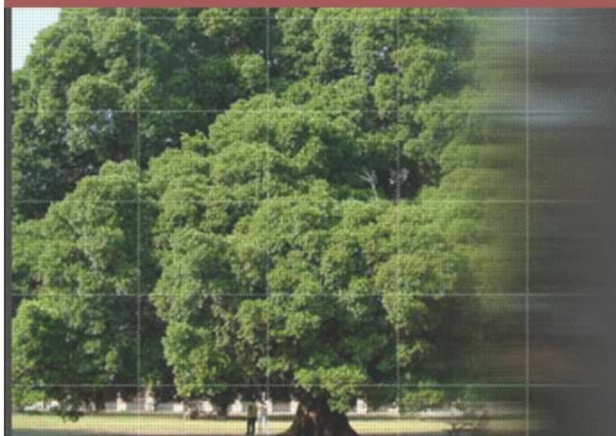


資訊理論之觀念

第2章

資訊（Information）

- 假設我們有一個含有q個符號$s_1, s_2, \cdots\cdots, s_q$之符號源，其出現機率分別為$p(s_1)=p_1, p(s_2)=p_2, \cdots\cdots, p(s_q)=p_q$。當我們收到其中一個符號時，我們得到多少資訊？
- 如果$p_1= 1$，那麼沒什麼好驚訝、也沒什麼資訊，因為我們知道收到的訊息一定是什麼。
- 因此，資訊就某方面來講與出現機率存在著倒數的關係。

資訊（Information）

- 對於一個出現機率為p的事件，我們希望能找出一個用來估算資訊量（驚訝度、不確定性）的函數，$I(p)$。

- 對於$I(p)$ 我們可以做以下三項既自然而又合理的假設：

1. $I(p) \geq 0$（非負實數）。
2. 對於兩個獨立事件，$I(p_1p_2)=I(p_1)+I(p_2)$
3. $I(p)$ 是p的連續函數。

資訊（Information）

- 若p_1與p_2都等於p（但不見得是同一事件），則由第二條假設可得到

$$I(p^2)=I(p)+I(p)=2I(p)$$

- 若$p_1=p$，且$p_2=p^2$，則我們可得

$$I(p^3)=I(p)+I(p^2)=3I(p)，$$

推廣之，得到$I(p^n)=n\times I(p)$。

資訊（Information）

- 透過這個線索，將原來之正整數指數再推廣至正分數指數，設$p^n=y \Rightarrow p=y^{1/n}$，於是

$$I(y)=n\times I(y^{1/n})\text{，}$$

進一步的整理可得$I(y^{m/n})=I(y)$。

因此，函數$I(p)$即使使用分數值，仍然遵循log函數的公式。

- 使用以2為底的log函數計算出來的資訊量是以位元（*bit*）為單位。使用*e*為底所得到之資訊量的單位，我們稱之為*nat*。使用10為底，計算出來的資訊量是以*Hartley*為單位

熵（Entropy）

- 當收到符號s_i時，我們得到$I(s_i)$單位的資訊，由於得到$I(s_i)$資訊的機率是p_i，因此對於每個符號s_i，平均得到的資訊量是

$$p_i I(s_i)=p_i \log_2(1/p_i)。$$

- 對於整個符號源平均可以得到的資訊量為$\sum_{i=1}^{q} p_i \log_2 \frac{1}{p_i}$或者把基底也考慮進去，$H_r(S)=\sum_{i=1}^{q} p_i \log_r \frac{1}{p_i}$，我們稱$H_r(S)$為訊號系統(或符號源)$S=\{s_i \mid p(s_i)=p_i\}$的熵。

• $H_r(S)=H_2(S)\log_r 2$是DMS的熵函數，所考慮到的只是每個符號的出現機率。

熵（Entropy）

例2.1：假設符號源$S=\{s_1, s_2, s_3, s_4\}$ 且$p_1=1/2$, $p_2=1/4$, $p_3=1/8=p_4$。以$r=2$為基底我們可得到熵為

$$H_2(S)=1/2\times\log_2 2+1/4\times\log_2 4+1/8\times\log_2 8 +1/8\times\log_2 8$$
$$=1/2\times1+1/4\times2+1/8\times3+1/8\times3$$
$$=1.75 \text{ 資訊位元。}$$

例2.2：

- 丟擲一個兩面出現機率相同的銅板之熵值：

$$I(s_1)=I(s_2)=\log_2(1/2)^{-1}=\log_2 2=1，$$
$$H_2(S)=I(s_1)+I(s_2)=1，$$

$H_2(S)$ 的值即等於$I(s_i)$ 的值。

熵（Entropy）

- 如果銅板的兩面出現機率不相等，熵會有什麼改變？
  依定義：
  $$H_2(S) = p \times \log_2(\frac{1}{p}) + (1-p) \times \log_2(\frac{1}{1-p}) = H_2(P)$$
  其中p為正面出現之機率(當只有兩種符號時，$H_2(S)$ 與 $H_2(P)$ 在許多資訊理論的書中是互用的)。
- 圖2.1所示為$H_2(P)$ 的值以p為參數所形成的曲線。最大值發生於p=1−p=1/2時。

- 同樣地，丟擲六面出現機率均等的骰子：
  $I(s_i)=\log_2(1/6)^{-1}=\log_2 6$，
  $H_2(S)=6[I(s_i)]=\log_2 6=2.5850$資訊位元。

熵（Entropy）

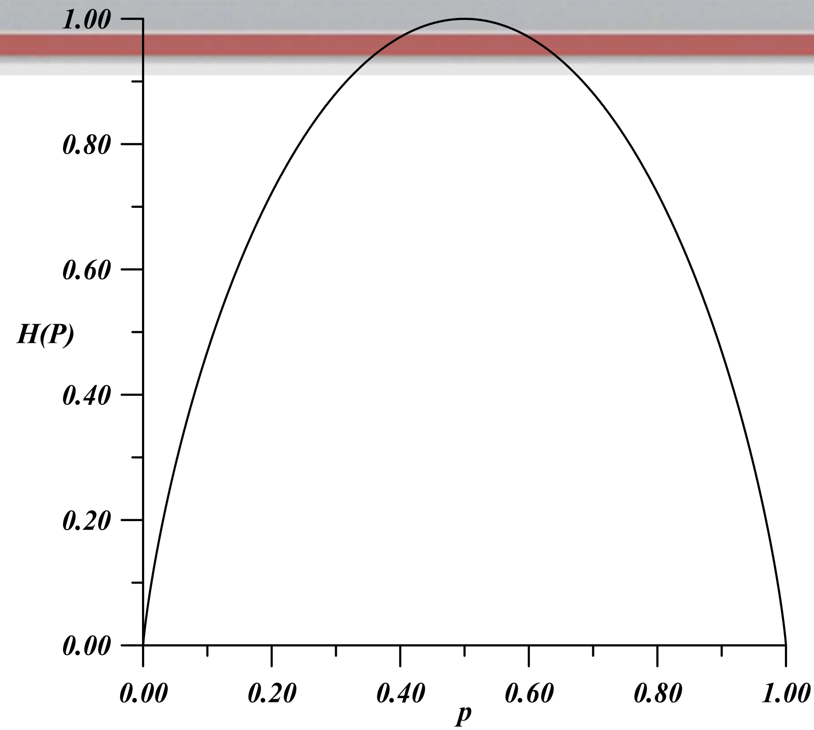


圖2.1　兩個機率之熵函數。

- 從例2.2得知，當所有符號的出現機率一樣時，熵的值即等於任何一事件(出現任何一個符號)的資訊量，這也是熵的最大可能值。

熵（Entropy）

- 證明：

從圖2.2可以看出 $\ln(x)$（$=\log_e x$）函數的性質：

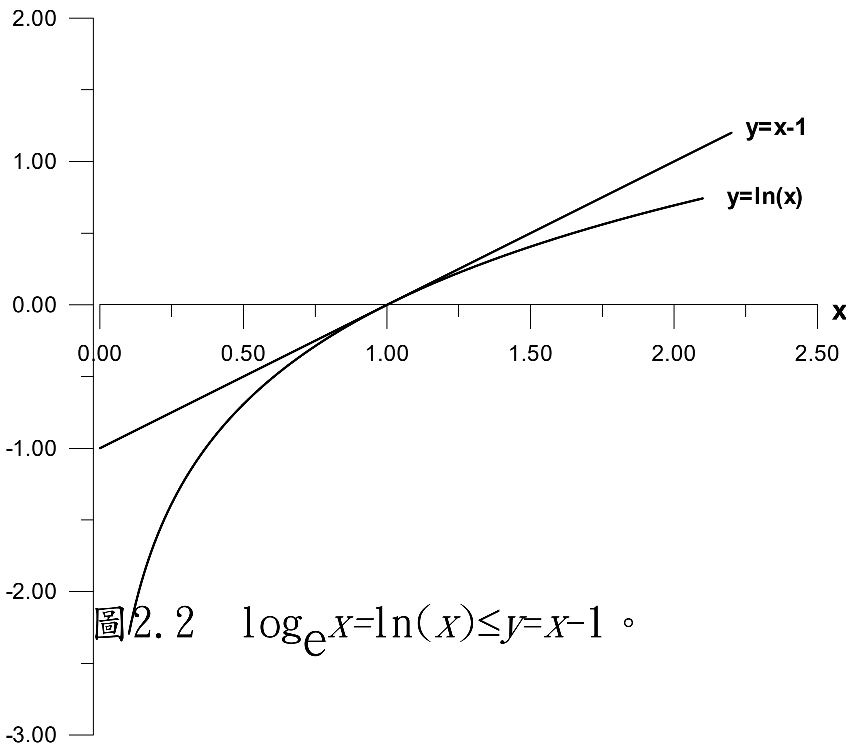


圖2.2　$\log_e x=\ln(x)\le y=x-1$。

在點(1, 0)做切線，斜率是 $\left.\dfrac{d[\ln(x)]}{dx}\right|_{x=1}=1$，因此這條切線函數是

$$y=x-1,$$

當 $x\ge 0$，$\ln(x)\le x-1$ 成立。

熵（Entropy）

- *Gibbs*的基本不等式：$\sum_{i=1}^{q} x_i \log_2(\frac{y_i}{x_i}) \le 0$

  令x_i為第一個機率分布而y_i為第二個機率分布，

$$
\begin{aligned}
& \sum_i x_i \log_2({y_i}/{x_i}) \\
= \quad & \frac{1}{\ln 2}\sum_i x_i \ln({y_i}/{x_i}) \\
\le \quad & \frac{1}{\ln 2}\sum_i x_i({y_i}/{x_i} - 1) \qquad [\because \ln(x) \le x-1,\ \ x \ge 0] \\
\le \quad & \frac{1}{\ln 2}\sum_i (y_i - x_i) \\
\le \quad & \frac{1}{\ln 2}(\sum_i y_i - \sum_i x_i) = 0
\end{aligned}
$$

熵（Entropy）

熵之定義：$H_2(S)=\sum_i p_i \log_2 \frac{1}{p_i}$，其中 $\sum_i p_i = 1$。

由 $Gibbs$ 不等式得知 $H_2(S) \le \log_2 q$，等號發生於所有 $x_i = y_i$ 時，即 $p_i = x_i = y_i = 1/q$。

$$
\begin{aligned}
& H_2(S) - \log_2 q \\
= & \sum_{i=1}^{q} p_i \log_2 \left(\frac{1}{p_i}\right) - \log_2 q \sum_{i=1}^{q} p_i \\
= & \sum_{i=1} p_i \log_2 \left(\frac{1}{q p_i}\right) \\
= & \sum_{i=1} x_i \log_2 \left(\frac{y_i}{x_i}\right)
\end{aligned}
$$

令 $x_i = p_i$ 而 $y_i = 1/q$，則 $H_2(S) - \log_2 q \le 0$

編碼（Coding）

- 唯一解碼（*unique decodable*）—指的是所收到的訊息只有唯一的一種解釋。
- 即時碼（*instantaneous code*）—一旦一個完整的符號碼被收到，接收者馬上就能知道，不需要多看後面幾個位元，便能確定已收到某個符號。在這個編碼中沒有一個符號碼是另一個符號碼的字首（*prefix*）。

編碼 (Coding)

- 例：

s_1=0
s_2=10
s_3=110
s_4=111
instantaneous
101100111- $s_2s_3s_1s_4$

s_1=0
s_2=01
s_3=011
s_4=111
non-instantaneous
011111111- $s_3s_4s_4$

- 一個編碼是即時碼若且唯若沒有一個符號碼是另一個符號碼的字首，即時碼一定具有唯一解碼性，反之不必然。

熵與編碼

- *Kraft*不等式（*Kraft inequality*）
  - 定理：假設含有q個符號之符號源其編碼之長度分別為l_1, l_2, …，l_q且假設$l_1 \le l_2 \le \cdots \le l_q$。這種長度的即時碼存在之充份必要條件為 $\sum_{i=1}^{q} \frac{1}{r^{l_i}} \le 1$，$r$為編碼時使用的碼的基底(*radix*)。

熵與編碼

- 定理：任何即時編碼的平均編碼長度都大於等於其熵值。

- 證明：給定任何的即時碼，它都有確定的編碼長度 l_i 及基底 r。由 $Kraft$ 不等式我們知道 $K=\sum_{i=1}^{q}\frac{1}{r^{l_i}}\le 1$

  定義 Q_i（機率分布）：$Q_i=\frac{r^{-l_i}}{K}$ 其中當然滿足 $\sum_{i=1}^{q}Q_i=1$ 很明顯地，Qi 是一個機率分布。因此我們可以使用 $Gibbs$ 不等式。

熵與編碼

$$\sum_{i=1}^{q} p_i \log_2\left(\frac{Q_i}{p_i}\right) \le 0$$

$$\Rightarrow \sum_{i=1}^{q} p_i \log_2\left(\frac{1}{p_i}\right) \le \sum_{i=1}^{q} p_i \log_2\left(\frac{1}{Q_i}\right)$$

$$= \sum_{i=1}^{q} p_i (\log_2 K - \log_2 r^{-l_i})$$

$$= \log_2 K + \sum_{i} p_i l_i \log_2 r$$

熵與編碼

由$Kraft$不等式知$K\le 1$，因此$\log_2 K\le 0$。去掉這一項只會強化這個不等式的成立。於是我們得到：

$$
\begin{aligned}
H_2(S) &= \sum_{i=1}^{q} p_i \log_2\left(1/p_i \right) \\
&\le \log_2 K + \sum_i p_i l_i \log_2 r \\
&\le \sum_i p_i l_i \log_2 r \\
&= L_{avg} \log_2 r
\end{aligned}
$$

或$H_r(S)\le L$，其中L是平均編碼長度。

Shannon-Fano編碼

- *Shannon-Fano*編碼的效率雖然不如*Huffman*編碼，但我們可以直接從機率值p_i知道編碼長度l_i，而*Huffman*編碼所使用的每一個碼的長度則決定於所有的p_i值。

Shannon-Fano編碼

- 對於一符號源內之符號s_1, s_2, …, s_q以及它們的出現機率p_1, p_2, …, p_q，*Shannon-Fano*編碼可以針對每一個p_i值獨立地決定出l_i的值，即取l_i為滿足下列式子之整數：

$$\log_r(1/p_i) \le l_i < \log_r(1/p_i)+1$$
$$\Rightarrow 1/p_i \le r^{l_i} < r/p_i$$
$$\Rightarrow \sum_{i=1}^{q} p_i \ge \sum_{i=1}^{q} \frac{1}{r^{l_i}} > \sum_{i=1}^{q} p_i/r$$
$$\Rightarrow 1 \ge \sum_{i=1}^{q} \frac{1}{r^{l_i}} > \frac{1}{r} \quad \text{即}Kraft\text{不等式}$$

- 因此，存在一個即時碼其編碼長度為l_1, l_2, …, l_q，我們稱之為*Shannon-Fano*長度。

Shannon-Fano編碼

- 要知道*Shannon-Fano*編碼的熵值有多大就必須由l_i的取法開始：

$$\log_r\left(\frac{1}{p_i}\right) \le l_i < \log_r\left(\frac{1}{p_i}\right) + 1$$

$$\Rightarrow H_r(S) = \sum_{i=1}^{q} p_i \log_r \frac{1}{p_i} \le \sum_{i=1}^{q} p_i l_i < H_r(S) + 1$$

$$\Rightarrow H_r(S) \le L < H_r(S) + 1，$$

- 因此對於*Shannon-Fano*編碼，我們仍然可以有熵值做為其平均編碼長度之下限，它同時也是上限的一部分。

Shannon-Fano編碼

- 例：$p_1=p_2=1/4$，$p_3=p_4=p_5=p_6=1/8$，
  得到*Shannon-Fano*長度為：
  $$l_1=l_2=2,\quad l_3=l_4=l_5=l_6=3，$$
  我們接下來給值：
  $$s_1=00,\qquad s_3=100,$$
  $$s_2=01,\qquad s_4=101,$$
  $$s_5=110,$$
  $$s_6=111;$$
- 從*Kraft*不等式可以得到保證：一定有足夠的位元串，讓我們安排使其成為即時碼，我們上面的安排也符合了字首的條件，所以是即時碼。

Shannon-Fano編碼

- *Shannon-Fano*編碼可能有多差？
  - 假設一個符號源 $\{s_1, s_2\}$，其出現機率分別為

$$p_1 = 1 - \frac{1}{2^k},\ \ p_2 = \frac{1}{2^k} \quad (k \geq 2)$$

我們得到 $\log_2(\frac{1}{p_1}) \leq \log_2 2 = 1$，因此 l_1=1。

但是對於 l_2，我們會得到 $\log_2(\frac{1}{p_2}) \leq \log_2 2^k = k = l_2$

- 因此在*Huffman*編碼會用一個位元來表示每個符號的情況下，*Shannon-Fano*卻用一個位元來表示 s_1，用k個位元來表示 s_2。

Shannon-Fano編碼

- 在擔心*Shannon-Fano*編碼缺乏效率之前，先來計算它的平均碼長度。
  - 很明顯的，*Huffman*編碼因用一個位元來表示s_1及s_2，因此$L_H=1$。
  - 對於*Shannon-Fano*編碼，我們得到：

$$
\begin{aligned}
L_{SF} &= 1\times(1-\frac{1}{2^k})+k(\frac{1}{2^k}) \\
&= 1+\frac{k-1}{2^k}。
\end{aligned}
$$

Shannon-Fano編碼

- 於是我們得到下列之表：

K	$1+[(K-1)/2^K]$
2	1+ 1/4 = 1.25
3	1+ 1/4 = 1.25
4	1+ 3/16 = 1.1875
5	1+ 1/8 = 1.125
6	1+ 5/64 = 1.078125
⋮	⋮

etc.

不難發現，情況其實也還好。

擴充碼 (Extension code)

- 如果我們每一次不是編碼訊號的一個符號, 而是連續n個符號，那麼便有希望可以得到更接近於下限$H_r(S)$的平均編碼長度。
- 但更重要的是擴充碼的機率分布會比原來的機率分布更富變化；
- 因此如果我們充分地利用高次的擴充碼，那麼*Huffman*碼與*Shannon-Fano*編碼都會變得更有效率。

擴充碼（Extension code）

- 定義：符號源$S=\{s_1, s_2, \cdots, s_q\}$之第$n$次擴充碼的形式為 $s_{i_1}s_{i_2}\cdots s_{i_n}$，其機率為$Q_i = p_{i_1}p_{i_2}\cdots p_{i_n}$，其中。由$n$個原符號所構成的區段（*block*）變成單一的符號t_i，而其出現機率為Q_i。我們以$T=S^n$來表示。

擴充碼 (Extension code)

- 例 :假設我們有符號源S={s_1, s_2}，其編碼及出現機率分別為

	編碼
s_1 : p_1 = 2/3	$s_1 \longrightarrow 0$
s_2 : p_2 = 1/3	$s_2 \longrightarrow 1$ L_{avg} = 1

S的第二次擴充T=S_2出現機率則分別為：

	編碼
s_1s_1 : p_{11} = 4/9	$s_1s_1 \longrightarrow 1$
s_1s_2 : p_{12} = 2/9	$s_1s_2 \longrightarrow 01$
s_2s_1 : p_{21} = 2/9	$s_2s_1 \longrightarrow 000$
s_2s_2 : p_{22} = 1/9	$s_2s_2 \longrightarrow 001$

L_{avg} = 17/18 =0.9444

第三次擴充的L_{avg} 為 76/87 = 0.93827

第四次擴充的L_{avg} 為 0.93827.

擴充碼 (Extension code)

- 擴充碼的熵值可以計算如下：

$$
\begin{aligned}
H_r(T) = H_r(S^n) &= \sum_{i=1}^{q^n} Q_i \log_r(\frac{1}{Q_i}) \\
&= \sum_{i=1}^{q^n} Q_i \log_r(\frac{1}{p_{i_1} p_{i_2} \cdots p_{i_n}}) \\
&= \sum_{i=1}^{q^n} \sum_{k=1}^{n} Q_i \log_r(\frac{1}{p_{i_k}}) \\
&= \sum_{k=1}^{n} \sum_{i=1}^{q^n} Q_i \log_r(\frac{1}{p_{i_k}}),
\end{aligned}
$$

擴充碼 (Extension code)

$$\sum_{i=1}^{q^n} Q_i \log_r\left(\frac{1}{p_{i_k}}\right)$$

$$=\sum_{i_1=1}^{q}\sum_{i_2=1}^{q}\sum_{i_3=1}^{q}\cdots\cdots\sum_{i_k=1}^{q}\cdots\cdots\sum_{i_n=1}^{q} p_{i_1}p_{i_2}\cdots\cdots p_{i_k}\cdots p_{i_n}\log_r\left(\frac{1}{p_{i_k}}\right)$$

$$=\sum_{i_1=1}^{q} p_{i_1}\sum_{i_2=1}^{q} p_{i_2}\cdots\cdots\sum_{i_k=1}^{q} p_{i_k}\cdots\cdots\sum_{i_n=1}^{q} p_{i_n}\log_r\left(\frac{1}{p_{i_k}}\right)$$

$$=1\times 1\times\cdots\cdots\times\sum_{i_k=1}^{q} p_{i_k}\times\cdots\cdots\times 1\times\log_r\left(\frac{1}{p_{i_k}}\right)$$

$$=\sum_{i_k=1}^{q} p_{i_k}\log_r\left(\frac{1}{p_{i_k}}\right)=H_r(S),$$

所以 $H_r(T)=H_r(S^n)=\sum_{k=1}^{n} H_r(S)=n\times H_r(S)$。

擴充碼（Extension code）

- *Shannon*的無雜訊編碼定理(*Shannon 's noiseless coding theorem*)：

  一個碼的第*n*次擴充符合

  $$H_r(S) \le L < H_r(S) + 1/n$$

  證明：$H_r(S^n) \le L_n < H_r(S^n) + 1$

  $\Rightarrow \quad nH_r(S) \le L_n < nH_r(S) + 1$

  $\Rightarrow \quad H_r(S) \le L_n/n < H_r(S) + 1/n$

  由於每個擴充碼含有n個s_i比較好的估算是$L=L_n/n$，因此只要擴充碼的次數n夠大，我們可以讓平均編碼長度L如我們所願地逼近於熵$H_r(S)$。

擴充碼（Extension code）

- *Huffman*編碼為所有即時碼中最好的，假設其平均編碼長度為L_H，因此 $L_H \log_2 r \le L_{SF} \log_2 r$，對於第*n*次擴充，我們可以得到

$$H_2(S) \le \frac{L_H^n(n)\log_2 r}{n} \le \frac{L_{SF}^n(n)\log_2 r}{n} < H_2(S) + \frac{\log_2 r}{n}$$

其中 L_H^n及L_{SF}^n分別表示*Huffman*編碼及*Shannon-Fano*編碼之第*n*次擴充碼的平均編碼長度。

- 理論上，當*n*取得夠大，*Huffman*編碼與*Shannon-Fano*編碼的效率幾乎沒有差別。實用上，當*n*很小時，*Shannon-Fano*編碼可能明顯地不如*Huffman*編碼。

擴充碼 (Extension code)

- 例：

  $s_1 : p_1 = 2/3, \ s_2 : p_2 = 1/3$

  $H_2(S) = 2/3 \log_2(3/2) + (1/3)\log_2 3 = 0.9182958....$

  (lower bound)

n	$L^{H}_{avg}(T)/n$	$L^{SF}_{avg}(T)/n$
1	1.00000	1.33333
2	0.94444	1.33333
3	0.93827	1.00000
4	0.93827	1.08333
5	0.92263	0.93333

馬可夫過程之熵
(Entropy of a Markov process)

- 在第 *j* 階的馬可夫過程（*j-th order Markov process*）裡 $p(s_i \mid s_{i_1}, s_{i_2}, \ldots, s_{i_j})$ 指的是你依序看到了 $s_{i_1}, s_{i_2}, \ldots, s_{i_j}$ 後，緊接著看到 s_i 的條件機率。則

$$p\left(s_i \middle| s_{i_1}, s_{i_2}, \ldots, s_{i_j}\right) = p\left(s_i \middle| s_{i_1}, s_{i_2}, \ldots, s_{i_j}, \ldots\right)$$

  表示知道過去j個符號的結果與知道過去所有符號的結果對接下來發生的符號的機率是一樣的，即僅和目前的狀態有關。

- 一個 *DMS*（*discrete memoryless source*）獨立地使用每一個符號；對於前一個出現的符號是什麼完全忽略不計。一個 *j* 個記憶源（*j-memory source*）模式則會考慮前面剛剛才出現過的 *j* 個符號，這個觀念就正好相當於一個第 *j* 階的馬可夫過程。

馬可夫過程之熵 (Entropy of a Markov process)

- 一個相同且獨立分佈(independently and identically distributed, 簡稱iid)的符號源為馬可夫過程的一個特例，其關係為

$$p(s_i|s_{i_1}) = p(s_i)$$

- 範例：一個符號源S={0,1},若滿足下列關係即為iid。

  $p(0|0)=p(0|1)=p(0)$，$p(1|1)=p(1|0)=p(1)$

- 範例：如果有一個符號源S={w,b}
  - 假設利用機率模型並將其視為iid的情形，令$p(w)=0.8$，$p(b)=0.2$，則$H_2(S)=0.206$ bits。
  - 但若將其視為一階馬可夫過程，令$p(w|w)=0.99, p(b|w)=0.01$，$p(b|b)=0.7$，$p(w|b)=0.3$，則可得到$p(w)=30/31$，$p(b)=1/31$。$H_2(S)=0.107$ bits。

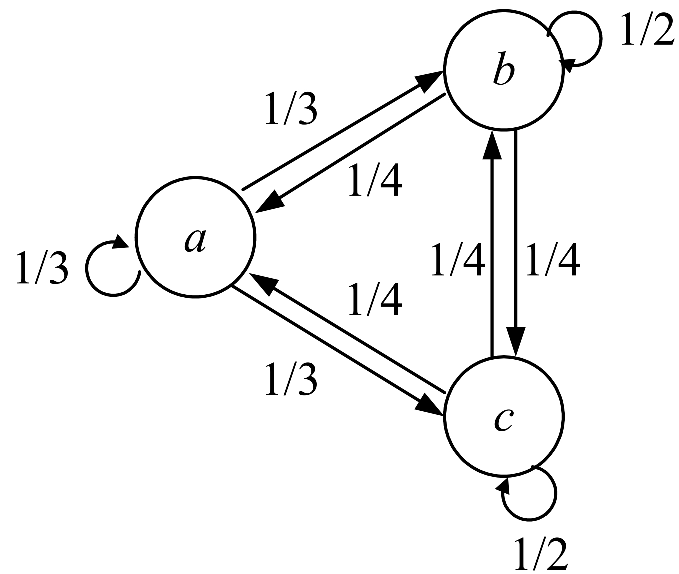
→一階馬可夫過程的entropy比iid要小，表示該source存在符號間的冗贅。

s_{i_1}, s_i	$p(s_i\|s_{i_1})$	$p(s_i)$	$p(s_{i_1}, s_i)$
ww	0.99	30/31	0.958
wb	0.01	30/31	0.010
bw	0.7	1/31	0.023
bb	0.3	1/31	0.010

馬可夫過程之熵 (Entropy of a Markov process)

- 例：

  $p(a|a)=1/3$，$p(b|a)=1/3$，$p(c|a)=1/3$，
  $p(a|b)=1/4$，$p(b|b)=1/2$，$p(c|b)=1/4$，
  $p(a|c)=1/4$，$p(b|c)=1/4$，$p(c|c)=1/2$。



馬可夫過程之熵 (Entropy of a Markov process)

- 我們可以用矩陣的形式來表示有向圖，矩陣左邊所標示的是目前的狀態，上方所標示的是下一個狀態，矩陣內的值則是由一個狀態轉換到另一個狀態的機率。

$$
\begin{array}{c c}
 & \begin{array}{ccc} a & b & c \end{array} \\
\begin{array}{c} a \\ b \\ c \end{array} & \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}
\end{array}
$$

- 一個轉換矩陣中任何一橫列的值之和必然為一，因為目前的狀態必然要轉換到某個狀態。

馬可夫過程之熵
(Entropy of a Markov process)

$$p_a=p_a\times p(a/a)+ p_b\times p(a/b)+p_c\times p(a/c)，$$
$$p_b=p_a\times p(b/a)+ p_b\times p(b/b)+p_c\times p(b/c)，$$
$$p_c=p_a\times p(c/a)+ p_b\times p(c/b)+p_c\times p(c/c)；$$

這三個方程式只有兩個是獨立的，因此取前兩個再加上已知的$p_a+p_b+p_c=1$代數運算後，我們得到所謂的平衡解（*equilibrium solution*）為$p_a=3/11$，$p_b=p_c=4/11$。

馬可夫過程之熵 (Entropy of a Markov process)

- 假設我們已經讀到前面m個符號為 $s_{i_1}, s_{i_2}, \ldots, s_{i_m}$，那麼下一個符號是$s_i$的機率為 $p(s_i \mid s_{i_1}, s_{i_2}, \ldots, s_{i_m})$，符號$s_i$所得到的資訊量為，當我們已經看到（在這個狀態）時再收到符號s_i所得到的資訊量為

$$I(s_i \mid s_{i_1}, s_{i_2}, \ldots, s_{i_m}) = \log_2\left(\frac{1}{p(s_i \mid s_{i_1}, s_{i_2}, \ldots, s_{i_m})}\right)$$

含所有s_i的符號源S的條件熵函數（*conditional entropy*）很自然地可以寫成下列的方程式

$$H(S \mid s_{i_1}, s_{i_2}, \ldots, s_{i_m})$$

$$= \sum_S p(s_i \mid s_{i_1}, s_{i_2}, \ldots, s_{i_m}) \log_2\left(\frac{1}{p(s_i \mid s_{i_1}, s_{i_2}, \ldots, s_{i_m})}\right),$$

馬可夫過程之熵 (Entropy of a Markov process)

- 定義整個馬可夫過程的熵為所有在一個狀態的機率乘以在該狀態下的條件熵函數之和，即

$$H(S)=\sum_{S^m} p(s_{i_1},s_{i_2},\ldots,s_{i_m})H(S|s_{i_1},s_{i_2},\ldots,s_{i_m})$$

$$=\sum_{S^m} p(s_{i_1},s_{i_2},\ldots,s_{i_m})\sum_{S} p(s_i|s_{i_1},s_{i_2},\ldots,s_{i_m})\log_2\left(\frac{1}{p(s_i|s_{i_1},s_{i_2},\ldots,s_{i_m})}\right)$$

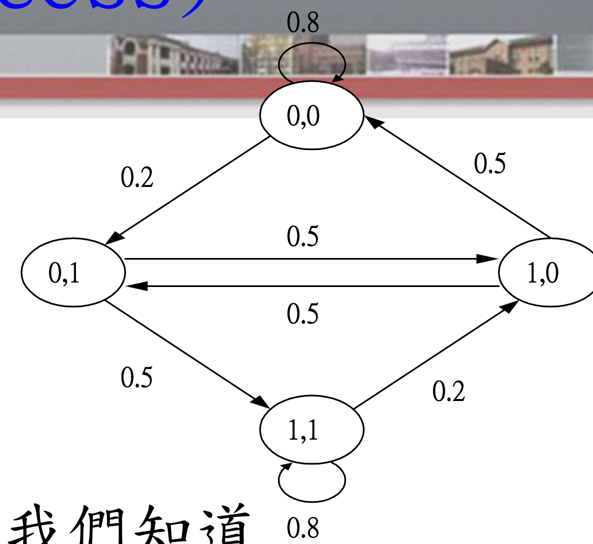
$$=\sum_{S^m}\sum_{S} p(s_{i_1},s_{i_2},\ldots,s_{i_m})p(s_i|s_{i_1},s_{i_2},\ldots,s_{i_m})\log_2\left(\frac{1}{p(s_i|s_{i_1},s_{i_2},\ldots,s_{i_m})}\right)$$

$$=\sum_{S^{m+1}} p(s_{i_1},s_{i_2},\ldots,s_{i_m},s_i)\log_2\left(\frac{1}{p(s_i|s_{i_1},s_{i_2},\ldots,s_{i_m})}\right)$$

馬可夫過程之熵 (Entropy of a Markov process)

- 例：

$$p(0|0,0)=0.8=p(1|1,1)$$
$$p(1|0,0)=0.2=p(0|1,1)$$
$$p(0|0,1)=0.5=p(1|1,0)$$
$$p(1|0,1)=0.5=p(0|1,0)$$



利用這個馬可夫過程所表現的對稱性，我們知道

$$p(0,0)=p(1,1) \text{ 且 } p(0,1)=p(1,0)$$

因此我們得到下面的方程式：

$$p(0,0)=0.5p(1,0)+0.8p(0,0)=0.5p(0,1)+0.8p(0,0)$$
$$p(0,1)=0.2p(0,0)+0.5p(1,0)=0.2p(0,0)+0.5p(0,1)$$

這兩個方程式都可化簡成$0.2p(0,0)-0.5p(0,1)=0$

馬可夫過程之熵 (Entropy of a Markov process)

由於馬可夫過程一定處在其中某一個狀態，所以

$$p(0,0)+p(0,1)+p(1,0)+p(1,1)=1$$

或$2p(0,0)+2p(0,1)=1$

$$\Rightarrow p(0,0)=5/14=p(1,1)\ ;\ p(0,1)=2/14=p(1,0).$$

s_{i_1}, s_{i_2}, s_i	$p(s_i \mid s_{i_1}, s_{i_2})$	$p(s_{i_1}, s_{i_2})$	$p(s_{i_1}, s_{i_2}, s_i)$
0 0 0	0.8	5/14	4/14
0 0 1	0.2	5/14	1/14
0 1 0	0.5	2/14	1/14
0 1 1	0.5	2/14	1/14
1 0 0	0.5	2/14	1/14
1 0 1	0.5	2/14	1/14
1 1 0	0.2	5/14	1/14
1 1 1	0.8	5/14	4/14

馬可夫過程之熵
(Entropy of a Markov process)

從這個表我們可以計算得：

$$
\begin{aligned}
H(S) &= \sum_{2^3} p(s_{i_1}, s_{i_2}, s_i) \log_2 \left(\frac{1}{p(s_i \mid s_{i_1}, s_{i_2})}\right) \\
&= 2\left[\frac{4}{14}\log_2\left(\frac{1}{0.8}\right)\right] + 2\left[\frac{1}{14}\log_2\left(\frac{1}{0.2}\right)\right] + 4\left[\frac{1}{14}\log_2\left(\frac{1}{0.5}\right)\right] \\
&= \frac{4}{7}(\log_2 10 - 3) + \frac{1}{7}(\log_2 10 - 1) + \frac{2}{7}\log_2 2 \\
&= 0.801377 \text{ 位元。}
\end{aligned}
$$

馬可夫過程之熵
(Entropy of a Markov process)

- 很明顯的，由於馬可夫過程也考慮進去符號間冗贅，我們可以利用它來改進壓縮的結果。

- 每一個狀態都各有一個解碼樹或相同功能的編碼表，共q^m個（第m階馬可夫過程）。依每一個狀態上離開該狀態的轉換之機率變化情況 ，我們收益或多或少。但是當我們提高馬可夫過程的階次，我們所得到的邊際收益將愈來愈小而狀態數卻急遽增加。