Quick Learning for Batch-Learning Self-Organizing Map

Makoto Kinouchi[1,3,4]
kinouchi@yz.yamagata-u.ac.jp

Naoshi Takada[1]
m01843@eie.yz.yamagata-u.ac.jp

Yoshihiro Kudo[1,3,4]
chemics@yz.yamagata-u.ac.jp

Toshimichi Ikemura[2,3]
tikemura@ddbj.nig.ac.jp

[1] Department of Bio-System Engineering, Faculty of Engineering, Yamagata University, 4-3-16 Jounan, Yonezawa, Yamagata 992-8510, Japan
[2] Department of Population Genetics, National Institute of Genetics, and the Graduate University for Advanced Studies, Mishima, Shizuoka, 441-8540, Japan
[3] ACT, JST (Japan Science and Technology)
[4] CREST, JST (Japan Science and Technology)

Keywords: Batch-Learning Self-Organizing Map, principal component analysis, quick learning

1 Introduction

We have developed the batch-learning algorithm for the self-organizing map (BLSOM, Batch-Learning Self-Organizing Map) where the learning does not depend on the input order, and applied it to some analyses based on the similarity [1, 2]. The size of the map must be increased, as the number of the input vector increases, and the learning time grows in proportion to the product of them. In this study, we propose the fast learning algorithm for the BLSOM based from the following two viewpoints. (1) The initial map is made based on the principal component analysis. (2) The input vector does not move on the map in before and after the updating of the weight vectors very much.

2 Methodology

2.1 Batch-Learning Self-Organizing Map

Initializing of the Weight Vectors: The initial weight vectors $w_{ij}^{(\mathrm{init})}$ are calculated based on the principal component analysis of the input vectors x_k as follows:

$$w_{ij}^{(\mathrm{init})} = \bar{x} + 5\sigma_1 \frac{i - I/2}{I} b_1 + 5\sigma_2 \frac{j - J/2}{J} b_2. \tag{1}$$

Here, $\bar{x}$ is the average vector of x_k, b_1 and b_2 are eigen vectors for the first and second principal components, and σ_1 and σ_2 are the standard deviations of the first and second principal components. The second dimension J is defined by $J = \lceil \sigma_2/\sigma_1 I \rceil$.

Classifying of the Input Vectors: The distances between the input vector x_k and the weight vector w_{ij} are calculated, and x_k is classified into the weight vector $w_{i'j'}$ with the smallest distance.

Updating of the Weight Vectors: The ijth weight vector is updated with

$$w_{ij}^{(\mathrm{new})} = w_{ij}^{(\mathrm{old})} + \alpha(t) \left\{ \frac{\sum_{x_k \in S_{ij}} x_k}{N_{ij}} - w_{ij}^{(\mathrm{old})} \right\}. \tag{2}$$

Here, the components of set S_{ij} are input vectors classified into $w_{i'j'}$ satisfying $i - \beta(t) \leq i' \leq i + \beta(t)$ and $j - \beta(t) \leq j' \leq j + \beta(t)$, and N_{ij} are the numbers of components of S_{ij}. The two parameters

Table 1: Learning time (Pentium 4, 2GHz) and total error.

Learning Method		Time	Error
Conventional		885 min.	3.29×10^5
Quick	$\beta'(t) = 2\beta(t) + 2$	161 min.	3.43×10^5
(proposed)	$\beta'(t) = 2\beta(t) + 25$	179 min.	3.32×10^5
	$\beta'(t) = \beta(t) + 25$	155 min.	3.31×10^5

$\alpha(t)(0 < \alpha(t) < 1)$ and $\beta(t)(0 \leq \beta(t))$ are learning coefficients for the tth cycle defined by

$$\alpha(t) = \max\left\{0.01, \alpha_{\text{init}}\left(1 - \frac{t}{\tau_\alpha}\right)\right\}, \quad \beta(t) = \left\lfloor \max\left\{0, \beta_{\text{init}}\left(1 - \frac{t}{\tau_\beta}\right)\right\}\right\rfloor \tag{3}$$

where, α_{init} and β_{init} are the initial values, and τ_α and τ_β are the time constants.

2.2 Quick Leaning for BLSOM

In the learning process, the classification of the input vectors requires much time, because the distances between the input vector and all of the weight vectors must be calculated in the conventional learning method. The proposed Quick Learning tries to reduce the classification time.

Classifying on the Initial Map: The weight vector $w_{i'j'}$ with the smallest distance on the initial map can be easily found by

$$i' = \left\lfloor \frac{(x - \bar{x})b_1}{\sigma_1} I + \frac{1}{2} \right\rfloor, \quad j' = \left\lfloor \frac{(x - \bar{x})b_2}{\sigma_2} J + \frac{1}{2} \right\rfloor, \tag{4}$$

because the initial map is made based on the principal component analysis.

Classifying on the Updated Map: The input vector is classified into $w_{i'j'}$ with the smallest distance satisfying $i^{(\text{prev})} - \beta'(t) \leq i' \leq i^{(\text{prev})} + \beta(t)$ and $j^{(\text{prev})} - \beta'(t) \leq j' \leq j^{(\text{prev})} + \beta(t)$. Here, $(i^{(\text{prev})}, j^{(\text{prev})})$ is the location of the weight vector in which x_k was classified before updating of the weight vectors.

3 Results and Discussion

Four SOMs were constructed with 59,122 genes from the 29 bacterial genes described in [2]. Table 1 shows the learning time and the total error after learning with 100 updates, represented as

$$E = \sum_k (x_k - w_{i'j'})^2. \tag{5}$$

The parameters were $I \times J = 200 \times 114$, $\alpha^{(\text{init})} = 0.5$, $\tau_\alpha = 100$, $\beta^{(\text{init})} = 50$, and $\tau_\beta = 50$. It can be observed that the learning time was able to be shortened to about 1/5 without dropping of the quality.

References

[1] Abe, T., Kanaya, S., Kinouchi, M., Kudo, Y., Mori, H., Matsuda, H., Carlos, D.C., and Ikemura, T., Gene classification method based on batch-learning SOM, *Genome Informatics*, 10:314–315, 1999.

[2] Kanaya, S., Kinouchi, M., Abe, T., Kudo, Y., Yamada, Y., Nishi, T., Mori, H., and Ikemura, T., Analysis of codon usage diversity of bacterial genes with a self-organizing map: characterization of horizontally transferred genes with emphasis on *E. coli* 157 genome, *Gene*, 276:89–99, 2001.