Time series prediction of the CATS benchmark using Fourier bandpass filters and competitive associative nets

Shuichi Kurogi [a,*] Miho Sawa [a] Takamasa Ueno [a]

[a] *Department of Control Engineering, Kyushu Institute of Technology, Tobata, Kitakyushu 804, Japan*

Abstract

An approach to time series prediction of the CATS benchmark (for Competition on Artificial Time Series) is presented, where we use Fourier bandpass filters and competitive associative nets (CAN2s). Since one of the difficulties of this prediction is that the given time series does not seem to involve sufficient number of data for obtaining the underlying dynamics of the time series to reproduce low frequency components, we apply the CAN2 only for learning high frequency components extracted via Fourier bandpass filters with trial parameter values of the upper and lower cutoff frequencies and the missing last value of the given time series. Supposing that the optimal values among the trial values will give the best prediction performance for high frequency components, we can identify such optimal values via a certain reasonable validation method, with which we predict the missing high frequency components, and then we obtain the missing data to be predicted via adding high and low frequency components.

Key words: Time series prediction; CATS benchmark; Competitive associative net; Fourier bandpass filter

1 Introduction

The time series of the CATS benchmark (for Competition on Artificial Time Series) is provided for the time series prediction competition held at 2004 International Joint Conference on Neural Networks (IJCNN'04) [1], where with 4,900 given data we should predict 100 missing data consisting of five blocks of 20 data for the time 981-1,000, 1,981-2,000, 2,981-3,000, 3,981-4,000, and 4,981-5,000. One of

* Corresponding author.
Email address: kuro@cntl.kyutech.ac.jp (Shuichi Kurogi).

the difficulties of this prediction is that the given time series does not seem to involve sufficient number of data for obtaining the underlying dynamics to reproduce low frequency components of the data because we cannot see any periodicity for linear dynamics and any boundary of attractor for nonlinear dynamics. In order to overcome this problem, we decide to use Fourier bandpass filters to separate low and high frequency components, where we use trial values of the upper and lower cutoff frequencies and the missing last value of the time series. Supposing that the optimal values among the trial values for separating low and high frequency components will also give the best performance in prediction of high frequency components predicted by a certain method, we can identify such optimal values via a certain validation method.

For predicting high frequency components, we use competitive associative net called CAN2 which we have developed to utilize conventional competitive and associative schemes [3,4] for learning to achieve efficient piecewise linear approximation of nonlinear functions. The CAN2 has been shown effective in function approximation, control and rainfall estimation problems [5]-[9]. In function approximation problems, online learning methods for the CAN2 are shown to achieve better performance than BPNs (back-propagation nets), RBFNs (radial basis function nets) and SVR (support vector regression)[6,8,9]. Further, a batch learning method has been developed recently and shown more effective than online learning methods for a finite number of given data [9]. In control problems [7], the ability of learning to achieve piecewise linear approximation is shown useful for learning multiple linear models of nonlinear and time-varying plants, and then for applying an appropriate linear model at each time to the conventional efficient linear controller such as GPC (generalized predictive controller). In the rainfall estimation contest held by the IEICE (Institute of Electronics, Information and Communication Engineers) of Japan[5], the CAN2 has achieved the second least mean square error (MSE) in estimating a huge number of rainfall data. At the CATS benchmark prediction competition [2] in IJCNN'04, the prediction method using the CAN2 has achieved the third least MSE for all prediction data among the 17 predictions selected from 24 submitted predictions[1].

In the following sections, we first describe the prediction method involving a summary of the CAN2 and the procedure to execute the CATS benchmark prediction using Fourier bandpass filters and the CAN2. Next, we summarize the results obtained on the initial values of the benchmark, analyze the results, and clarify the advantages and disadvantages of the current method. Further, we have improved our method so that it can be applied to the new data of the CATS benchmark. Finally, we show the conclusion.

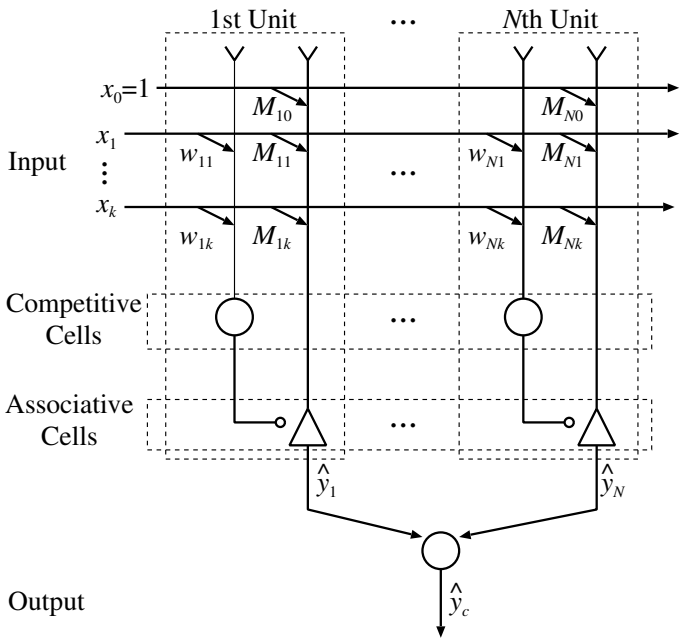


Fig. 1. Schematic diagram of the CAN2

2 Method for prediction

Here, we briefly show the CAN2 and its learning method (see [9] for details), and then describe the procedure to solve the CATS benchmark prediction.

2.1 The CAN2 for time series prediction

Let us suppose a time series $y(t)$ for discrete time $t = 1, 2, \cdots$ is generated as an output of a discrete dynamical system with a k-dimensional vector input $\boldsymbol{x}(t) \triangleq (x_1(t), x_2(t), \cdots, x_k(t)) \triangleq (y(t-1), y(t-2), \cdots, y(t-k))^T \in \mathbb{R}^{k\times 1}$ as follows;

$$y(t) = f(\boldsymbol{x}(t)) + d(t), \tag{1}$$

where $d(t)$ is zero-mean noise with the variance σ_d^2. To keep the notation simple in the following, we sometimes drop the time t as $y = f(\boldsymbol{x}) + d$. A CAN2 has N units, and the ith unit has a weight vector $\boldsymbol{w}_i \triangleq (w_{i1}, \cdots, w_{ik})^T \in \mathbb{R}^{k\times 1}$ and an associative (row) vector $\boldsymbol{M}_i \triangleq (M_{i0}, M_{i1}, \cdots, M_{ik}) \in \mathbb{R}^{1\times(k+1)}$ for $i \in I = \{1, 2, \cdots, N\}$ (see Fig. 1). The CAN2 approximates the above function $y = f(\boldsymbol{x})$, or the dynamics without noise, by

$$\widehat{y} \triangleq \widehat{y}_c \triangleq \boldsymbol{M}_c \widetilde{\boldsymbol{x}}, \tag{2}$$

where $\widetilde{\boldsymbol{x}} \triangleq (1, \boldsymbol{x}^T)^T \in \mathbb{R}^{(k+1)\times 1}$, and the cth unit has the weight vector $\boldsymbol{w}_c$ closest to the input vector $\boldsymbol{x}$, or

$$c \triangleq \underset{j\in I}{\text{argmin}} \|\boldsymbol{x} - \boldsymbol{w}_j\|. \tag{3}$$

The above function approximation partitions the input space $V = \mathbb{R}^k$ into the Voronoi (or Dirichlet) regions

$$V_i \triangleq \{\boldsymbol{x} \;\Big|\; i = \underset{j\in I}{\text{argmin}} \|\boldsymbol{x} - \boldsymbol{w}_j\|\}, \tag{4}$$

for $i \in I$, and performs piecewise linear approximation of the function $y = f(\boldsymbol{x})$. This function approximation also indicates that the single-step ahead prediction $y = y(t)$ is done with the previous output $y(t-j)$ for $j = 1, 2, \cdots$ which are supposed to be known or given. Further, multi-step prediction $y(t+i)$ for $i = 0, 1, 2, \cdots$ can be obtained recursively (see below for details).

2.2 Batch learning method for the CAN2

We use a batch learning method introduced in [9] for this prediction. Here, note that there are several parameter values to be tuned for a good performance, but the optimal values, except the number of units, for many applications seem almost the same as those used for function approximation problems [9]. Thus, we have not tuned the parameter values so much, and here we show the values in this section when introducing the parameters. The method is described as follows; for a given training data set $D \triangleq \{(\boldsymbol{x}_j, y_j = f(\boldsymbol{x}_j) + d_j) \mid j \in J\}$ for $J = \{1, 2, \cdots, n\}$, we modify $\boldsymbol{w}_i$ and $\boldsymbol{M}_i$ for $i \in I = \{1, 2, \cdots, N\}$ to minimize the mean square error (MSE) or the energy given by

$$E \triangleq \frac{1}{n} \sum_{i\in I} \sum_{\boldsymbol{x}\in X_i} \|e(\boldsymbol{x})\|^2 = \sum_{i\in I} E_i, \tag{5}$$

where $e(\boldsymbol{x}_j) \triangleq \widehat{y}_c - y_j \triangleq \boldsymbol{M}_c \widetilde{\boldsymbol{x}}_j - f(\boldsymbol{x}_j)$ is the prediction error, $X_i = X \cap V_i$ for $X = \{\boldsymbol{x}_j \mid j \in J\}$ is the training set of the input vectors in the Voronoi region V_i, and $E_i \triangleq (1/n) \sum_{\boldsymbol{x}\in X_i} \|e(\boldsymbol{x})\|^2$ is the energy of the region. To solve this nonlinear minimization problem, we apply iterations of a batch modification of $\boldsymbol{w}_i$ for all $i \in I$, and a batch modification of $\boldsymbol{M}_i$ for all $i \in I$, and then reinitialization, as follows.

Modification of weight vectors: Provided that $\boldsymbol{M}_i$ $(i \in I)$ are constant, we can optimize $\boldsymbol{w}_i$ via the following gradient method; let the boundary of a Voronoi region V_i and the adjacent V_l with the width W_θ (< 1; we have used 0.2 for the

competition) be

$$W_{il} \triangleq \left\{ \boldsymbol{x} \;\middle|\; \boldsymbol{x} \in X_i \cup X_l \quad \text{and} \quad \frac{\left|(2\boldsymbol{x} - \boldsymbol{w}_i - \boldsymbol{w}_l)^T(\boldsymbol{w}_i - \boldsymbol{w}_l)\right|}{\|\boldsymbol{w}_i - \boldsymbol{w}_l\|^2} \leq W_\theta \right\}. \quad (6)$$

When a training vector $\boldsymbol{x}$ is in W_{il} and moves from V_i to V_l (or from V_l to V_i) owing to the change of $\boldsymbol{w}_i$ by $\Delta\boldsymbol{w}_i$, the energy E increases by $(1/n)(e_i^2(\boldsymbol{x}) - e_l^2(\boldsymbol{x})) \times s$ where $s \triangleq \text{sign}(\Delta\boldsymbol{w}_i^T(\boldsymbol{x} - \boldsymbol{w}_i))$, while E does not change when $\boldsymbol{x} \in V_i$ and $s = 1$ or $\boldsymbol{x} \in V_l$ and $s = -1$. Thus, the increase of E is discontinuous, but it can be stochastically approximated by

$$\Delta E \simeq \frac{1}{2n}\Delta\boldsymbol{w}_i^T\boldsymbol{\xi}_i, \quad (7)$$

where

$$\boldsymbol{\xi}_i \triangleq \sum_{l \in A_i} \sum_{\boldsymbol{x} \in W_{il}} (e_i^2(\boldsymbol{x}) - e_l^2(\boldsymbol{x})) \frac{\boldsymbol{x} - \boldsymbol{w}_i}{||\boldsymbol{x} - \boldsymbol{w}_i||}, \quad (8)$$

and A_i is the index set of V_l adjacent to V_i. Here, note that $(\boldsymbol{x} - \boldsymbol{w}_i)/||\boldsymbol{x} - \boldsymbol{w}_i||$ indicates the direction of $\boldsymbol{\xi}_i$. Thus, in order to decrease E, we modify the weight vectors as $\Delta\boldsymbol{w}_i = -\gamma\boldsymbol{\xi}_i$, or

$$\boldsymbol{w}_i := \boldsymbol{w}_i - \gamma\boldsymbol{\xi}_i \quad (9)$$

for $i \in I$, where $:=$ indicates the substitution. We use the learning rate γ given by $\gamma \triangleq \gamma_0 d_x/d_\xi$, where $\gamma_0 (< 1$; we have used $0.001)$ is a positive constant, d_x is the maximum width between the elements x_{jl} of $\boldsymbol{x}_j$ as follows

$$d_x \triangleq \max_{l=1,\cdots,k} \left(\max_{\substack{i \in J \\ j \in J}} |x_{il} - x_{jl}| \right), \quad (10)$$

and d_ξ is the maximum value of the element ξ_{il} of $\boldsymbol{\xi}_i$ as follows,

$$d_\xi \triangleq \max_{l=1,\cdots,k} \max_{i \in I} \xi_{il}. \quad (11)$$

Thus, $\gamma = \gamma_0 d_x/d_\xi$ guarantees that the absolute value of the element of weight change, $|\Delta w_{ij}| = |\gamma\xi_{ij}|$, is less than the maximum span of the elements of input vectors, $d_x = \max_{l,i,j} |x_{il} - x_{jl}|$ multiplied by γ_0.

We have set the initial weight vectors $\boldsymbol{w}_i$ $(i \in I)$ to the vectors selected randomly from the training input vectors $\boldsymbol{x}_j$ $(j \in J)$ at only the first batch learning iteration.

Modification of associative vectors: Provided that the weight vectors $\boldsymbol{w}_i$ $(i \in I)$ are constant, the nonlinear problem of minimizing $E = \sum_{i \in I} E_i$ becomes a linear one to minimize

$$E_i = \frac{1}{n}\|\boldsymbol{M}_i\widetilde{\boldsymbol{X}}_i - \boldsymbol{Y}_i\|^2 \quad (12)$$

for each i, and the solution is given by $\boldsymbol{M}_i = \boldsymbol{Y}_i \widetilde{\boldsymbol{X}}_i^+$, where $\widetilde{\boldsymbol{X}}_i^+$ is the generalized inverse of the matrix $\widetilde{\boldsymbol{X}}_i \in \mathbb{R}^{(k+1)\times n_i}$ which consists of $\widetilde{\boldsymbol{x}} = (1, \boldsymbol{x}^T)^T$ for all $\boldsymbol{x} \in X_i$, and $\boldsymbol{Y}_i \in \mathbb{R}^{1\times n_i}$ is the matrix consisting of $y = f(\boldsymbol{x}) + d$ for all $\boldsymbol{x} \in X_i$. In order to avoid the situation where n_i or the number of the vectors in X_i is so small that the approximation error may become large, we do not use the unit with $n_i = 0$ for modifying $\boldsymbol{w}_i$ and $\boldsymbol{M}_i$ and calculating the output of the CAN2 until the reinitialization (see below) is triggered. Further, for X_i with $n_i \geq 1$, we compensate training vectors near V_i up to a certain number n_θ (we have used $n_\theta = 3$), or search the training vectors in

$$\begin{aligned} \Delta X_i \triangleq \{\boldsymbol{x}_j \,|\boldsymbol{x}_j \in X\backslash X_i, \|\boldsymbol{x}_j - \boldsymbol{w}_i\| \leq \|\boldsymbol{x}_l - \boldsymbol{w}_i\| \text{ for } \boldsymbol{x}_l \in X\backslash\Delta X_i, \\ |\Delta X_i| = n_\theta - n_i\}, \end{aligned} \tag{13}$$

where $|\Delta X_i|$ is the number of the vectors in ΔX_i, and then set $X_i := X_i \cup \Delta X_i$ for calculating $\boldsymbol{M}_i = \boldsymbol{Y}_i \widetilde{\boldsymbol{X}}_i^+$. Further, for stable learning performance with modifying $\boldsymbol{w}_i$ $(i \in I)$, we do not directly calculate $\boldsymbol{M}_i = \boldsymbol{Y}_i \widetilde{\boldsymbol{X}}_i^+$, but apply the following RLS (recursive least square) method,

$$\boldsymbol{M}_i := \boldsymbol{M}_i + \frac{(y - \boldsymbol{M}_i\widetilde{\boldsymbol{x}})\,\widetilde{\boldsymbol{x}}^T\boldsymbol{\Psi}_i}{1 + \widetilde{\boldsymbol{x}}^T\boldsymbol{\Psi}_i\widetilde{\boldsymbol{x}}}, \tag{14}$$

where $\boldsymbol{\Psi}_i \in \mathbb{R}^{(k+1)\times(k+1)}$ is also updated as

$$\boldsymbol{\Psi}_i := \boldsymbol{\Psi}_i - \frac{\boldsymbol{\Psi}_i\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^T\boldsymbol{\Psi}_i}{1 + \widetilde{\boldsymbol{x}}^T\boldsymbol{\Psi}_i\widetilde{\boldsymbol{x}}}, \tag{15}$$

and the above two updates are applied for all $\boldsymbol{x} \in X_i$ and the corresponding $y \in Y_i$ for all $i \in I$ once at each iteration. Further, at only the first iteration, we have set the initial values to the vectors as $\boldsymbol{M}_i = \boldsymbol{O}$ and $\boldsymbol{\Psi}_i = \boldsymbol{I}/\epsilon$, respectively, where $\boldsymbol{O}$ is the null (zero) vector, $\boldsymbol{I}$ is the unit matrix, and ϵ is a small constant (we have used $\epsilon := 10^{-4}$ for the competition).

Reinitialization: In order to avoid the local minima problem owing to the gradient method for modifying $\boldsymbol{w}_i$ shown above, the condition called asymptotic optimality, where a large number of weight vectors are supposed, has been derived and the online learning methods embedding this condition are shown effective [6,8]. Here, we also embed it to the present batch learning, as follows; first suppose there are many input data and weight vectors, and let the energy be given by

$$E = \sum_{i\in I}\int_{V_i} \|e_i(\boldsymbol{x})\|^2 p(\boldsymbol{x})d\boldsymbol{x} = \sum_{i\in I} E_i, \tag{16}$$

where $p(\boldsymbol{x})$ is the probability density function. Further, suppose the area of V_i is small and $p(\boldsymbol{x})$ is approximated by a constant p_i in each V_i, and $f(\boldsymbol{x})$ is the

function of class C^2, then we have

$$E = \sum_{i \in I} E_i = \sum_{i \in I} (C_i p_i v_i^{1+4/k} + \sigma_2^2 p_i v_i) \geq \quad N^{-4/k} \|C(\boldsymbol{x}) p(\boldsymbol{x})\|_{\frac{1}{1+4/k}} + \sigma_d^2, \tag{17}$$

where $C_i \triangleq C(\boldsymbol{w}_i)$ is called quantization coefficient which represents the complexity of $f(\boldsymbol{x})$ at $\boldsymbol{x} = \boldsymbol{w}_i$, and $\|g(\boldsymbol{x})\|_\alpha \triangleq (\int_V |g(\boldsymbol{x})|^\alpha d\boldsymbol{x})^{1/\alpha}$, $\|C(\boldsymbol{x}) p(\boldsymbol{x})\|_{\frac{1}{1+4/k}}$ is constant for given $f(\boldsymbol{x})$ and $p(\boldsymbol{x})$. Further, the right hand side of Eq.(17) is the minimum of E and the equality holds iff

$$\alpha_i \triangleq C_i p_i v_i^{1+4/k} = \text{constant} \quad \text{ for all } i \in I. \tag{18}$$

This equation represents the condition of asymptotic optimality, which can be used as follows. From Eq.(5), Eq.(17) and Eq.(18), the total square error S_i of the ith unit is given by

$$S_i \triangleq \sum_{\boldsymbol{x} \in X_i} \|e(\boldsymbol{x})\|^2 \simeq n\alpha_i + \sigma_d^2 n_i. \tag{19}$$

When there is a region V_i where $f(\boldsymbol{x})$ is approximated by a linear function, C_i and α_i are 0, and we can estimate the variance of the noise d_i by

$$\widehat{\sigma}_d^2 := \min\{S_i/n_i \mid i \in I \text{ and } n_i \geq \theta_U\}, \tag{20}$$

where θ_U is a constant larger than the dimension k of $\boldsymbol{x}$, and we have used $\theta_U = k + 5$ for the competition. Then, from Eq.(19) and Eq.(20), we can estimate α_i as

$$\widehat{\alpha}_i := \frac{S_i - \widehat{\sigma}_d^2 n_i}{n}. \tag{21}$$

In order to decide whether α_i $(i \in I)$ satisfy the asymptotic optimality of Eq.(18) or not, we use the following condition (see [10–12]);

$$\frac{\widehat{\alpha}_i}{\langle \widehat{\alpha}_i \rangle} \geq \theta_\alpha \quad \text{and} \quad \frac{H}{\ln(N)} \leq \theta_H, \tag{22}$$

where θ_α $(> 1;$ we have used $5)$ and θ_H $(< 1;$ we have used $0.9)$ are positive constants, $\langle \widehat{\alpha}_i \rangle$ is the mean of $\widehat{\alpha}_i$, and H is the entropy given by

$$H \triangleq -\sum_{i \in I} \frac{\alpha_i}{\sum_{j \in J} \alpha_j} \ln \left(\frac{\alpha_i}{\sum_{j \in I} \alpha_j} \right). \tag{23}$$

When the above condition in Eq.(22) is fulfilled, we reinitialize the $s(j)$th unit that has the jth smallest $\widehat{\alpha}_i$ for all $i \in I$ (the unit with $n_i = 0$ as described above is supposed to have the smallest $\widehat{\alpha}_i = 0$), and move it near to the $b(j)$th unit that

satisfies the former inequality in Eq.(22) and have the jth biggest $\widehat{\alpha}_i$ for all $i \in I$, as follows

$$\boldsymbol{w}_{s(j)} := \boldsymbol{w}_{b(j)} + \theta_r(\boldsymbol{x}_{c(b(j))} - \boldsymbol{w}_{b(j)}), \tag{24}$$

$$\boldsymbol{M}_{s(j)} := \boldsymbol{M}_{b(j)}, \tag{25}$$

where $\boldsymbol{x}_{c(b(j))}$ is the training vector nearest to $\boldsymbol{w}_{b(j)}$. We use the value $\theta_r = 1.9$, which guarantees that the region $V_{s(j)}$ of the new $\boldsymbol{w}_{s(j)}$ involves at least one training vector $\boldsymbol{x}_{c(b(j))}$.

2.3 Outline of the CATS benchmark prediction using the CAN2

For describing the time series prediction of the CATS benchmark, we denote the time series by $r(t)$ $(t = 1, 2, \cdots)$. For the competition, $r(t)$ for t in

$$T^{\text{given}} \triangleq \bigcup_{b=0}^{4} T_b^{\text{given}} \triangleq \bigcup_{b=0}^{4} \{1000b + j \mid j = 1, 2, \cdots, 980\}, \tag{26}$$

is given, and $y(t)$ for t in

$$T^{\text{pred}} \triangleq \bigcup_{b=0}^{4} T_b^{\text{pred}} \triangleq \bigcup_{b=0}^{4} \{1000b + j \mid j = 981, \cdots, 1000\}. \tag{27}$$

should be predicted.

In order to extract the data for learning via the CAN2, we separate the given signal $r(t)$ into two signals $y(t)$ and $r_c(t) \triangleq r(t) - y(t)$, where $y(t)$ is obtained via the bandpass filter described below and used for training and prediction via the CAN2. After training the CAN2 with $y(t)$ for T^{given}, the CAN2 performs multistep prediction of $y(t)$ for T^{pred} as

$$\widehat{y}(t) = \boldsymbol{M}_{c(t)}\widetilde{\boldsymbol{z}}(t), \tag{28}$$

where $\widetilde{\boldsymbol{z}}(t) \triangleq (1, \boldsymbol{z}^T)^T \triangleq (1, z(t-1), z(t-2), \cdots, z(t-k))^T$ whose elements are

$$z(t-j) \triangleq \begin{cases} y(t-j) & \text{for } t-j \in T^{\text{given}}, \\ \widehat{y}(t-j) & \text{for } t-j \in T^{\text{pred}}, \end{cases} \tag{29}$$

and $c(t)$ is obtained from Eq.(3) whose $\boldsymbol{x}$ is replaced by $\boldsymbol{z}(t)$. The above equation is applied consecutively from $t = \min T_b^{\text{pred}}$ to $t = \max T_b^{\text{pred}}$ for each block. With $\widehat{y}(t)$ for T^{pred}, we derive the designated prediction $\widehat{r}(t)$ of $r(t)$ by means of the following equation;

$$\widehat{r}(t) := \widehat{y}(t) + r_c(t). \tag{30}$$

2.4 Detailed procedure to solve the problem

We here describe the original procedure for the prediction submitted to the competition; we have several trial values to be optimized for obtaining a good result, and the actual steps to solve this problem we have employed are as follows.

Step 0: Linear interpolation

In order to apply the bandpass filter shown below, we need $r(t)$ for T^{pred}. So, we first derive a linear prediction $\widehat{r}^{(0)}(t)$ of $r(t)$, i.e. we obtain $\widehat{r}^{(0)}(t)$ for T_b^{pred} via the following linear interpolation

$$\widehat{r}^{(0)}(t) := \frac{r(t_1) - r(t_0)}{t_1 - t_0}(t - t_0) + r(t_0) \tag{31}$$

where $t_0 = 1000b + 980$, $t_1 = 1000b + 1001$, $t_0 < t < t_1$, and we use trial values r_{last} for $r(5001) = r_{\text{last}}$ which will be optimized via the steps below. We denote the obtained $r(t)$ for all $t \in T$ by

$$r^{(m)}(t) \triangleq \begin{cases} r(t) & \text{for } t \in T^{\text{given}}, \\ \widehat{r}^{(m)}(t) & \text{for } t \in T^{\text{pred}}, \end{cases} \tag{32}$$

where $m = 0$ at this *Step 0*, and after setting $m := 1$ go to *Step 1*. Here, note that the following steps are iterated and the number of iterations is denoted by m, where $r^{(m-1)}(t)$ is used as the source signal $r(t)$ at the mth iteration.

Step 1: Bandpass Filtering

We apply the Fourier bandpass filter, or exactly the FFT (Fast Fourier Transform) bandpass filter, to $r(t)$ and obtain $y(t)$ as follows; (1) apply the FFT to $r(t)$, (2) remove high and low frequency components, (3) apply the IFFT (Inverse FFT) which derives $y(t)$, where for the fundamental period $L_0 = 8192 = 2^{13} (> 5000)$ we use trial lower and higher cutoff frequencies $f_l = 1/L_l$ and $f_h = 1/L_h$ (≤ 0.5), respectively, and select the best ones after evaluating the performance by means of the following steps.

Step 2: Validation of trial parameter values

In order to evaluate trial values of the parameters for a good prediction, we introduce a test period of time given by

$$T_{bPQ}^{\text{test}} \triangleq \{1000b + P + j \mid j = 1, 2, \cdots, Q\} \subset T_b^{\text{given}}, \tag{33}$$

which is used for the following prediction test. Namely, first we train the CAN2 with $y(t)$ for $T_{bPQ}^{\text{train}} \triangleq T^{\text{given}} \backslash T_{bPQ}^{\text{test}}$, and do multistep prediction $\widehat{r}(t)$ for T_{bPQ}^{test} via Eqs.(28), (29) and (30) whose T^{given} and T^{pred} are replaced by T_{bPQ}^{train} and T_{bPQ}^{test}, respectively. Then, we have the MSE of $\widehat{r}(t)$ as follows,

$$MSE_{bPQ}^{\text{test}} \triangleq \frac{1}{Q} \sum_{t \in T_{bPQ}^{\text{test}}} (\widehat{r}(t) - r(t))^2. \tag{34}$$

Further, we calculate the average $MSE_{b(P+l)Q}^{\text{test}}$ for $l = 0, 1, \cdots, M$ as

$$\left\langle MSE_{bPQ}^{\text{test}} \right\rangle_M \triangleq \frac{1}{M} \sum_{l=0}^{M-1} MSE_{b(P+l)Q}^{\text{test}}, \tag{35}$$

where we use $M = 20$ for a stable evaluation because the single MSE with $M = 1$ changes largely for the change of P or the starting point of the evaluation. Among all trial values of parameters for this mth iteration, we select the best values achieving

$$PI_b^{(m)} \triangleq \min \left\{ \left\langle MSE_{bPQ}^{\text{test}} \right\rangle_M \text{ for all trial values} \right\}, \tag{36}$$

If the condition given by

$$PI_b^{(m)} \geq PI_b^{(m-1)} \tag{37}$$

is fulfilled, we quit the iteration and decide that the best prediction has been achieved by $\widehat{r}^{(m-1)}(t)$. Otherwise, we calculate $\widehat{y}(t)$ and $\widehat{r}(t)$ through Eq.(28) and Eq.(30) with the best parameter values, set $\widehat{r}^{(m)}(t) := \widehat{r}(t)$ for T^{pred}, $m := m + 1$, and go to *Step 1*.

Note that the above process for optimizing parameter values is executed for each block.

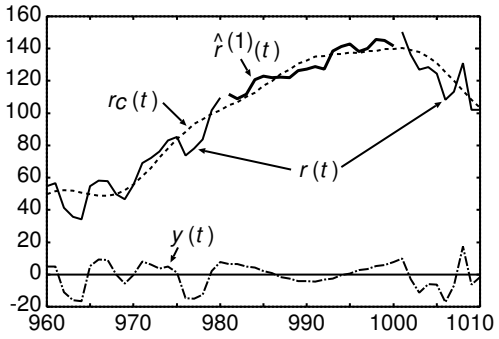


Fig. 2. Prediction $\widehat{r}^{(1)}(t)$ for T_0^{pred} after the first iteration, where the selected parameter values are as follows; embedding dimension $k = 17$, number of units $N = 22$, lower cutoff frequency $f_l = 551/8192$, higher cutoff frequency $f_h = 4011/8192$, the last value $r_{\text{last}} = -25$.

3 Results obtained on the initial values of the benchmark

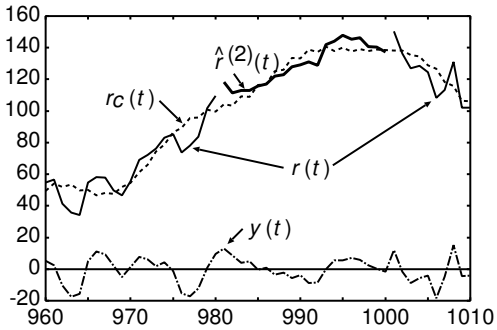
The results submitted to the competition are as follows; the parameter values selected and the prediction $\widehat{r}^{(1)}(t)$ for T_0^{pred} after the first iteration are shown in Fig. 2, where $PI_b^{(1)} = \left\langle MSE_{bPQ}^{\text{test}} \right\rangle_M = 43.16$ for $b = 0$ and we have used $P = 929$ because it has achieved the smallest $\left\langle MSE_{bPQ}^{\text{test}} \right\rangle_M$ around $P = 930$, and $Q = 20$ because the duration of the designated prediction period T_b^{pred} is also 20. Further, $M = 20$ for a stable evaluation as described above.

After $m = 2$ iterations, the performance index $PI_b^{(m)}$ for all $b = 0, 1, 2, 3, 4$ became smaller, therefore we ran the next iteration. After $m = 3$ iterations, $PI_b^{(m)}$ for $b = 1, 2, 4$ did not become smaller although that for $b = 0$ and 3 were reduced. Here, the latter result shows that $\widehat{r}^{(3)}(t)$ for T_0^{pred} and T_3^{pred} may be better than $\widehat{r}^{(2)}(t)$. On the other hand, $PI_b^{(m)}$ is affected by $\widehat{r}^{(m-1)}(t)$ of the bth block as well as other blocks, so that $\widehat{r}^{(2)}(t)$ for $b = 0$ and 3 had a possibility to provide the improvement of $\widehat{r}^{(3)}(t)$ for $b = 1, 2, 4$. Since the difference between $PI_b^{(2)}$ and $PI_b^{(3)}$ is not so big for all b, we decided to quit the iteration and submit the prediction $\widehat{r}^{(2)}$ to the competition. The prediction $\widehat{r}^{(2)}$ and the parameter values for each block obtained by the optimization procedure described above are shown in Fig. 3, and see the next section for an analysis of the result.

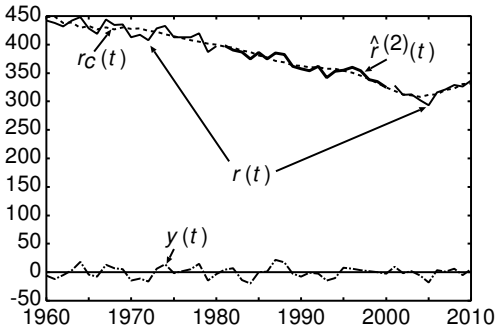
The predictions submitted to the competition have been ranked by the following MSE, or E_1, for all 100 prediction data,

$$E_1 = \frac{1}{100} \sum_{b=0}^{4} \sum_{t \in T_b^{\text{pred}}} (r(t) - \widehat{r}(t))^2. \tag{38}$$

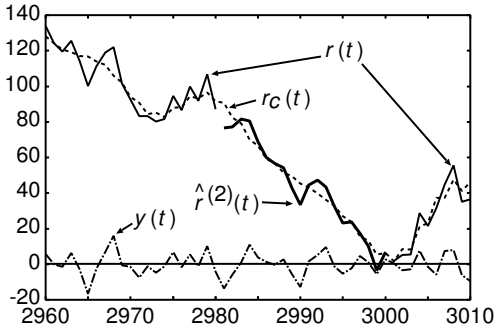
Among 17 predictions selected from 24 submitted ones (see [1] for details), our



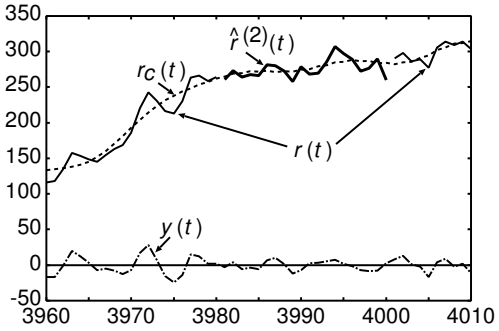
(a) $\widehat{r}^{(2)}(t)$ for T_0^{pred}



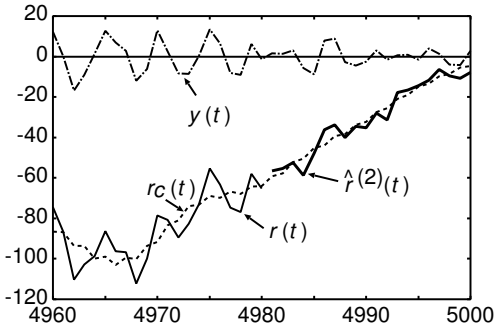
(b) $\widehat{r}^{(2)}(t)$ for T_1^{pred}



(c) $\widehat{r}^{(2)}(t)$ for T_2^{pred}



(d) $\widehat{r}^{(2)}(t)$ for T_3^{pred}



(e) $\widehat{r}^{(2)}(t)$ for T_4^{pred}

Fig. 3. Submitted prediction $\widehat{r}^{(2)}(t)$ for all T_b^{pred} $(b = 0, 1, 2, 3, 4)$ with optimized parameter values as follows; (a) $k = 17$, $N = 25$, $f_l = 550/8192$, $f_h = 3916/8192$, $r_{\text{last}} = -25$, (b) $k = 11$, $N = 43$, $f_l = 793/8192$, $f_h = 4037/8192$, $r_{\text{last}} = 80$, (c) $k = 11$, $N = 43$, $f_l = 793/8192$, $f_h = 4037/8192$, $r_{\text{last}} = 80$, (d) $k = 20$, $N = 42$, $f_l = 797/8192$, $f_h = 3918/8192$, $r_{\text{last}} = 130$, and (e) $k = 38$, $N = 51$, $f_l = 814/8192$, $f_h = 4046/8192$, $r_{\text{last}} = 0$.

prediction with $E_1 = 509$ has taken the third place while the best $E_1 = 408$ has been achieved by the Kalman smoother method. Further, the MSE for the first 80

prediction data,

$$E_2 = \frac{1}{80}\sum_{b=0}^{3}\sum_{t\in T_b^{\text{pred}}}(r(t)-\widehat{r}(t))^2 \tag{39}$$

has also been reported, and $E_2 = 418$ of ours has taken the ninth place while the best $E_2 = 222$ has been achieved by the ensemble model method.

4 Analysis of the results: advantages and disadvantages of the current method

We have analyzed our result submitted to the competition with the real data of missing data which have been provided after the competition.

4.1 Effectiveness of FFT bandpass filtering

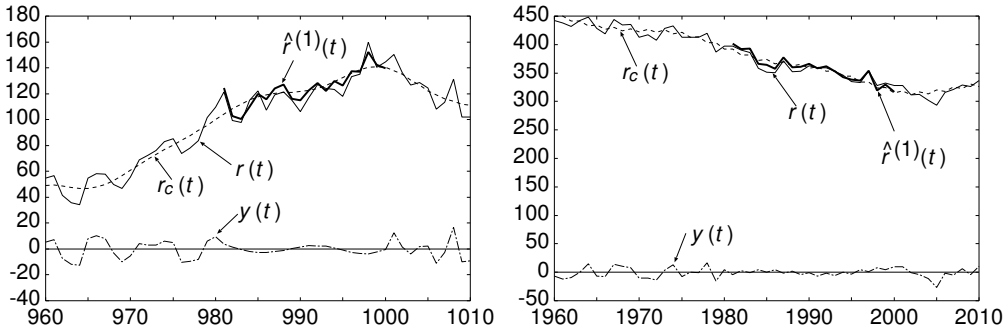
First, we have examined the effectiveness of the FFT bandpass filtering which neglects high frequency components just like the Kalman smoother method. We have found out that the smooth data $r_c(t)$ obtained via the FFT bandpass filters has achieved $E_1 = 440, 450, 454$ and $E_2 = 390, 378, 379$, respectively, for the mth ($m = 1, 2, 3$) iterations of the parameter optimization process as described in Section **2.4**. Among these results, the one at the first iteration seems to be good compared with the predictions submitted to the competition. Further, in order to clarify the ability of the FFT bandpass filtering, we have executed the following simple examination; we have optimized only the trial values $r_{\text{last}} = -120, -119, \cdots, 19, 20$, with constant $f_l = 410/8192$ and $f_h = 0.5$ which remove the frequencies higher than the one with the period $L_l = 1/f_l \simeq 20$ which seems reasonable because the time length of each missing block is also 20. Then, we have the optimum $r_{\text{last}} = -60$ which has achieved the minimum MSE (=128) of the smooth data $r_c(t)$ for all given test data, and we have $E_1 = 566$ and $E_2 = 410$ for the prediction data. Further, with the same procedure as above for $f_l = 205/8192 \simeq 1/40$, we have $r_{\text{last}} = 7$ which achieves the minimum MSE (=181) for all given test data, $E_1 = 430$ and $E_2 = 346$, where $E_1 = 430$ is better than $E_1 = 441$ of the recurrent neural network method which achieves the second least E_1 at the competition, and $E_2 = 314$ is better than $E_2 = 346$ of the Kalman smoother method. These results indicate that the FFT filtering works well, but it needs a method to decide which value of f_l is better for the CATS benchmark prediction, where note that E_1 and E_2 for prediction data do not decrease with the decrease of the MSE for all given data as shown above. Further, the prediction by the FFT filtering does not involve high frequency components, which we are not satisfied with.

4.2 Effectiveness of the Parameter Validation Method

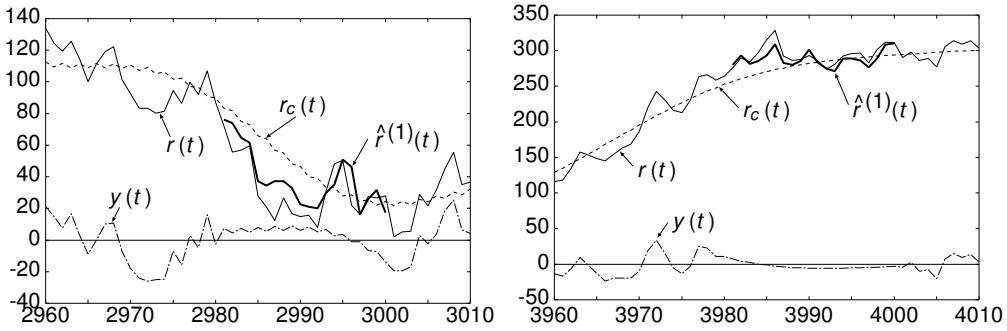
Next, we have calculated the MSEs of our prediction data obtained at the iterations of the parameter optimization process as follows; $E_1 = 466$, 509, 503 and $E_2 = 390$, 418, 414, respectively, for the mth ($m = 1, 2, 3$) iteration, which indicates that our submitted prediction, or the one obtained at the 2nd iteration, is not better than the prediction at the first iteration which is also worse than the smooth data obtained by the FFT filtering as shown above. This disorder basically caused by the parameter validation method in the optimization process, which is affected by the properties of the time series, the learning ability of the CAN2, and so on. To examine these factors much more, we have run an experiment, where we use the real data for parameter validation instead of using the performance index $PI_b^{(m)}$ in Eq.(36) and then we have a very good prediction as shown in Fig. 4 whose MSEs are $E_1 = 80$ and $E_2 = 73$. This result indicates that the CAN2 has a possibility to learn the underlying dynamics of the data to be predicted from the training data excluding the prediction data. However, we here have to take an account of the overfitting phenomenon, namely, the prediction data validated with the real data may fit the real data to be predicted very well but it may not fit other data so well. So we have compared the performance index $PI_b^{(m)}$ of our method and the MSE for the real data with respect to a number of trial parameter values, and found that the performance index as well as the MSE have a number of local minima which do not correspond each other exactly on a wide range of parameter values except some ranges, which is supposed to be the main reason that our predictions prepared for submitting the competition had involved the disorder. However, inversely, our validation process seems to have worked in some ranges of parameter values so that the result was not so bad. Especially, our method is capable of selecting the parameters for smoothing (f_l and f_h) and estimating the last value r_{last} of the time series, where the former contributes to obtaining good E_1 and E_2 and the latter works especially for a good E_1, which is one of the advantages of our method.

4.3 Different Parameter Values for Different Blocks

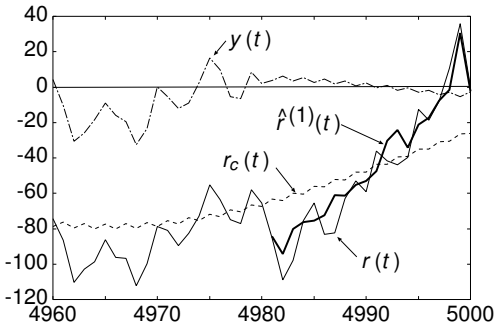
From Fig. 3 and Fig. 4, we can see that the parameter values optimized are different for different blocks. From the point of view of understanding the nature of the time series, this result might seem strange especially because the dimension of the input vector or the embedding dimension k is usually set constant even if the time series involves chaotic, time-varying, and other complicated properties. Actually, at a beginning stage of this research for the competition, we tried to identify a constant k for all blocks, but we could not obtain a good k which minimizes the performance index $PI_b^{(1)}$ for all blocks, and we gave up this strategy. Here, note that we had already given up the prediction of low frequency components because of insufficient number of given data as described above, which indicates an information loss and



(a) $\widehat{r}^{(1)}(t)$ for T_0^{pred} (b) $\widehat{r}^{(1)}(t)$ for T_1^{pred}



(c) $\widehat{r}^{(1)}(t)$ for T_2^{pred} (d) $\widehat{r}^{(1)}(t)$ for T_3^{pred}



(e) $\widehat{r}^{(1)}(t)$ for T_4^{pred}

Fig. 4. Prediction $\widehat{r}^{(1)}(t)$ optimized by the parameter validation using the real data. The parameter values are as follows; (a) $k = 9$, $N = 41$, $f_l = 732/8192$, $f_h = 4061/8192$, $r_{\text{last}} = -139$, (b) $k = 41$, $N = 47$, $f_l = 492/8192$, $f_h = 4059/8192$, $r_{\text{last}} = 122$, (c) $k = 7$, $N = 30$, $f_l = 232/8192$, $f_h = 4069/8192$, $r_{\text{last}} = -32$, (d) $k = 9$, $N = 43$, $f_l = 203/8192$, $f_h = 4049/8192$, $r_{\text{last}} = -182$, and (e) $k = 28$, $N = 40$, $f_l = 99/8192$, $f_h = 4036/8192$, $r_{\text{last}} = -29$.

may cause the difficulty of identifying a single k for all blocks. For example, when a signal is of such as a nonlinear system or a deterministic chaos, the identification of a single k for all blocks may be possible but difficult if we do not have sufficient number of data. Further, when a signal is of a time varying system, the embedding dimension as well as other parameter values may change from block to block. Moreover, since the given signal is said artificial, there are many other possibili-

ties. Thus, we took the strategy that we select the parameter values for each block independently. However, the amount of the change of parameter values, especially in Fig. 4, seem too big, which we have not analyzed enough so far.

4.4 Computational Cost

A disadvantage of our method was that we had to tune a number of parameter values combinatorially, which takes a lot of time for obtaining the final prediction. Another disadvantage is that our validation method had not been matured as described above for the CATS benchmark, which however may be true for all other methods and a key to solve the prediction problem lies on making a good validation method. In order to overcome these disadvantages, we have improved our method as shown in the next section.

5 Improved method for prediction

We have improved our method so that it performs better and it can be easily applied to the new data of the CATS benchmark (see [1]). We first modify the linear interpolation given in Eq.(31) as the following equation using the moving average of two data, or $\tilde{r}(t) = (r(t) + r(t+1))/2$ for $t = t_0 = 1000b + 979$ and $t = t_1 = 1000b + 1001$,

$$\widehat{r}^{(0)}(t) = \frac{\tilde{r}(t_1) - \tilde{r}(t_0)}{t_1 - t_0}(t - t_0 - 0.5) + \tilde{r}(t_0) \tag{40}$$

where, for the last block, we use trial values r_{last} for $\tilde{r}(5001) = r_{\text{last}}$.

As mentioned in the previous section, we have to overcome the overfitting problem, and one of the solutions, in general, is to reduce the freedom of parameter values. To have this done as well as for overcoming the time consuming parameter tuning and specifying the range of reasonable validation, we have obtained a reduced set of trial values for the CATS benchmark as follows: first we decide the step length $Q = 2$ for predicting the test (or validation) data $y(t)$ of t in $T_{bPQ}^{\text{test}} = \{1000b+P+j \mid j = 1, 2, \cdots, Q\}$ because long term prediction of the CATS benchmark is not so easy and we have fond that the values Q bigger than 3 are not so stable for validating trial parameter values although we have to predict up to 20 step ahead data. We use $P = 969$ and $M = 10$ for calculating the mean MSE $\left\langle MSE_{bPQ}^{\text{test}} \right\rangle_M$, where M is set as small as possible for a stable validation and P is set as near to the target prediction period $T_b^{\text{pred}} = \{1000b + j \mid j = 981, \cdots, 1000\}$ as possible because the CATS benchmark seems to have time dependency. Incidentally, we only use 400 training data $y(t)$ of t in $T_{bP}^{\text{train}} \triangleq \{1000b+P-j \mid j = 0, 1, 2, \cdots, 399\}$ for predicting $y(t)$

of t in T_{bPQ}^{test}. The trial last values r_{last} is important but the performance index is not so sensitive to the change of r_{last}, and we decided to use $r_{\text{last}} = r(4980) \pm \Delta_r$ for $\Delta_r = 0, \pm 30$, or $r_{\text{last}} = -35, -65, -95$ since $r(4980) \simeq -65$. Further, we use trial cutoff frequencies $f_l = 205/8192$ and $405/8192$ which correspond to the periods $L_l = 1/f_l \simeq 40$ and 20, respectively, and $f_h = 4058/8192$ corresponding to the period $L_h \simeq 2$. And we use the trial embedding dimensions $k = 7, 8, 9$ and the number of cells $N = 9$, where they affect the stability of the CAN2 and we set them small relatively to the best ones found so far (see Fig. 4) because the smaller ones achieve stable performance although they provide conservative performance. With these parameter values we have obtain the following E_1 and E_2 via the parameter optimization iterations described in Section **2.4**; $E_1 = 275$, 262, 251, 268, and $E_2 = 239$, 230, 212, 233, respectively, for the mth ($m = 1, 2, 3, 4$) iteration. These are all competitive to the predictions submitted to the competition, and the best $E_1 = 251$ and $E_2 = 212$ at the third iteration are better than all of the submitted predictions.

6 Conclusions

We have presented an approach to the CATS benchmark prediction, which uses Fourier bandpass filters for separating low and high frequency components of the time series and the CAN2 for learning and predicting high frequency components. For the CATS benchmark prediction, smoothing and estimating the last value of the time series are very important, and one of the advantages of our method is that the parameter values for smoothing and estimating the last value are obtained automatically by means of the optimization process of our method.

Finally, we would like to note that our works on the CAN2 are partially supported by the Grant-in-Aid for Scientific Research (B) 16300070 of the Japanese Ministry of Education, Science, Sports and Culture.

References

[1] A. Lendasse, E. Oja, O. Simula, M. Verleysen, "Time Series Prediction Competition: The CATS Benchmark," *Proc. of IJCNN 2004, International Joint Conference on Neural Networks, Budapest (Hungary)*, vol. II, pp.1615–1620, 25–29 July 2004.

[2] S.Kurogi, T.Ueno and M.Sawa "Batch Learning Competitive Associative Net and Its Application to Time Series Prediction," *Proc. of IJCNN 2004, International Joint Conference on Neural Networks, Budapest (Hungary)*, CD-ROM, 25–29 July 2004.

[3] T.Kohonen, "Associative Memory," *Springer Verlag*, 1977.

[4] D.E.Rumelhart and D.Zipser, "A feature discovery by competitive learning," *ed. D.E.Rumelhart, J.L.McClelland and the PDP Research Group, Parallel Distributed Processing, The MIT Press, Cambridge*, vol.1, pp.151–193, 1986.

[5] S.Kurogi, M.Tou and S.Terada, "Rainfall estimation using competitive associative net," *Proc. of 2001 IEICE General Conference (in Japanese)*, vol.SD-1, pp.260–261, 2001.

[6] S. Kurogi, "Asymptotic optimality of competitive associative nets for their learning in function approximation," *Proc. of the 9th International Conference on Neural Information Processing*, vol. 1, pp. 507–511, 2002.

[7] S. Kurogi, H. Sakamoto, H.Nobutomo, Y.Fuchikawa, T.Nishida, M.Mimata, K.Itoh, "Asymptotic minimization of the approximation error of competitive associative nets and its application to temperature control of RCA cleaning solutions," *Proc. of the 9th International Conference on Neural Information Processing*, vol. 4, pp. 1900–1904, 2002.

[8] S. Kurogi, "Asymptotic optimality of competitive associative nets and its application to incremental learning of nonlinear functions," *Trans. of IEICE D-II (in Japanese)*, vol. J86-D-II no. 2, pp. 184–194, 2003.

[9] S. Kurogi, T. Ueno and M. Sawa, "A batch learning method for competitive associative net and its application to function approximation," Proc. of SCI2004, no.V, pp. 24–28, 2004.

[10] A.Gersho, "Asymptotically optimal block quantization," IEEE Trans. Information Theory, vol.IT-25, no.4, pp.373–380, 1979.

[11] C.Chinrungrueng and C.H.Séquin, "Optimal adaptive k-means algorithm with dynamic adjustment of learning rate," IEEE Trans. Neural Networks, vol.6, no.1, pp.157–169, 1995.

[12] T. Nishida and S. Kurogi, "Adaptive vector quantization using re-initialization method," *Trans.of IEICE D-II (in Japanese)*, vol.J84-D-II, no.7, pp.1503–1511, 2001.

[13] S.Kurogi and T.Nishida, "Adaptive and predictive control using competitive associative net for learning and switching of multiple models," *Trans. of SICE (in Japanese)*, vol.37, no.3, pp.203–212, 2001.