S&M

Split and Merge Compression Algorithm

By

Abdullah Hashim

S&M algorithm: The Norms

S&M algorithm: The Norms
The Set-Norm: $H_{Set}(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

The Entropy: Sets have equiprobable state probability, then:

$$H_{set}(n) = \sum_{i=1}^{|S|} [p_{s_i}(n) \log_2(1/p_{s_i})] = L_{average} = \text{Average set codeword length.}$$

Optimum $H_{Set}(n)$ is equal to the average codeword of $s_i = \log_2(|S|)$

Expediency: The algorithm is said to be ***absolutely expedient*** if:

$$E\,[\,H_{Set}(n+1)\,] > E\,[\,H_{Set}(n)\,].$$

Optimality: The algorithm is said to be ***optimal*** if:

$\lim_{n\to\infty} E\,[H_{Set}(n)] = \log_2(|S|)$ and said to be ***e-optimal*** if: $\lim_{n\to\infty} E\,[H_{Set}(n)] = \log_2(|S|) + e$.

Where e is arbitrarily small positive number.

S&M algorithm: The Norms
The Set-Norm: $Q_{set}(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

Sets ***MSE vector***: Sets have equiprobable state probability, then:

$$\varepsilon(n) = \frac{1}{|S|}\sum_{i=1}^{|S|}\left(p_{s_i}(n) - \frac{1}{|S|}\right)^2 = \frac{1}{|S|}\sum_{i=1}^{|S|} p_{s_i}(n)^2 - \frac{2}{|S|^2}\sum_{i=1}^{|S|} p_{s_i}(n) + \frac{1}{|S|^2}$$

When $\varepsilon(n) = 0$; $\sum_{i=1}^{N} (p_{s_i}(n) = \frac{1}{|S|}$

Expediency: The algorithm is said to be ***absolutely expedient*** if:

$$E[\,Q_{Set}(n+1)\,] < E[\,Q_{Set}(n)\,].$$

Optimality: The algorithm is said to be ***optimal*** if:

$\lim_{n\to\infty} E[Q_{Set}(n)] = 1/|S|$ and said to be ***e-optimal*** if: $\lim_{n\to\infty} E[Q_{Set}(n)] = 1/|S| + e$.

Where e is arbitrarily small positive number.

S&M algorithm: The Norms
Word-Norm: $H_{word}(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

<u>The Entropy</u>: Assume all words in the source dictionary Δ are equiprobable, then; all source words have equal codeword length of $\log_2(|\Phi|)$.

At interval n, The entropy $H_{word}(n)$ is given by the expression:

$$H_{word}(n) = \sum_{i=1}^{|\Phi|} [p[w_i(n)] \log_2\{1/p[w_i(n)]\}$$

= Average codeword length of all words in the dictionary D.

$= L_{average}\{w_i(n)\}$.

$H_{word}(n)$ has a minimum value of **0** when one word has a probability of **1** and all other **($|\Phi|$-1)** words have probability **0**, and a maximum value of $\log_2(|\Phi|)$,when all words of D, are equiprobable.

S&M algorithm: The Norms

The Word-Norm: $Q_{word}(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_{\Phi}\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_{\Phi}\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

Word ***MSE vector***: Assume the action is equiprobable random variable, then:

$$\varepsilon(n) = \frac{1}{|\Phi|}\sum_{i=1}^{|\Phi|}\left\{p[w_i(n)] - \frac{1}{|\Phi|}\right\}^2 = \frac{1}{|\Phi|}\sum_{i=1}^{|\Phi|}\{p[w_i(n)]\}^2 - \frac{1}{|\Phi|^2}$$

When $\varepsilon(n) = 0$; $\sum_{i=1}^{|\Phi|}\{p[w_i(n)]\}^2 = \frac{1}{|\Phi|}$

Expediency: The algorithm is said to be ***absolutely expedient*** if:

$$E[\,Q_{word}(n+1)\,] < E[\,Q_{word}(n)\,].$$

Optimality: The algorithm is said to be ***optimal*** if:

$\lim_{n\to\infty} E\,[Q_{word}(n)] = 1/|S|$ and said to be ***e-optimal*** if: $\lim_{n\to\infty} E\,[Q_{word}(n)] = 1/|\Phi| + e$.

Where e is arbitrarily small positive number.

S&M algorithm: The Strategy

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$\boldsymbol{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$\boldsymbol{F} = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$\boldsymbol{S} = \{s_1, s_2, \ldots, s_{|S|}\}$
$\boldsymbol{D} = \{w_1, w_2, \ldots, w_\Phi\}$
$\boldsymbol{P}(\boldsymbol{S})$ and $\boldsymbol{P}(\boldsymbol{D})$

Output:
(s_s, s_j & $s_k \in \boldsymbol{S}$)

Discrete time interval: ***n***

The Strategy:

At every interval n, **(n = 1, 2, ..., |F|)**, the element of set $s_k(n)$ is merged with element of set $s_j(n)$:

$$s_j(n+1) = s_j(n) \cup s_k(n) \text{ and}$$
$$p[s_j(n+1)] = p[s_j(n)] + p[s_k(n)].$$

The element of the split set $s_s(n)$ is divided into two sets (s_{s1}) and (s_{s2}) such that:

$$p(s_{s1}) = \mathbf{a} \cdot p(s_s) \text{ and } p(s_{s2}) = (1-\mathbf{a}) \cdot p(s_s).$$
$$s_s(n+1) = s_{s1} \text{ and } s_k(n+1) = s_{s2}.$$

$$0 \le \mathbf{a} \le 1$$

S&M algorithm: The Strategy
-Optimum value of a-

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

$Q_S(n) = \{\ldots\} + p_{s_s}{}^2 + p_{s_j}{}^2 + p_{s_k}{}^2$; Split p_{s_s} into $a\, p_{s_s}$ and $(1 - a)\, p_{s_s}$.

$$Q_S(n+1) = \{\ldots\} + a^2 p_{s_s}{}^2 + (1\text{-}a)^2 P_{s_s}{}^2 + (p_{s_j} + p_{s_k})^2$$
$$= \{\ldots\} + a^2 p_{s_s}{}^2 + p_{s_s}{}^2 - 2\,a\, p_{s_s}{}^2 + a^2 p_{s_s}{}^2 + p_{s_j}{}^2 + 2\, p_{s_j} p_{s_k} + p_{s_k}{}^2$$
$$= \{\ldots\} + p_{s_s}{}^2 + p_{s_j}{}^2 + p_{s_k}{}^2 + 2\, a^2 p_{s_s}{}^2 - 2\,a\, p_{s_s}{}^2 + 2\, p_{s_j} p_{s_k}$$

$$Q_S(n) - Q_S(n+1) = -2\, a^2 p_{s_s}{}^2 + 2\,a\, p_{s_s}{}^2 - 2\, p_{s_j} p_{s_k}$$

$$d\{ Q_S(n) - Q_S(n+1) \} / d(a) = -4\, a\, p_{s_s}{}^2 + 2\, p_{s_s}{}^2$$
$$= 0; \quad \text{when } a = ½.$$

$$d^2\{ Q_S(n) - Q_S(n+1) \} / d(a)^2 = -4\, p_{s_s}{}^2;$$ Maximum value, when $a = ½$.

Optimum value of splitting, when $a = ½$.

S&M algorithm: Set Norms

The Set-Norm: $H_S(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

<u>H_S Norm</u>: We have established that: $I_S(p_{max}) \leq H_S$, and

$$\lim_{n\to\infty} \sqrt{(2/3)} / |S| \leq Q_S(n) \leq p_{max} \leq 2 / |S|;$$ Let $|S| = 256$, then:

<u>Maximum value of H_S</u>:

$$\lim_{n\to\infty} p_{max} \leq \sqrt{(2/3)} / 256 = 0.816/256, \quad I_S = -\log_2(0.816/256) = 8.29,$$

Maximum: $\lim_{n\to\infty} H_S < 8.29$

<u>Minimum value of H_S</u>: $\lim_{n\to\infty} p_{max} \leq 2 / 256$, $I_S = -\log_2(2/256) = 7$,

Minimum: $\lim_{n\to\infty} H_S > 7$.

$$\boxed{\lim_{n\to\infty} 7 < H_S(n) < 8.2; \; e \to 0}$$

S&M algorithm: Set Norms

The Set-Norm: $Q_S(n)$

Input: Action (ω)

$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$

$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$

$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:

$S = \{s_1, s_2, \ldots, s_{|S|}\}$

$D = \{w_1, w_2, \ldots, w_\Phi\}$

$P(S)$ and $P(D)$

Output:

(s_s, s_j & $s_k \in S$)

Discrete time interval: n

$Q_S(n) = \{\ldots\} + p_{s_s}^2 + p_{s_j}^2 + p_{s_k}^2$; split set s_s into two half and merge s_j with s_k:

$$Q_S(n+1) = \{\ldots\} + \quad + 1/4 \,.\, p_{s_s}^2 + 1/4 \,.\, P_{s_s}^2 + \quad (p_{s_j} + p_{s_k})^2$$

$$Q_S(n+1) = \{\ldots\} + \quad + \quad 1/2 \,.\, p_{s_s}^2 \quad + \; p_{s_j}^2 + 2\, p_{s_j} p_{s_k} + p_{s_k}^2$$

$$Q_S(n) - Q_S(n+1) = p_{s_s}^2 + p_{s_j}^2 + p_{s_k}^2 \quad - \quad 1/2 \,.p_{s_s}^2 \quad - \; p_{s_j}^2 - 2\, p_{s_j} p_{s_k} - p_{s_k}^2$$

$$= \quad 1/2 \,.p_{s_s}^2 \quad - 2\, p_{s_j} p_{s_k}$$

For expediency, $E\,[Q_S(n) - Q_S(n+1)] \quad > \quad 0;$

then: $E\,(1/2 \,.\, p_{s_s}^2) \quad > \quad E\,(2\, p_{s_j} p_{s_k})$

$$E\,(1/4 \,.\, p_{s_s}^2) \quad > \quad E\,\{[1/(1-|s|)] \,.\, [1/(1-|s|)]\}$$

$$E\,(p_{s_s}^2) \quad > \quad E\,(4 / |s|^2)$$

$$\boxed{\text{As } n \to \infty,\ Q_S(n) \le p_{max} \to 2 / |S|}$$

S&M algorithm: Set Norms
The Set-Norm: $Q_S(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

For simplicity, assume, s_s, s_j and s_k are singleton sets. Let p_1, p_2 and p_3 equal to $p(w_1)$, $p(w_2)$ and $p(w_3)$ respectively, where $w_1 \in s_s$, $w_2 \in s_j$ and $w_3 \in s_k$. In the process of splitting, p_1 will doubled, while, p_2 and p_3 will be halved in the merging process.

$$Q_S(n) = \{\ldots\} + p_1^2 + p_2^2 + p_3^2$$

$$Q_S(n+1) = \{\ldots\} + (2 \,.\, p_1)^2 + (1/2 \,.\, p_2 + 1/2 \,.\, p_3)^2$$

$$= \{\ldots\} + 4\, p_1^2 + 1/4\, p_2^2 + 2/4\, p_2 p_3 + 1/4\, p_3^2$$

$$Q_S(n+1) - Q_S(n) = + 3\, p_1^2 - 3/4\, p_3^2 - 2/4\, p_2 p_3 - 3/4\, p_3^2$$

For expediency, $E\,[Q_S(n+1)) - Q_S(n] > 0;$

$$\text{then: } E\,(3\, p_1^2) > E\,(3/4\, p_2^2) + E\,(2/4\, p_2 p_3) + E\,(3/4\, p_3^2)$$

$$E\,(3\, p_1^2) > E\,\{[3/4(1-|S|)^2] + [2/4(1-|S|)^2] + [3/4(1-|S|)^2]\}$$

$$E\,(p_1^2) > E\,(2/3\,|S|^2)$$

$$\text{As } n \to \infty,\ 2/|S| \le Q_S(n) \le p_{max} \le \sqrt{(2/3)}/|S|$$

S&M algorithm: Dictionary Norms
The Word-Norm: $H_W(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
(s_s, s_j & $s_k \in S$)

Discrete time interval: n

$\underline{H_W \text{ Norm}}$: We have established that: $I_D(p_{max}) \leq H_D$, and

$$\lim_{n\to\infty} \sqrt{(1/2)} / |\Phi| \leq p_{max} \text{ ; Let } |\Phi| = 256 \text{, then:}$$

Maximum value of H_W:

$$\lim_{n\to\infty} p_{max} \leq \sqrt{(1/2)} / 256 = 0.707/256,\ I_S = -\log_2(0.707/256) = 8.5,$$

$$\text{Maximum: } \lim_{n\to\infty} H_W < 8.5$$

Minimum value of H_W: $\lim_{n\to\infty} p_{max} \leq 2/256,\ I_S = -\log_2(2/256) = 7,$

$$\text{Minimum: } \lim_{n\to\infty} H_S > 7.$$

$$\boxed{\lim_{n\to\infty} 7 < H_W(n) < 8.5;\ e \to 0}$$

S&M algorithm: Dictionary Norms
The Word-Norm: $Q_W(n)$

Input: Action (ω)
$\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$
$A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$
$F = (c_1, c_2, \ldots, c_{|F|})$

Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Output:
$(s_s, s_j \;\&\; s_k \in S)$

Discrete time interval: n

For simplicity, assume, s_s, s_j and s_k are singleton sets. Let p_1, p_2 and p_3 equal to $p(w_1)$, $p(w_2)$ and $p(w_3)$ respectively, where $w_1 \in s_s$, $w_2 \in s_j$ and $w_3 \in s_k$. In the process of splitting, p_1 will doubled, while, p_2 and p_3 will be halved in the merging process.

$$Q_W(n) = \{\ldots\} + p_1^2 + p_2^2 + p_3^2$$
$$Q_W(n+1) = \{\ldots\} + (2 \,.\, p_1)^2 + (1/2 \,.\, P_2)^2 + (1/2 \,.\, p_3)^2$$
$$= \{\ldots\} + 4\, p_1^2 + 1/4\, p_2^2 + 1/4\, p_3^2$$
$$Q_W(n+1) - Q_W(n) = + 3\, p_1^2 - 3/4\, p_3^2 - 3/4\, p_3^2$$

For expediency, $E\,[Q_W(n+1)) - Q_D(n] > 0;$

$$\text{then: } E\,(3\, p_1^2) > E\,(3/4\, p_2^2) + E\,(3/4\, p_3^2)$$
$$E\,(3\, p_1^2) > E\,\{[3/4(1-|\Phi|)^2] + [3/4(1-|\Phi|)^2]\}$$
$$E\,(p_1^2) > E\,(1/2\,|\Phi|^2)$$

As $n \to \infty$, $Q_W(n) \le p_{max} \to \sqrt{(1/2)} / |\Phi|$