S&M

Split and Merge Compression Algorithm

By

Abdullah Hashim

S&M algorithm: The Concept

S&M algorithm: The Concept
-Action-

Input: Action (ω)

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

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

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

Discrete time interval: n

Action (ω): ω is random variable of a stationary environment Θ over a language with alphabet set (A); $A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$, N is the number of characters in the alphabet. The i^{th} action (ω_i) is a string, (word), in the source dictionary (Δ); $\Delta = \{\omega_1, \omega_2, \ldots, \omega_\Phi\}$, Φ is the file size of the dictionary ($|\Delta|$); i.e. the number of words in Δ. The dictionary must contains at least all the characters in alphabet A; (Min ($|\Delta|$) = N). The prior probabilities distribution and the statistical parameters of the words in Δ are not known explicitly.
The input source file (F) contains a sequence of symbols, (characters) of the alphabet A. At interval n, an input string of source symbols is matched with the longest string in the dictionary ($\omega(n)$). $\omega(n)$ is known as the n^{th} action.

S&M algorithm: The Concept
-Environment-

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

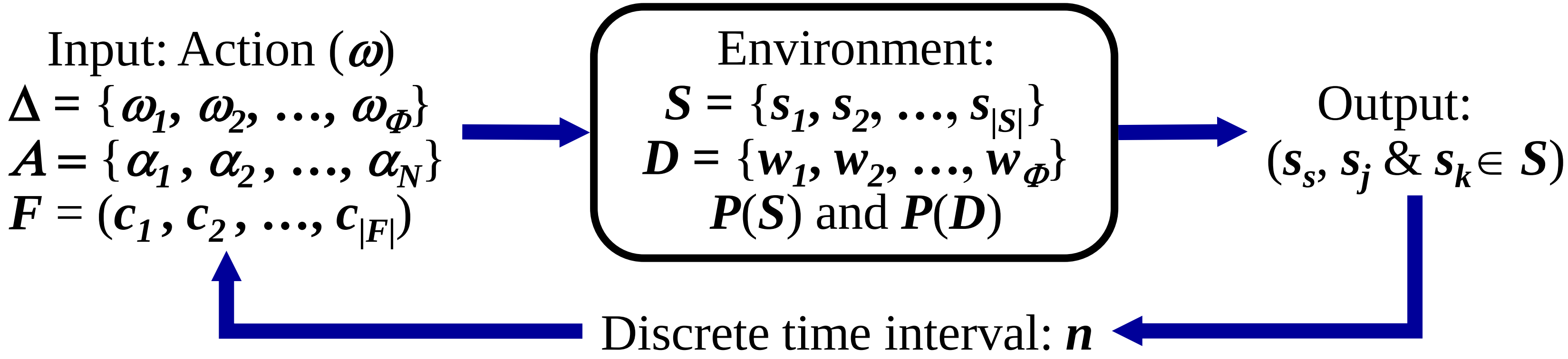
Environment:
$S = \{s_1, s_2, \ldots, s_{|S|}\}$
$D = \{w_1, w_2, \ldots, w_\Phi\}$
$P(S)$ and $P(D)$

Discrete time interval: n

Environment (S, D): Set S is a set of $|S|$ mutually exclusive sub sets, $s_1, s_2, .., s_{|S|}$. Each sub set contains a number of unique words (w_i) of a dictionary (D), (where, s_i is a subset of D, and $s_i = \{w_{i1}, w_{i2}, \ldots, |s_i|\}$. $P(S)$ is the set state probability vector, $P(S) = (\, p(s_1), p(s_2), \ldots, p(s_{|S|})\,) \mid p(s_i) = \sum_{j=1}^{|S_i|} p(w_{ij})$. $P(D)$ is the dictionary state probability vector $P(D) = (\, p(w_1), p(w_2), \ldots, p(w_\Phi)\,)$.

The dictionary $D = \Delta$, however, the state probability vector of D is updated by a pre-defined updating scheme. Since the action $\omega_i = w_i \mid \omega(n) = \omega_i$ and $w(n) = w_i$, $\omega_i \in \Delta$ and $w_i \in D$, the environment has no reward-penalty response.

S&M algorithm: The Concept
-Output-



Output (s_s, s_j, s_k): For every action ($\omega(n)$), the environment respond with three duple output ((s_s, s_j and $s_k \in S$). The first is a set ($s_s \in S$), called (***the split set***): The split set is that set contains the word, which, matches the action $\omega(n)$.

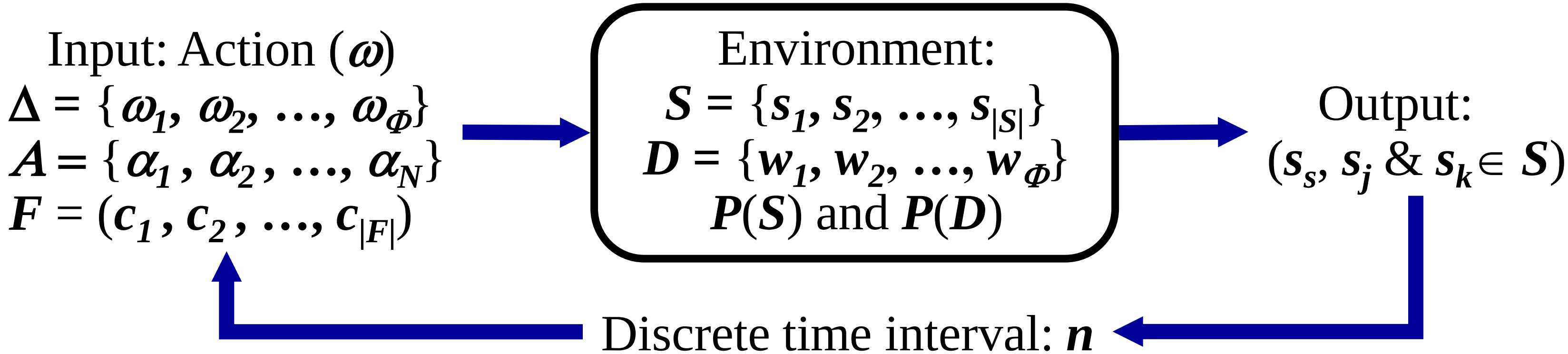
$$\text{If } w_s = \omega(n) \mid \omega(n) = \omega_s, \text{ and } w_s \in s(n) \mid s(n) = s_s.$$

The two, other sets (s_j & $s_k \in S$), called (***the merger sets***), are selected randomly, from the $|S|$ subsets of S, such that:

$$j < k, \text{ and } j \neq k \neq s.$$

S&M algorithm: The Concept

-Transition-



<u>Transition **($P(S)$, $P(D)$)**</u>: The state, set probability vectors **($P(S)$)** and the state dictionary probability vector **($P(D)$)** are updated by a pre-defined updating scheme. The scheme should ensure expediency and asymptotic convergence of:

1) the set state probability vector $P(S)$ to $(1/|S|, 1/|S|, \ldots, 1/|S|)$. i. e. the set probability $\lim_{n\to\infty} P(s_i) = 1 / |S|$, for $i = 1, 2, \ldots, |S|$.

2) the dictionary state probability vector $P(D)$ to the real probability vector of the source dictionary Δ.

S&M algorithm: The Concept
The Action-Norms

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 & $s_k \in S$)

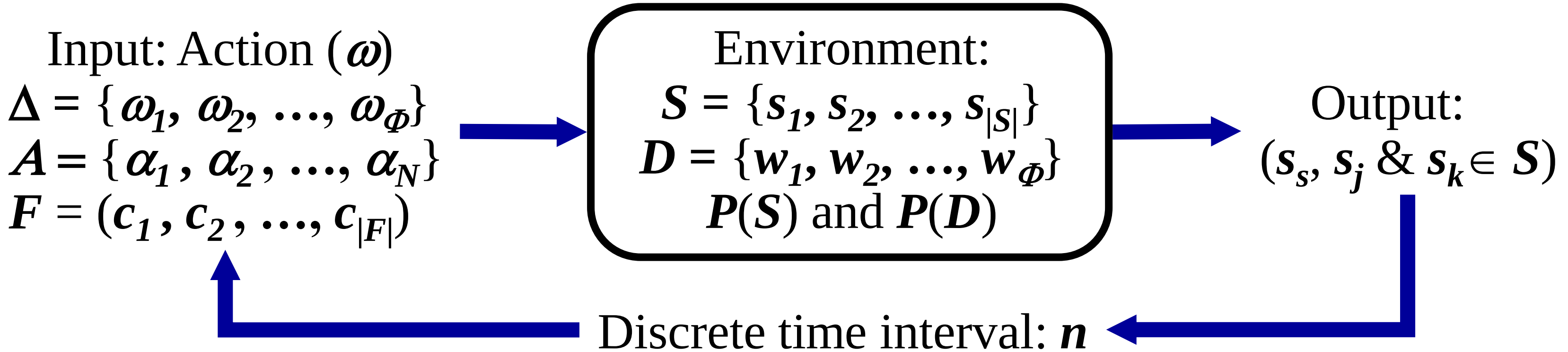
Discrete time interval: n

<u>Action Norms (H_{a}, Q_α)</u>: The norms used as a datum reference for the automaton learning capability is called the ***Action-Norms***, or (***α-Norms***) of the probability vectors $P(\omega)$. H_α is the action entropy and the ***MSE variable*** (Q_α), is the sum of, the square of, the word probabilities, for all words in Δ.

$$H_\alpha(n) = \sum_{i=1}^{|S|} \Big[\sum_{j=1}^{|s_i|} \big(-p(\omega_{ij}) \mid \omega_{ij} \in s_i\big)\Big] \cdot \log_2 \Big[\sum_{j=1}^{|s_i|} \big(p(\omega_{ii}) \mid \omega_{ij} \in s_i\big)\Big]$$

$$Q_\alpha(n) = \sum_{i=1}^{|S|} \sum_{j=1}^{|s_i|} \big[\big(p(\omega_{ij}) \mid \omega_{ij} \in s_i\big)\big]^2$$

S&M algorithm: The Concept
The Set-Norms



<u>Set Norms (H_S, Q_S)</u>: The norms used to test the expediency and optimality of the set state probability vectors $P(S)$ are called the ***Set-Norms***, or (***S-Norms***). H_S is the set real entropy, Q_S is the set real ***MSE Variable***, while the state set entroy equal to $\log_2(|S|)$ and the state set ***MSE variabl*** equal to $(1 / |S|)$.

$$H_S(n) = \sum_{i=1}^{|S|} \Big[\sum_{j=1}^{|s_i|}(-\, p(\omega_{ij}) \mid \omega_{ij} \in s_i) / FBL(s_i)\Big] \,.\, \log_2 \Big[\sum_{j=1}^{|s_i|}(\, p(\omega_{ii}) \mid \omega_{ij} \in s_i) / FBL(s_i)\Big]$$

$$Q_S(n) = \sum_{i=1}^{|S|} \Big[\sum_{j=1}^{|s_i|}(\, p\,(\omega_{ij}) \mid \omega_{ij} \in s_i) / FBL(s_i)\Big]^2$$

S&M algorithm: The Concept
The Word-Norms

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 Norms (H_W, Q_W): The norms used to test the expediency and optimality of the set state probability vectors $P(w)$ are called ***Word-Norms***, or (***W-Norms***). H_W is the state word entropy and the state word ***MSE variable*** (Q_W) is the sum of, the square of, word probabilities, for all words in D.

$$H_w(n) = \sum_{i=1}^{|S|} \left[\sum_{j=1}^{|s_i|} (-p_{inset}(w_{ij})|w_{ij} \in s_i)\; FBL(s_i)/|S|\right] \,.\, \log_2 \left[\sum_{j=1}^{|s_i|} (p_{inset}(w_{ii})|w_{ij} \in s_i)\; FBL(s_i)/|S|\right]$$

$$Q_W(n) = \sum_{i=1}^{|S|} \sum_{j=1}^{|s_i|} \left[(\, p_{inset}(w_{ij}) \mid w_{ij} \in s_i) \,.\, FBL(s_i) \,/\, |S|\right]^2$$