the received message affected by noise, with as small a probability of error as desired. The computer science analogy is binary data flowing through a wire subject to some noise. How is the bandwidth best utilized? How is noise handled? How can financial transactions be carried out accurately? The answer is to design codes with redundancy.

This theorem has resulted in modern digital transmission achievements (clear telephonic conversations, digital music/CDs etc.)

In biology, noise = mutation, changing (possibly) the amino acid sequence.

Noise can affect the transmission of information. To overcome noise, a simple approach would be to resend the message several times. But this wastes bandwidth given a channel capacity. It would be particularly wasteful if most of the message, say 99.99% makes it over fine, and you have to resend the message because of the 0.01%.

Instead the info may be transmitted by a code with redundancy built in, so that certain mutations can be withstood. This would be an error correcting code. For example {U, A, G, C} may be mapped to {00, 01, 10, 11} which is a binary code using words of length 2 bits. However any line noise flipping a 0 into a 1 would catastrophically change the transmitted message. If instead of two-bit words, let the 4 nucleotides be mapped to the 5-bit 'sense' words {11000, 00110, 10011, 01101}. First observe that there are at most $2^5 = 32$ 5-bit long words.

Nucleotide	U	A	G	C
Sense word	11000	00110	10011	01101
Code words at	11001	00111	10010	01100
Hamming distance 1	11010	00100	10001	01111
	11100	00010	10111	01001
	10000	01110	11011	00101
	01000	10110	00011	11101
Code words at	11110	00000	01011	10101
Hamming distance 2	01010	10100	11111	00001

The number of positions in which synonymous source words differ as a result of noise during

transmission (mutation) is called the Hamming distance. Thus, if a 11001 is received, then it is not one of the 4 source words, but differs from 11000 by one position only and is hence most likely the nucleotide U (because it differs from A = 00110 by 5 positions, G = 10011 by 2 positions and C = 01101 by 2 positions). Hence any words transmitted with one error may be received correctly. However the code words which are at a Hamming distance = 2 may be decoded incorrectly: 00000 is distance 2 away from both U and A. It was assumed above that {U, A, G, C} was first converted to the four 5-bit sense words and then transmitted and that noise altered some of the bits. However in general, the other 28 code words may also be transmitted complicating things and reducing the accuracy of error detection. On the other hand if the sender sends only the 4 5-bit words, then from the received signal, an estimate of noise on the line may be determined.

Space craft near Neptune transmit images back to Earth using a code word of 32 bits (of which 26 are redundant bits).

In the genetic code, assuming that one codon is mapped to each of the 20 amino acids, then the remaining 44 codons must be assigned to reduce susceptibility to mutation. In the modern genetic code, the best protected codons are CUA, CUG (Leu) and CGA, CGG (Arg). Current usage (nature has had 4 billion years to change its codon usage patterns) favors CUG for Leu with CUA being rarely used in *E. coli*. CGG and CGA are rarely used in *E. coli* and yeast. The least protected codon is Trp which can be easily converted to a termination codon.

Channel capacity:

The channel capacity C (bits/second) is defined as

$$C = W \log\left(\frac{P}{N} + 1\right)$$

where W = bandwidth (cycles per second), P is the signal power at the receiver (J/s) and N is the thermal noise at the receiver (J/s). Then to increase capacity, increase P/N (eg. shout into a phone if the line is noisy to get heard at the other end) or increase W (a stereo FM station which broadcasts on additional and adjacent frequencies to achieve stereo sound).

Shannon's theorem

If information is sent at a rate $R > C$, then the transmission fails with at most C getting through with noise destroying the rest. If information is sent at a rate $R \leq C$, then it may be done *with as few errors as desired*. This is done by encoding the signal being sent out, as described above, thus rendering the information resistant to machine noise.

Biological machines

This has been used to explain how ribosomes can rapidly determine where to bind mRNA (i.e. how ribosomal binding sites are rapidly identified). If ribosomes started translation elsewhere, this would be energetically wasteful. Hence rapid choices may be made by a ribosome with as few errors as possible, as long as $R \leq C$ for its system. The same logic holds for a PCR reaction or for antibody-antigen binding. The theorem indicates that molecular machines may be as precise as possible in their operation as long as they operate at or below their machine capacity.