最適な符号

符号を作る方法であるシャノン・ファノ符号化とハフマン符号化を示します。ハフマン符号化は $\mathcal{B} = \{0, 1\}$ の場合にしています。

文字列に確率を組み込みます。例えば、サイコロを振ったとき出る目は 1 から 6 でそれぞれの確率は 1/6 です。サイコロの出た目を出力して左から並べていくなら

$$5135624126\cdots$$

といった文字列が作られます。この文字列はアルファベット $\mathcal{A} = \{1, 2, 3, \ldots, 6\}$ とその元が出力される確率

$$\boldsymbol{p} = \{p_1, p_2, \ldots, p_6\} = \{\frac{1}{6}, \frac{1}{6}, \ldots . \frac{1}{6}\}$$

に依存しています。このように、文字列に使われるアルファベットとアルファベットの元が現れる確率を合わせた $(\mathcal{A}, \boldsymbol{p})$ を情報源 (source) と言います（太字にしているのは区別のためで、ベクトルとは無関係)。このときの $\boldsymbol{p}$ は $\mathcal{A}$ の元の数を n とすれば

$$p_1 + p_2 + \cdots + p_n = 1$$

となっている必要があります。

ここでは元 $a_1, a_2, \ldots$ の確率は

$$p_1 = P(a_1)\ ,\ p_2 = P(a_2)\ ,\ \ldots$$

と書くことにし、p_i の確率は全て独立とします。独立は、$\mathcal{A}$ の元 $a_i, a_j\ (i \neq j)$ が同時に現れる結合確率 $P(a_i, a_j)$ に対して

$$P(a_i, a_j) = P(a_i)P(a_j)$$

となるという意味です。また、確率が独立である情報源は無記憶情報源 (memoryless source) と言われます。ここでは無記憶情報源とします。

$\mathcal{A}$ の元の数を α、元を $a_i\ (i = 1, 2, \ldots, \alpha)$、$\mathcal{A}$ の元から作られる文字列（$\mathcal{S}_A^*$ の元）の長さを r、符号アルファベット $\mathcal{B}$ による符号語 $\phi(a_i)\ (i = 1, 2, \ldots, \alpha)$ の長さを l_i とします。$\mathcal{A}$ による文字列の長さ r が十分大きければ、文字列の中の a_i の個数は $rP(a_i)$ に近くなっているはずです（確率 $P(a_i)$ で文字列の中に現れるから)。そうすると、$\mathcal{S}_A^*$ の長さ r の文字列を符号化したとき、長さ l_i の $\phi(a_i)$ の符号語が $rP(a_i)$ 個なので、符号化された文字列の近似的な長さは

$$rP(a_1)l_1 + rP(a_2)l_2 + \cdots + rP(a_\alpha)l_\alpha = r(P(a_1)l_1 + P(a_2)l_2 + \cdots + P(a_\alpha)l_\alpha) \tag{1}$$

$\mathcal{S}_A^*$ の文字列に含まれる a_i の個数が λ_i なら、文字列に含まれる $\mathcal{A}$ の元の個数は

$$\Lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_\alpha$$

文字列が十分長いとしているので、確率は $P(a_i) = \lambda_i/\Lambda$ です。$\mathcal{S}_B^*$ の文字列では、λ_1 個の $\phi(a_1)$、λ_2 個の $\phi(a_2)$、$\cdots$ というようになっており、文字列に含まれる符号語の数は Λ です。なので、$\phi(a_i)$ が文字列の中に現れる確率は $\lambda_i/\Lambda = P(a_i)$ です。このため、(1) の括弧内は符号語の平均的な長さになっていて

$$< l >= P(a_1)l_1 + P(a_2)l_2 + \cdots + P(a_\alpha)l_\alpha$$

これを $(\mathcal{A}, \boldsymbol{p})$ での平均符号長 (average word length) と言います。

長い文字列では相手に伝えるときに問題が起きやすくなるので、符号化で重要なのはなるべく短い文字列にすることです。このため、文字列を短くする符号を作る必要があり、その目安として平均符号長が使われています。情報源と符号アルファベットに対して平均符号長が最小になる一意復号可能な符号化関数（符号）は最適 (optimal) と言われ、最適な符号化関数を作ること (平均符号長が最小になる符号を作ること) を最適化 (optimization) と言います。最適な符号化関数による符号にも最適と言っていきます。

符号と平均符号長の例を見ておきます。情報源が

$$\mathcal{A} = \{a_1, a_2, a_3\}\ ,\ \boldsymbol{p} = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$$

と与えられているとし、これによって文字列 r が作られているとします。$\mathcal{B} = \{0, 1\}$ として符号語を

$$\phi(a_1) = 0\ ,\ \phi(a_2) = 10\ ,\ \phi(a_3) = 11$$

と作ったとすれば、文字列 r を符号化したときの平均符号長は

$$< l >= \frac{1}{4} \times 1 + \frac{1}{2} \times 2 + \frac{1}{4} \times 2 = \frac{7}{4}$$

一方で、符号語を

$$\phi(a_1) = 10\ ,\ \phi(a_2) = 0\ ,\ \phi(a_3) = 11$$

と作ってみると

$$< l >= \frac{1}{4} \times 2 + \frac{1}{2} \times 1 + \frac{1}{4} \times 2 = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \frac{3}{2} < \frac{7}{4}$$

となって平均符号長が小さくなります (高い確率では符号語の長さが小さい方を選んだだけ)。このように、平均符号長は符号語の作り方によって変わるので、平均符号長を小さくする（最適化する）にはどうすればいいのかが重要になります。

符号化関数は一意復号可能として、平均符号長の関係を求めます。平均符号長を短くしたいですが 0 にすることはできないので、どこかに下限があります。それを H として

$$H \leq < l >$$

と仮定し、H を求めます。一意復号可能と prefix-free はクラフトの不等式が関係しているので、クラフトの不等式が使えるようにします。

$\mathcal{A}$ の元 a_i $(i = 1, 2, \ldots, \alpha)$ から作られる符号語の長さが l_i になっているとし、符号アルファベット $\mathcal{B}$ の元の数を β として

$$K = \sum_{i=1}^{\alpha} \frac{1}{\beta^{l_i}}$$

これを変形して

$$\frac{1}{K} \sum_{i=1}^{\alpha} \frac{1}{\beta^{l_i}} = 1$$

と作ると、$Q_i = 1/K\beta^{l_i}$ は確率のようになっています。このことから、対数の不等式

$$\sum_{i=1}^{n} x_i \log_b \frac{1}{x_i} \leq \sum_{i=1}^{n} x_i \log_b \frac{1}{y_i} \quad (\sum_{i=1}^{n} x_i = 1 \ , \ \sum_{i=1}^{n} y_i = 1) \tag{2}$$

を使います (下の補足参照)。底を b とする対数を取ってみると

$$\log_b Q_i = \log_b \frac{1}{K\beta^{l_i}} = \log_b \frac{1}{K} + \log_b \frac{1}{\beta^{l_i}}$$

b を β に選べば

$$\log_\beta \frac{1}{Q_i} = \log_\beta K + \log_\beta \beta^{l_i} = \log_\beta K + l_i$$

これの左辺で (2) を使うと

$$\sum_{i=1}^{\alpha} P(a_i) \log_\beta \frac{1}{P(a_i)} \leq \sum_{i=1}^{\alpha} P(a_i) \log_\beta \frac{1}{Q_i} = \log_\beta K \sum_{i=1}^{\alpha} P(a_i) + \sum_{i=1}^{\alpha} P(a_i) l_i = \log_\beta K + < l > \tag{3}$$

一意復元可能な符号化関数が存在するにはクラフトの不等式 $K \leq 1$ が成立する必要があるので、$K \leq 1$ を要求します。また、この要求によって prefix-free にもなります。

そうすると、$\log_\beta K \leq 0$ $(K \leq 1)$ から

$$-\sum_{i=1}^{\alpha} P(a_i) \log_\beta P(a_i) \leq < l > \quad (\log_\beta K + < l > \leq < l >) \tag{4}$$

となり、この不等式を満たす prefix-free な符号化関数が存在することになります。底を任意の b にするなら

$$\log_\beta P(a_i) = \frac{\log_b P(a_i)}{\log_b \beta}$$

から

$$-\frac{1}{\log_b \beta}\sum_{i=1}^{\alpha} P(a_i)\log_b P(a_i) \leq < l > \qquad (\log_\beta K + < l > \leq < l >) \tag{5}$$

として下限が与えられます。このときの

$$H_b(P) = -\sum_{i=1}^{\alpha} P(a_i)\log_b P(a_i)$$

をエントロピー (entropy) と呼びます。ここではエンロトピーについては触れずに先に進みます。

今度は平均符号長の上限を求めます。(2) で等式になるのは $x_i = y_i$ のときなので

$$K = 1 \quad (\log_b K = 0)$$

$$P(a_i) = Q_i = \frac{1}{K\beta^{l_i}}$$

となっているなら、(3) の不等式での下限 $H_\beta(P) =< l >$ になります。このときの β^{l_i} は $i = 1, 2, \ldots, \alpha$ に対して

$$\beta^{l_i} = \frac{1}{P(a_i)}$$

β, l_i は正の整数なので、等式が正確に成立するには確率の逆数が整数でないといけないという制限があることに注意してください。

例えば、$\mathcal{B} = \{0, 1\}$ のときでは、l_i は符号語 $\phi(a_i)$ $(i = 1, 2, \ldots, \alpha)$ の長さなので、$l_1 = 1, l_2 = 2, \ldots$ のように与えたとすれば

$$\beta^{l_1} = 2 = \frac{1}{P(a_1)} \quad \Rightarrow \; P(a_1) = \frac{1}{2}$$

$$\beta^{l_2} = 2^2 = \frac{1}{P(a_2)} \quad \Rightarrow \; P(a_2) = \frac{1}{4}$$

$$\beta^{l_3} = 2^3 = \frac{1}{P(a_3)} \quad \Rightarrow \; P(a_3) = \frac{1}{8}$$

$$\vdots$$

このように、確率が $1/2^n$ になっているなら $H_\beta(P) =< l >$ になり、平均符号長を最も短くできます。

$\beta^{l_i} = 1/P(a_i)$ となっていなくても、$1/P(a_i)$ になるべく近くなるように l_i を与えれば平均符号長は短くなります。そのような正の整数 l_i が与えられたとして、そこからさらに l_i を小さくして $l_i - 1$ にしたとします。l_i は大きな値から小さくしていき

$$\beta^{l_i} \geq \frac{1}{P(a_i)}$$

となるように選んだとすれば、$l_i - 1$ では $1/P(a_i)$ を通り過ぎないといけないので ($\beta^{l_i-1} \geq 1/P(a_i)$ だと β^{l_i} よりも $1/P(a_i)$ に近くなる)

$$\beta^{l_i-1} < \frac{1}{P(a_i)}$$

底を β とする対数を取ると

$$\begin{aligned} \log_\beta \beta^{l_i-1} &< \log_\beta \frac{1}{P(a_i)} \\ l_i - 1 &< -\log_\beta P(a_i) \\ l_i &< -\log_\beta P(a_i) + 1 \end{aligned}$$

平均符号長に書き換えれば

$$< l >= \sum_{i=1}^{\alpha} l_i P(a_i) < \sum_{i=1}^{\alpha} (-\log_\beta P(a_i) + 1) P(a_i) = -\sum_{i=1}^{\alpha} P(a_i) \log_\beta P(a_i) + 1 = H_\beta(P) + 1$$

となり、上限が $H_\beta(P) + 1$ で与えられます。底を b にするなら

$$< l > < \frac{H_b(P)}{\log_b \beta} + 1 \tag{6}$$

となります。

結果をまとめます。$K \leq 1$ を要求したとき、(5),(6) から平均符号長が

$$\frac{H_b(P)}{\log_b \beta} \leq < l > < \frac{H_b(P)}{\log_b \beta} + 1$$

となる prefix-free な符号化関数が存在します。そして、$\beta^{l_i} = 1/P(a_i)$ のとき、平均符号長は最小になります。これをシャノンの情報源符号化定理 (Shanonn's source coding theorem) や noiseless coding theorem と言います。

シャノンの情報源符号化定理を利用する符号化をシャノン・ファノ符号化 (Shannon-Fano rule) と言います。例として、確率が $\mathcal{A}$ の元 a_i $(i = 1, 2, 3, 4)$ と確率

$$\boldsymbol{p} = \{P(a_1), P(a_2), P(a_3), P(a_4)\} = \{\frac{1}{5}, \frac{2}{5}, \frac{1}{10}, \frac{3}{10}\}$$

による情報源があるとします。$\mathcal{B} = \{0, 1\}$ とすれば、符号語の長さ l_i を

$$2^{l_i} \geq \frac{1}{P(a_i)}$$

となる、最小の正の整数に選ぶことで平均符号長を最小に近づけられます。これに $\boldsymbol{p}$ を入れれば

$$5 \leq 2^{l_1}\ ,\ \frac{5}{2} \leq 2^{l_2}\ ,\ 10 \leq 2^{l_3}\ ,\ \frac{10}{3} \leq 2^{l_4}$$

これらから、最小の l_i を選ぶと

$$l_1 = 3\ ,\ l_2 = 2\ ,\ l_3 = 4\ ,\ l_4 = 2$$

この長さになるように prefix-free の符号を作るのがシャノン・ファノ符号化です。平均符号長と情報エントロピーを求めてみると

$$< l >= 3 \times \frac{1}{5} + 2 \times \frac{2}{5} + 4 \times \frac{1}{10} + 2 \times \frac{3}{10} = 2.4$$

$$H_2 = -\sum_{i=1}^{4} P(a_i) \log_2 P(a_i) = -\frac{1}{5} \log_2 \frac{1}{5} - \frac{2}{5} \log_2 \frac{2}{5} - \frac{1}{10} \log_2 \frac{1}{10} - \frac{3}{10} \log_2 \frac{3}{10} \simeq 1.8$$

となるので、平均符号長の値は最適となる値 H_2 とはそれなりに離れています。このように、β^{l_i} が整数という制限のために一般的には近似的な値を使うことになるので、これがどこまで最適なのかははっきりしないです。これを改善する方法としてブロック符号 (block coding) を用いた拡張がありますが、話がそれるので省きます。

シャノン・ファノ符号化には制限があるので、より一般的に使えるハフマン (Huffman) 符号化を見ていきます。ここから符号アルファベットは $\mathcal{B} = \{0, 1\}$ とします。

情報源を $(\mathcal{A}, \boldsymbol{p})$、$\phi$ を最適な prefix-free の符号化関数としたとき、次の性質を持ちます。

(i) $P(a) > P(a')$ なら $\phi(a), \phi(a')$ の長さ l, l' は $l \leq l'$。

符号語 $\phi(a)$ の長さを l、$\phi(a')$ の長さを l' とします。別の符号化関数 ψ を $\psi(a) = \phi(a')$、$\psi(a') = \phi(a)$ とし、他の元に対しては同じ符号語を与えるとします。ϕ での平均符号長 $< l(\phi) >$ と ψ での平均符号長 $< l(\psi) >$ は

$$< l(\phi) >= lP(a) + l'P(a') + O\ ,\ < l(\psi) >= l'P(a) + lP(a') + O$$

O は a, a' 以外での部分です。ϕ は最適なので $< l(\psi) > \geq < l(\phi) >$ となっていることから

$$0 \leq < l(\psi) > - < l(\phi) >= (l' - l)P(a) + (l - l')P(a') = (l' - l)(P(a) - P(a'))$$

よって、$P(a) > P(a')$ なら $l \leq l'$ です。もしくは、$l > l'$ なら $P(a) \leq P(a')$ とも言えます。

(ii) 最も長い符号語の中に $s0, s1$ $(s \in \mathcal{S}_B^*)$ の符号語が存在する。

最も長い符号語が 1 個あるとします。prefix-free なので、その符号語を文字列 s $(s \in \mathcal{S}_B^*)$ と $\mathcal{B}$ の元 b によって sb と書いたとすると、b を外して s にしても prefix-free のままです。そうすると、最も長いと言っていながら最後の 1 文字を外しても符号語として使えてしまうので（prefix-free なら s は他の符号語になっていない)、平均符号長を最小にする符号語となっていません。このため、最も長い符号語が 1 個では矛盾してしまいます。

というわけで、同じ文字列 s を持つ最も長い符号語は 2 個います ($\mathcal{B}$ の元が 2 個なので 3 個作れない)。実際に、それらを sb_1 と sb_2 とすれば、b_1 (b_2) を外すと s, sb_2 (sb_1, s) となってしまい prefix-free でなくなるので、b_1 もしくは b_2 を外して符号語の長さを減らすことはできません。よって、最も長い符号語の中には $s0, s1$ が含まれます。

ハフマン符号化の手順を示します。ハフマン符号化は (i) を利用して確率を割り振ることから始めます。元が 5 個の $\mathcal{A}_5$ とし、確率が

$$(\mathcal{A}_5, \boldsymbol{p}_5):\ P(a_1) = 0.4\ ,\ P(a_2) = 0.3\ ,\ P(a_3) = 0.15\ ,\ P(a_4) = 0.1\ ,\ P(a_5) = 0.05$$

と与えられているとします。(i) から文字列が長い方が確率が低くなるので、$\mathcal{A}_5$ の確率が低い 2 個を 1 個にまとめて、4 個の元による $\mathcal{A}_4$ に移動させるとします。そうすると、a'_4 の確率を $P(a'_4) = P(a_4) + P(a_5)$ として

$$(\mathcal{A}_4, \boldsymbol{p}_4):\ P(a_1) = 0.4\ ,\ P(a_2) = 0.3\ ,\ P(a_3) = 0.15\ ,\ P(a'_4) = 0.15$$

このように確率の低いもの同士を足すことを繰り返して、今は $\mathcal{B} = \{0, 1\}$ なので $\mathcal{A}_2$ まで作ると

$$(\mathcal{A}_3, \boldsymbol{p}_3):\ P(a_1) = 0.4\ ,\ P(a_2) = 0.3\ ,\ P(a'_3) = 0.3$$

$$(\mathcal{A}_2, \boldsymbol{p}_2):\ P(a_1) = 0.4\ ,\ P(a'_2) = 0.6$$

この確率に (ii) を使って符号語を割り振っていきます。

$\mathcal{B} = \{0, 1\}$ としているために $(\mathcal{A}_2, \boldsymbol{p}_2)$ での符号語は $0, 1$ だけで、どちらの確率に当てはめても最適です。なので、各 $(\mathcal{A}_n, p)$ での最適な符号化関数を ϕ_n として、符号語を

$$(\mathcal{A}_2, \boldsymbol{p}_2):\ \phi_2(a_1) = 1\ ,\ \phi_2(a'_2) = 0$$

とします。次の $(\mathcal{A}_3, \boldsymbol{p}_3)$ では 3 個の元があるので、符号語を $0,\ 1,\ 00,\ 01,\ 10,\ 11$ の中から適切に 3 個選ぶことになります。$(\mathcal{A}_3, \boldsymbol{p}_3)$ において最適であるなら、(ii) から最も長い符号語として $s0, s1$ がいるはずです。$\mathcal{A}_2$ から $\mathcal{A}_3$ へは 0 を割り振った a'_2 から分岐させているので、s としては 0 を選ぶことになり

$$(\mathcal{A}_3, \boldsymbol{p}_3):\ \phi_3(a_1) = 1\ ,\ \phi_3(a_2) = 01\ ,\ \phi_3(a'_3) = 00$$

このように作ると prefix-free でなくなる 0 は自動的に除外されます。これを $\mathcal{A}_5$ まで続ければ

$$(\mathcal{A}_2, \boldsymbol{p}_2):\ \phi_2(a_1) = 1\ ,\ \phi_2(a'_2) = 0$$

$$(\mathcal{A}_3, \boldsymbol{p}_3):\ \phi_3(a_1) = 1\ ,\ \phi_3(a_2) = 01\ ,\ \phi_3(a'_3) = 00$$

$$(\mathcal{A}_4, \boldsymbol{p}_4):\ \phi_4(a_1) = 1\ ,\ \phi_4(a_2) = 01\ ,\ \phi_4(a_3) = 001\ ,\ \phi_4(a'_4) = 000$$

$$(\mathcal{A}_5, \boldsymbol{p}_5):\ \phi_5(a_1) = 1\ ,\ \phi_5(a_2) = 01\ ,\ \phi_5(a_3) = 001\ ,\ \phi_5(a_4) = 0001\ ,\ \phi_5(a_5) = 0000$$

となり、元 $\{a_1, a_2, a_3, a_4, a_5\}$ の確率 $\{0.4,\ 0.3,\ 0.15,\ 0.1,\ 0.05\}$ に対応した最適な符号 $\{1,\ 01,\ 001,\ 0000,\ 0001\}$ が求まります。このように (i) を使って確率を $(\mathcal{A}_n, \boldsymbol{p}_n)$ から $(\mathcal{A}_2, \boldsymbol{p}_2)$ まで割り振り、そこに (ii) を使って $(\mathcal{A}_2, \boldsymbol{p}_2)$ から $(\mathcal{A}_n, \boldsymbol{p}_n)$ へ符号語を作って当てはめていくのがハフマン符号化の手順です。この手順から分かるように、ハフマン符号化は $\mathcal{A}_n$ の元の確率が分かっているときに使えます。また、今は $(\mathcal{A}_n, \boldsymbol{p}_n)$ での $\phi_n(a_n)$ の右端が 0 になるように作りましたが、確率が高いほうに $s1$ を配置させるとする場合もあります。

今は $\mathcal{A}_n$ の元 a_{n-1} と a_n での確率を足すという分かりやすい状況でしたが、常にこうなっているわけではないです。例えば、確率を

$$(\mathcal{A}_5, \boldsymbol{p}_5):\ P(a_1) = 0.35\ ,\ P(a_2) = 0.25\ ,\ P(a_3) = 0.18\ ,\ P(a_4) = 0.12\ ,\ P(a_5) = 0.1$$

とすれば

$$(\mathcal{A}_4, \boldsymbol{p}_4):\ P(a_1) = 0.35\ ,\ P(a_2) = 0.25\ ,\ P(a_3) = 0.18\ ,\ P(a'_4) = 0.22$$

$$(\mathcal{A}_3, \boldsymbol{p}_3):\ P(a_1) = 0.35\ ,\ P(a_2) = 0.25\ ,\ P(a'_3) = 0.4$$

$$(\mathcal{A}_2, \boldsymbol{p}_2):\ P(a'_1) = 0.6\ \ ,\ P(a'_3) = 0.4$$

これは $\mathcal{A}_3$ から $\mathcal{A}_2$ に行くときに a_2 と a'_3 でなく、a_1 と a_2 で和を取っています。このときの符号語の割り振りは、今度は $\mathcal{A}_2$ で $0, 1$ の並びにして始めてみると

$$(\mathcal{A}_2, \boldsymbol{p}_2):\ \phi_2(a'_1) = 0\ ,\ \phi_2(a'_3) = 1$$

$$(\mathcal{A}_3, \boldsymbol{p}_3):\ \phi_3(a_1) = 00\ ,\ \phi_3(a_2) = 01\ ,\ \phi_2(a'_3) = 1$$

$$(\mathcal{A}_4, \boldsymbol{p}_4):\ \phi_4(a_1) = 00\ ,\ \phi_4(a_2) = 01\ ,\ \phi_4(a_3) = 10\ ,\ \phi_4(a'_4) = 11$$

$$(\mathcal{A}_5, \boldsymbol{p}_5):\ \phi_5(a_1) = 00\ ,\ \phi_5(a_2) = 01\ ,\ \phi_5(a_3) = 10\ ,\ \phi_5(a_4) = 110\ ,\ \phi_5(a_5) = 111$$

となります。

ハフマン符号化が成立するためには、(ii) を使って符号語を割り振った手順から分かるように、$(\mathcal{A}_n, \boldsymbol{p}_n)$ で最適なら $(\mathcal{A}_{n+1}, \boldsymbol{p}_{n+1})$ も最適である必要があります。これを示します。

ハウマン符号化での $(\mathcal{A}_n, \boldsymbol{p}_n)$ と $(\mathcal{A}_{n+1}, \boldsymbol{p}_{n+1})$ の定義をはっきりさせます。$\mathcal{A}_{n+1}$ の元は $\{a_1, a_2, \ldots, a_n, a_{n+1}\}$ とします。この元の並びに対応するように、文字列に現れる確率 $\boldsymbol{p}_{n+1} = \{p_1, p_2, \cdots, p_n, p_{n+1}\}$ $(p_i = P(a_i))$ が与えられており

$$p_1 \geq p_2 \geq \cdots \geq p_n \geq p_{n+1}$$

となっているとします。このときの $(\mathcal{A}_n, \boldsymbol{p}_n)$ を

$$\mathcal{A}_n = \{a_1, a_2, \ldots, a'_n\}\ ,\ \boldsymbol{p}_n = \{p_1, p_2, \cdots, p'_n\} \quad (p'_n = p_n + p_{n+1})$$

と定義します。$\mathcal{A}_n$ の元 a'_n は文字列における a_n, a_{n+1} を置き換えたものです。

符号化関数を定義します。まず、上での手順のように $n = 2$ から始める場合を想定して定義を与えます。ϕ_n は $(\mathcal{A}_n, \boldsymbol{p}_n)$ での prefix-free の符号化関数とし、$\phi_n(a') = s$ $(a' \in \mathcal{A}_n)$ をその符号語としたとき、$(\mathcal{A}_{n+1}, \boldsymbol{p}_{n+1})$ での prefix-free の符号化関数 ϕ_{n+1} は符号語を

$$\phi_{n+1}(a_i) = s0\ ,\ \phi_{n+1}(a_j) = s1 \quad (a_i, a_j \in \mathcal{A}_{n+1})$$

$$\phi_{n+1}(a) = \phi_n(a) \quad (a \neq a_i, a_j\ ,\ a \in \mathcal{A}_n, \mathcal{A}_{n+1})$$

と与えるとします。a' の確率は $P(a') = P(a_i) + P(a_j)$ です。

符号語 $\phi_n(a')$ の長さを l とすれば、今の符号化関数の定義から $\phi_{n+1}(a_i), \phi_{n+1}(a_j)$ の長さは $l+1$ なので、平均符号長は

$$< l(\phi_{n+1}) >= (l+1)P(a_i) + (l+1)P(a_j) + O = (l+1)(P(a_i) + P(a_j)) + O$$

O は他の符号語での和です。ϕ_n での平均符号長 $< l(\phi_n) >$ は $lP(a') + O$ なので、これらの差を取ると

$$\begin{aligned} < l(\phi_{n+1}) > - < l(\phi_n) >&= (l+1)(P(a_i) + P(a_j)) - lP(a') \\ &= (l+1)(P(a_i) + P(a_j)) - l(P(a_i) + P(a_j)) \\ &= P(a_i) + P(a_j) \\ &= P(a') \end{aligned} \tag{7}$$

となります。

今度は $n+1$ から n を作ることを想定して別の符号化関数 Φ_n, Φ_{n+1} を定義します。Φ_{n+1} を $(\mathcal{A}_{n+1}, \boldsymbol{p}_{n+1})$ での最適な prefix-free の符号化関数と定義し、これによる符号語を $c_1, c_2, \ldots, c_n, c_{n+1}$ とします。最適なので (ii) から、最も長い符号語として同じ文字列 s を含むものが 2 個あり、それらを $c_n = s0,\ c_{n+1} = s1$ とし、c_n, c_{n+1} の長さを l_n とします。(i) から最も長い符号語は最も確率が低いので、$(\mathcal{A}_n, \boldsymbol{p}_n)$ での a' に対応する符号語を c_n, c_{n+1} から c' と与えて

$$(\mathcal{A}_n, \boldsymbol{p}_n) :\ c_1, c_2, \ldots, c_{n-1}, c' \quad (P(a') = P(a_n) + P(a_{n+1}))$$

c' の長さは $l_n - 1$ です。このときの prefix-free の符号化関数を Φ_n とします。

これらから、それぞれの平均符号長は

$$\begin{aligned} < l(\Phi_{n+1}) >&= l_1P(a_1) + l_2P(a_2) + \cdots + l_nP(a_n) + l_nP(a_{n+1}) \\ < l(\Phi_n) >&= l_1P(a_1) + l_2P(a_2) + \cdots + l_{n-1}P(a_{n-1}) + (l_n - 1)P(a') \end{aligned}$$

これらの差は

$$< l(\Phi_{n+1}) > - < l(\Phi_n) >= l_n(P(a_n) + P(a_{n+1})) - (l_n - 1)P(a') = P(a')$$

Φ_{n+1} は最適としているので、$(\mathcal{A}_{n+1}, \boldsymbol{p}_{n+1})$ での最適でない ϕ_{n+1} による平均符号長とは

$$< l(\Phi_{n+1}) > < < l(\phi_{n+1}) >$$

これと (7) を使えば

$$< l(\Phi_n) >=< l(\Phi_{n+1}) > -P(a') \ < < l(\phi_{n+1}) > -P(a') =< l(\phi_n) >$$

$< l(\Phi_n) > < < l(\phi_n) >$ から、ϕ_n は $(\mathcal{A}_n, p)$ で最適ではないです。というわけで、ϕ_{n+1} が最適でないなら ϕ_n も最適でないとなります。そして、これの対偶から ϕ_n が最適なら ϕ_{n+1} は最適となるので、ハフマン符号化の手順で最適な符号を与えられることになります。

・補足

底を e とする自然対数を ln と書きます。自然対数の微分は

$$\frac{d}{dx}\ln x = \frac{1}{x}$$

なので

$$f(x) = x - 1 - \ln x \ , \ \frac{d}{dx}f(x) = 1 - \frac{1}{x} \quad (x > 0)$$

とします。これの微分は

$$\frac{d}{dx}f(x) = 1 - \frac{1}{x} \ , \ \frac{d^2}{dx^2}f(x) = \frac{1}{x^2}$$

なので、$f(x)$ は $x = 1$ で極小値 $f(1) = 0$ を持ちます。このため、$x > 0$ に対して $f(x) \geq 0$ ($x = 1$ のとき等号) なので

$$x - 1 - \ln x \geq 0$$

$$x - 1 \geq \ln x \tag{8}$$

となります。これは覚えておくと便利です。

x_i, y_i $(i = 1, 2, \ldots, n)$ を

$$\sum_{i=1}^{n} x_i = 1 \ , \ \sum_{i=1}^{n} y_i = 1$$

として、(8) での x を y_i/x_i に置き換えて

$$\ln \frac{y_i}{x_i} \leq \frac{y_i}{x_i} - 1$$

$$x_i \ln \frac{y_i}{x_i} \leq y_i - x_i$$

と変形して和を取ると

$$\sum_{i=1}^{n} x_i \ln \frac{y_i}{x_i} \leq \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i = 0$$

左辺を変形させれば

$$\sum_{i=1}^{n} x_i \ln \frac{y_i}{x_i} = \sum_{i=1}^{n} x_i \ln \frac{1}{x_i} - \sum_{i=1}^{n} x_i \ln \frac{1}{y_i}$$

なので

$$\sum_{i=1}^{n} x_i \ln \frac{1}{x_i} \leq \sum_{i=1}^{n} x_i \ln \frac{1}{y_i}$$

底の変換

$$\log_a x = \log_a b \log_b x$$

を使えば

$$\sum_{i=1}^{n} x_i \log_b \frac{y_i}{x_i} \leq 0$$

$$\sum_{i=1}^{n} x_i \log_b \frac{1}{x_i} \leq \sum_{i=1}^{n} x_i \log_b \frac{1}{y_i}$$

となります。また、(8) では $x = 1$ のときに等号になるので、これが等号になるのは $x_i = y_i$ のときです。