Forecasting Using First-Order Difference of Time Series and Bagging of Competitive Associative Nets

Shuichi Kurogi, Ryohei Koyama, Shinya Tanaka, and Toshihisa Sanuki

***Abstract*— This article describes our method used for the 2007 Forecasting Competition for Neural Networks and Computational Intelligence. We have employed the first-order difference of time series for dealing with the seasonality of the monthly data. Since the differencing removes the trend of time series, we have developed a method to estimate the trend. Moreover, we have used the bagging of competitive associative net called CAN2 as a learning predictor, where the CAN2 is for learning an efficient piecewise linear approximation of a nonlinear function, and the bagging for reducing the variance of the prediction.**

I. Introduction

This article describes the method which we have used for the 2007 Forecasting Competition for Neural Networks and Computational Intelligence. At the competition, the competitors should forecast 11 or 111 time series as accurately as possible by means of using their methods. One of the difficulties of this problem is that the 11 and 111 time series are monthly data which involve various properties of time series, such that some are stationary but some have positive trend, not all but some involve seasonality with the period 12, some seem to have outliers, and so on. Thus, we would like to employ or develop a general method which can overcome the above difficulties. One of the techniques which we have employed is to use the first-order difference of time series for dealing with the seasonality. Here, since the differencing removes the trend of the time series, we have also developed a method to estimate the trend of time series.

On the other hand, to cope with the competition problem, our first decision was to use our competitive associative net called CAN2. Here, the CAN2 has been introduced for utilizing competitive and associative schemes [1], [2] and learning an efficient piecewise linear approximation of a nonlinear function. This approach has been shown effective in several areas such as function approximation, control, rainfall estimation and time series prediction [3]-[8]. Here, note that the differences of the CAN2 to other similar methods are as follows. The method of local linear models [9] uses linear models obtained from K-nearest neighbors of input vectors while the CAN2 utilizes linear models (associative memories) optimized by the learning method involving competitive and associative schemes. The CAN2 may be viewed as a mixture-of-experts model that utilizes linear models as experts and competitive scheme as gating. Although the MARS (multivariate adaptive regression splines) model [10] as a mixture-of-experts model executes continuous piecewise linear approximation, the CAN2 executes discontinuous one intending for optimizing each linear model in the corresponding Voronoi region. Further, we employ the bagging scheme [11] for reducing the variance of the prediction by the CAN2.

In the following sections, we describe our method in detail, some experimental results, and then the conclusion.

Shuichi Kurogi, RyoheiKoyama, Shinya Tanaka, and Toshihisa Sanuki are with the Department of Control Engineering, Kyushu Institute of Technology, Tobata, Kitakyushu 804-8550, Japan (phone: +81-93-884-3188; fax: +81-93-861-1159; email: kuro@cntl.kyutech.ac.jp).

II. Method for the Forecasting Competition

Here, we show our method employed for the forecasting competition after formalizing the competition problem.

A. The Problem to be Solved

At the competition, a set of real valued monthly time series $y_i(t)$ $(\in \mathbb{R})$ is provided, where $t \in T_i^{given} \triangleq I[s_i^{given}, t_i^{given}]$ denotes a point in time or a month, and i indicates the index of a time series in the index set $I^A \triangleq I[1, 111]$ of the dataset A (or the complete dataset) or $I^B \triangleq I[101, 111]$ of the dataset B (or the reduced dataset). Here, $I[s, t] \triangleq \{s, s+1, s+2, \cdots, t\}$ indicates a set of integers from s to t. For each time series $y_i(t)$, a competitor should predict (forecast) $y_i(t)$ for the fixed time horizon of $h = 18$ months, or for $t \in T_i^{forecast} \triangleq I[t_i^{given}+1, t_i^{given}+h]$. The forecasts $\hat{y}_i(t)$ for $t \in I[t_i+1, t_i+h]$ are evaluated based on the SMAPE (Symmetric Mean Absolute Percent Error) given by

$$SMAPE = \frac{1}{h} \sum_{t=t_i+1}^{t_i+h} \frac{|\hat{y}_i(t) - y_i(t)|}{(\hat{y}_i(t) + y_i(t))/2} \times 100. \quad (1)$$

B. Forecasting Using Original Time Series

Here, we show the forecasting using the original time series on which the forecasting using the first-order difference shown below is based. Firstly, we suppose that the time series $y_i(t)$ satisfies

$$y_i(t) = f_i(\boldsymbol{x}_i(t)) + \epsilon_i(t), \quad (2)$$

where $\epsilon_i(t)$ represents noise, and $f_i(\boldsymbol{x}_i(t))$ is a nonlinear function of a vector $\boldsymbol{x}_i(t) \triangleq (x_{i1}(t), x_{i2}(t), \cdots, x_{ik_i}(t))^T$. Here, the jth element $x_{ij}(t)$ of $\boldsymbol{x}_i(t)$ is a data point of a given time series with a delay, $y_l(t-\tau) = y_{l_{ij}}(t-\tau_{ij})$, whose index l_{ij} and delay τ_{ij} are selected from all l and τ

so that $y_{l_{ij}}(t-\tau_{ij})$ has the jth largest correlation:

$$R_i(l,\tau) = \frac{\sum_t \left(y_i(t)-\overline{y_i(t)}\right)\left(y_l(t-\tau)-\overline{y_l(t-\tau)}\right)}{\sqrt{\sum_t \left(y_i(t)-\overline{y_i(t)}\right)^2}\sqrt{\sum_t \left(y_l(t-\tau)-\overline{y_l(t-\tau)}\right)^2}}, \tag{3}$$

where $\overline{y_i(t)}$ and $\overline{y_l(t-\tau)}$ indicate the mean of $y_i(t)$ and $y_l(t-\tau)$, respectively. Here, the range of t for the mean is set so that both $y_i(t)$ and $y_l(t-\tau)$ can have provided values, and we set $R_i(l,\tau)=0$ if the number of $y_i(t)$ and $y_l(t-\tau)$ for the mean is little than $h=18$. In order to forecast unknown values of the time series, we use the bagging of the CAN2 (see below for details) as a predictor. Further, for estimating the performance of the predictor and tune the parameter values of the predictor, we run validation tests which use $y_i(t)$ $(t \in T_i^{train} = I[s_i^{given}, t_i])$ for training the predictor, and $y_i(t)$ $(t \in T_i^{pred} = I[t_i+1, t_i+h])$ for validating the prediction, where t_i is a point in time which satisfies $t_i \le t_i^{given} - h$ for validation. Of course, we can make the final forecast by using $t_i = t_i^{given}$. The predictor, at first, learns $y_i(t)$ $(t \in T_i^{train})$ to make the multi-step prediction given by

$$\widehat{y}_i(t) = f_i(\widehat{\boldsymbol{x}}_i(t), \theta_i), \tag{4}$$

where θ_i represents the parameter values of the predictor, and the element of the vector $\widehat{\boldsymbol{x}}_i(t) \triangleq (\widehat{x}_{i1}(t), \widehat{x}_{i2}(t), \cdots, \widehat{x}_{ik_i}(t))^T$ is given by

$$\widehat{x}_{ij}(t) \triangleq \begin{cases} y_{l_{ij}}(t-\tau_{ij}) & \text{for } t-\tau_{ij} \in T_{l_{ij}}^{train}, \\ \widehat{y}_{l_{ij}}(t-\tau_{ij}) & \text{for } t-\tau_{ij} \in T_{l_{ij}}^{pred}. \end{cases} \tag{5}$$

where the elements $\widehat{x}_{ij}(t)$ are determined successively for $t = t_i+1, t_i+2, \cdots$. Thus, the learning, forecasting, and validation can be done by the above procedure.

C. Forecasting Using First-Order Difference

The forecasting using the first-order difference is described as follows (see Section II-E.1 for detailed reasons of this method). First, we make the first-order difference $\Delta y_i(t) = y_i(t) - y_i(t-1)$, and use a predictor to learn $\Delta y_i(t)$ for $t \in T^{train} \backslash \{s_i^{given}\} = I[s_i^{given}+1, \cdots, t_i]$, and get the multi-step prediction $\widehat{\Delta y}_i(t) = f_i^d(\widehat{\Delta \boldsymbol{x}}_i(t), \theta_i^d)$ for $t \in T_i^{pred} = I[t_i+1, t_i+h]$, where the elements of the vector $\widehat{\Delta \boldsymbol{x}}_i(t) \triangleq (\widehat{\Delta x}_{i1}(t), \widehat{\Delta x}_{i2}(t), \cdots, \widehat{\Delta x}_{ik_i^d}(t))^T$ are generated by means of replacing $\widehat{x}_{ij}(t)$ and $y_{l_{ij}}(t-\tau_{ij})$ in Eq.(5) by $\widehat{\Delta x}_{ij}(t)$ and $\Delta y_{l_{ij}}(t-\tau_{ij})$, respectively. The dimension k_i^d is of the vector $\widehat{\Delta \boldsymbol{x}}_i(t)$, and $f_i^d(\widehat{\Delta \boldsymbol{x}}_i(t), \theta_i^d)$ is a nonlinear function of $\widehat{\Delta \boldsymbol{x}}_i(t)$ achieved by the predictor with the parameter values represented by θ_i^d. Then, the prediction of $y_i(t)$ for $t \in T_i^{pred}$ is constructed by

$$\widehat{y}_i^d(t)|_{t_i} \triangleq y_i(t_i) + \sum_{j=t_i+1}^{t} \widehat{\Delta y_i}(j)|_{t_i}. \tag{6}$$

Here, $\widehat{\Delta y_i}(j)|_{t_i}$ is the prediction $\widehat{\Delta y_i}(j)$ obtained by the predictor that has learned $\Delta y_i(t)$ for $t \in I[s_i^{given}+1, t_i]$, and we sometimes neglect "$|_{t_i}$" for simplicity.

As shown in Section II-E.1, the trend of the prediction $\widehat{y}_i^d(t)|_{t_i}$ is not so reliable, so we first employ an averaging method. Namely, we use

$$\widehat{y}_i^{d(a)}(t) \triangleq \frac{1}{a}\sum_{l=0}^{a-1}\sum_{j=t_i+1}^{t} \widehat{y}_i^d(t)|_{t_i-l}, \tag{7}$$

where $a(\ge 1)$ is the number of averaging. Further, we employ a method to modify the trend of the prediction as follows: first, we make a mth order polynomial approximation $\tilde{y}_i^{(m)}(t) = \sum_{j=0}^m a_j t^j$ of $y_i(t)$ for $t \in T^{given}$ and a trial value of $y_T = y(t_i+h)$ for $m = 0, 1, 2, \cdots$, where $\tilde{y}_i^{(m)}(t)$ represents the mth order polynomial trend over $T_i^{given} \cup \{t_i+h\}$. Next, we make a first-order approximation $\tilde{y}_i^{(m,1)}(t) = b_0 + b_1 t$ of $\tilde{y}_i^{(m)}(t)$ for $t \in I[t_i-h+1, t_i]$, where $\tilde{y}_i^{(m,1)}(t)$ represents the first-order short-term trend of $y_i(t)$. We can select the best y_T to minimize the loss function $L(\tilde{y}_i^{(m,1)}, y_i, t_i-h+1, t_i)$ by means of a line search of y_T for each $m = 0, 1, 2, \cdots$. Further, we can determine m from a view of $\tilde{y}_i^{(m)}(t)$ and $y_i(t)$ (see the experimental result shown below). With the selected $y_T = y_T^*$ and $m = m^*$, we obtain the first-order short-term trend $\tilde{y}_i^{(m,1)}(t) = b_0^* + b_1^* t$ for the forecast period $T_i^{forecast}$. On the other hand, from the prediction $\widehat{y}_i^d(t)$, we can obtain the first-order trend (approximation) as $\hat{y}_i^{(d,1)}(t) = c_0 + c_1 t$. Then, we can modify the trend of the prediction $\widehat{y}_i^d(t)$ as

$$\widehat{y}_i^{d(a,m)}(t) = \widehat{y}_i^{d(a)}(t) + (b_0^* - c_0) + (b_1^* - c_1)t. \tag{8}$$

D. CAN2 and the Bagging

1) Assumptions on the given dataset: Let $D^n \triangleq \{(\boldsymbol{x}_i, y_i) | i \in I^n\}$ be a given training dataset, where $I^n \triangleq \{1, 2, \cdots, n\}$ denotes the index set of the dataset, and $\boldsymbol{x}_i \triangleq (x_{i1}, x_{i2}, \cdots, x_{ik})^T$ and y_i denote an input vector and the target scalar value, respectively. Note that $\boldsymbol{x}_i$ and y_i, respectively, correspond to $\boldsymbol{x}_i(t)$ and $y_i(t)$, or $\Delta \boldsymbol{x}_i(t)$ and $\Delta y_i(t)$, introduced in the previous section. Here, we suppose the relationship given by

$$y_i \triangleq r_i + \epsilon_i = f(\boldsymbol{x}_i) + \epsilon_i, \tag{9}$$

where $r_i \triangleq f(\boldsymbol{x}_i)$ is a nonlinear function of $\boldsymbol{x}_i$, and ϵ_i represents zero-mean noise with the variance σ_i^2.

2) CAN2: A CAN2 has N units (see Fig. 1). The jth unit has a weight vector $\boldsymbol{w}_j \triangleq (w_{j1}, \cdots, w_{jk})^T \in \mathbb{R}^{k\times 1}$ and an associative matrix (or a row vector) $\boldsymbol{M}_j \triangleq (M_{j0}, M_{j1}, \cdots, M_{jk}) \in \mathbb{R}^{1\times(k+1)}$ for $j \in I^N \triangleq \{1, 2, \cdots, N\}$. The CAN2 approximates the above function $f(\boldsymbol{x}_i)$ by

$$\widehat{y}_i \triangleq \widehat{f}(\boldsymbol{x}_i) \triangleq \widetilde{y}_{c(i)} \triangleq \boldsymbol{M}_{c(i)} \widetilde{\boldsymbol{x}}_i, \tag{10}$$

where $\widetilde{\boldsymbol{x}}_i \triangleq (1, \boldsymbol{x}_i^T)^T \in \mathbb{R}^{(k+1)\times 1}$ denotes the (extended) input vector to the CAN2, and $\widetilde{y}_{c(i)} = \boldsymbol{M}_{c(i)} \widetilde{\boldsymbol{x}}$ is the output value of the $c(i)$th unit of the CAN2. The index $c(i)$ indicates

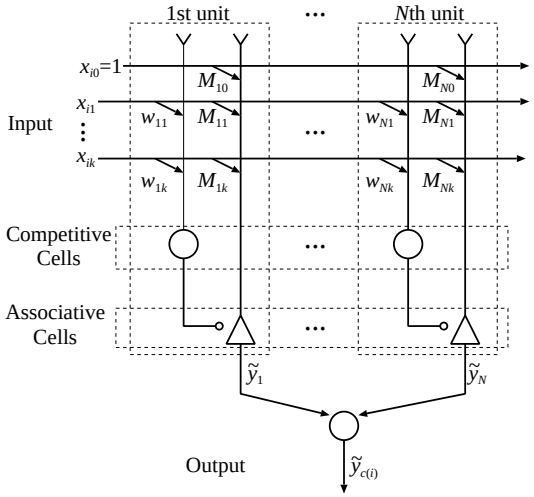

Fig. 1. Schematic diagram of the CAN2

the unit who has the weight vector $\boldsymbol{w}_{c(i)}$ closest to the input vector $\boldsymbol{x}_i$, or

$$c(i) \triangleq \underset{j \in I^N}{\text{argmin}} \|\boldsymbol{x}_i - \boldsymbol{w}_j\|. \tag{11}$$

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

$$V_j \triangleq \{\boldsymbol{x} \mid j = \underset{i \in I^N}{\text{argmin}} \|\boldsymbol{x} - \boldsymbol{w}_i\|\}, \tag{12}$$

for $j \in I^N$, and performs piecewise linear approximation of the function $f(\boldsymbol{x})$.

Note that we have developed an efficient batch learning method (see [6] for details), and we use it in the present competition. The method consists of iterations of (1) competitive learning based on a gradient method, (2) associative learning employing recursive least squares, and (3) reinitialization of units based on an "asymptotic optimality" criterion (see [4]) for overcoming local minima problems of the gradient method. We have used the same parameter values as for the function approximation problems shown in [6], except the number of units involved in the CAN2 which is tuned so that the prediction achieves smaller SMAPE for the validation periods (see Section II-B and Section III).

3) Bagging: Let $D_j^{\alpha n*}$ be the jth bootstrap sample set (multiset, or bag) involving αn elements, where the elements in $D_j^{\alpha n*}$ are resampled randomly with replacement from the given training dataset D^n, and $\alpha > 0$. Here, we would like to mention that an element in D^n is not in $D_j^{\alpha n*}$ with the probability $(1 - 1/n)^{\alpha n}$ which approximately is $\exp(-\alpha)$ when n is large. Thus, the number of "individual" or different elements in $D_j^{\alpha n*}$ approximately is $n_{\text{eff}}(\alpha) \triangleq n(1 - \exp(-\alpha))$. For example, $n_{\text{eff}}(1) \simeq 0.632n$ which is used in the conventional bagging methods [11], [12], and $n_{\text{eff}}(0.7) \simeq 0.503n$, which we have employed in the present method because of its empirical good performance in several prediction problems (see e.g. [8]).

The bagging (bootstrap aggregation) for estimating the target value $r_i = f(\boldsymbol{x}_i)$ is done by the mean given by

$$\hat{y}_i^{b^*} \triangleq \frac{1}{b} \sum_{j \in I^b} \hat{y}_i^j, \tag{13}$$

where $\hat{y}_i^j \triangleq \hat{y}^j(\boldsymbol{x}_i)$ denotes the prediction by the jth predictor (CAN2) which has learned $D_j^{\alpha n*}$.

E. Analysis of the Method

Here, we show some analysis of the present method for examining how the method works.

1) Forecasting using the first-order difference: An advantage of using the first-order difference is supposed to be based on that the range of the input vector $\Delta\boldsymbol{x}_i(t)$ is smaller than that of the original input vector $\boldsymbol{x}_i(t)$. Thus, even if there are few training data similar to the data to be predicted in the original input space, much more training data are available via the first-order difference. For example, suppose a time series $y_i(t) = y_1(t)+y_2(t)$ consisting of $y_1(t) = y_1(t-1)+1$ and $y_2(t) = y_2(t-1) - y_2(t-2) + 1$ with $y_1(1) = y_2(1) = y_2(2) = 1$. Then, $y_1(t) = t = 1, 2, \cdots$ is an increasing series, and $y_2(t) = y_2(t+6) = 1, 1, 0, -1, -1, 0, \cdots$ for $t = 1, 2, 3, 4, 5, 6, \cdots$ is a periodic series with the period 6. Thus, $y_i(t)$ is an increasing series with fluctuation. Then, the function $y_i(t) = f_i(\boldsymbol{x}_i(t))$ of the embedding vector $\boldsymbol{x}_i(t) = (y_i(t-1), y(t-2), \cdots, y_i(t-k_i))^T$ is hard to be learned because $\boldsymbol{x}_i(t)$ is different for every $t = 1, 2, \cdots$, where the dimension k_i is assumed to be sufficiently large for $f_i(\boldsymbol{x}_i(t))$ to be a function (e.g. $k_i \geq 6$). However, $\Delta y_i(t) = 1 - y_2(t-2)$ is a periodic series with the period 6, and then the function $\Delta y_i(t) = f_i^d(\Delta\boldsymbol{x}_i(t))$ of $\Delta\boldsymbol{x}_i(t) = (\Delta y_i(t-1), \Delta y_i(t-2), \cdots, \Delta y_i(t-k_i^d))$ is easy to be learned because there are only 6 different patterns of input-output pairs $(\Delta\boldsymbol{x}_i(t), y(t))$ to be learned and they appear many times in a training dataset with a sufficiently large number of members, where $k_i^d \geq 2$ is necessary for $f_i^d(\Delta\boldsymbol{x}_i(t))$ to be a function.

A disadvantage of using the first-order difference is that the trend of the prediction is unstable and unreliable, which we can see from the following example: suppose that a time series $y_i(t) = y_i(t-1) + a + \epsilon_t$ for $t = 1, 2, \cdots$ is given, where $y_i(0) = 0$, $a \neq 0$ and ϵ_t represents a noise. Then, the time series without noise is represented by $r_i(t) = at$, and the first-order difference without noise is written by $\Delta y_i(t) = (1-b)a + b\Delta y_i(t-1)$ for a certain b $(0 \leq b \leq 1)$. If a predictor learns to predict the ideal first-order difference without noise $\widehat{\Delta y}_i(t) = (1-b)a + b\Delta y_i(t-1)$, the prediction of $\Delta y_i(t)$ for $t = t_i + 1, t_i + 2, \cdots$ is given by $\widehat{\Delta y}_i(t)|_{t_i} = a + (\epsilon_{t_i} - \epsilon_{t_i-1})b^{t-t_i}$. Thus, the reconstructed prediction is given by $\widehat{y}_i^d(t)|_{t_i} = at + \epsilon_{t_i} + (\epsilon_{t_i} - \epsilon_{t_i-1}) \sum_{j=1}^{t-t_i} b^j$. So, the absolute value of the prediction error, $\left|\widehat{y}^d(t)|_{t_i} - r(t)\right| = |\epsilon_{t_i} + (\epsilon_{t_i} - \epsilon_{t_i-1}) \sum_{j=1}^{t-t_i} b^j|$ increases with the increase of time t for $\epsilon_{t_i} \neq \epsilon_{t_i-1}$ and $b \neq 0$, and the sign of the error depends on the noise ϵ_{t_i} and ϵ_{t_i-1}. Thus, in order to overcome this instability of the trend, we have developed the method described in Section II-C.

2) Coefficients of the bagging: Instead of the bagging prediction given by Eq.(13), a more general aggregation of predictions is given by

$$\tilde{y}_i^{b^*} \triangleq \sum_{j \in I^b} b_j \hat{y}_i^j, \tag{14}$$

where we suppose that $b_j \geq 0$ and $\sum_{j \in I^b} b_j = 1$. Since the prediction $\hat{y}_i^j$ involves the variation caused by the noise ϵ_i in the training data and the variation of the bootstrap resampling dataset $D_j^{\alpha n*}$ for $j \in I^b$, the mean μ_i, the variation δ_i^j and the variance ρ_i^2 of the prediction $\hat{y}_i^j = \mu_i + \delta_i^j$ are given by $\mu_i \triangleq E_{(\epsilon_i, D_j^{\alpha n*})}\left(\hat{y}_i^j\right)$, $\delta_i^j \triangleq \hat{y}_i^j - \mu_i$, $\rho_i^2 \triangleq E_{(\epsilon_i, D_j^{\alpha n*})}\left((\delta_i^j)^2\right)$, where $E_{(\epsilon_i, D_j^{\alpha n*})}(\cdot)$ is the mean with respect to the variation of ϵ_i and the variation of $D_j^{\alpha n*}$. Since the predictor learns $y_i = r_i + \epsilon_i$, the variation δ_i^j of the prediction is supposed to have a positive correlation with ϵ_i, or

$$E_{(\epsilon_i, D_j^{\alpha n*})}\left(\delta_i^j \epsilon_i\right) > 0. \tag{15}$$

Then, the expectation of the squared prediction error $(\tilde{e}_i^{b^*})^2 = (\tilde{y}_i^{b^*} - y_i)^2$ is given by

$$\begin{aligned} &E_{(\epsilon_i, D_j^{\alpha n*})}\left((\tilde{e}_i^{b^*})^2\right) \\ &= \sum_{l \in I^b} b_l^2 \left[(\mu_i - r_i)^2 + E_{(\epsilon_i, D_j^{\alpha n*})}\left((\delta_i^j - \epsilon_i)^2\right)\right], \\ &\geq \frac{1}{b}\left[(\mu_i - r_i)^2 + E_{(\epsilon_i, D_j^{\alpha n*})}\left((\delta_i^j - \epsilon_i)^2\right)\right], \end{aligned} \tag{16}$$

where the inequality is derived by the arithmetic-geometric mean inequality, and the equality holds when b_j is constant. Thus, the mean of the square error of the aggregation takes the minimum when $b_j = 1/b$ ($j \in I^b$). Therefore, the bagging prediction given by Eq.(13) is supposed to be the most effective predictions among the aggregation given by Eq.(14).

3) Bias and variance decomposition of prediction error: The generalization error or the prediction error for the population data is given by $L^{gen} \triangleq \sum_{i \in I^{pop}} \left(e_i^{b^*}\right)^2$, where I^{pop} indicates the index set of the population, $e_i^{b^*} = \hat{y}_i^{b^*} - y_i$ indicates the prediction error. Let us suppose $\hat{y}_i^{b^*} = \mu_i^{b^*} + \delta_i^{b^*}$, where $\mu_i^{b^*} \triangleq E_{(\epsilon_i, D_j^{\alpha n*})}\left(\hat{y}_i^{b^*}\right) (= \mu_i)$ is the prediction mean, $\delta_i^{b^*} = (1/b)\sum_{j \in I^{b^*}} \delta_i^j$ the variation from the mean. Then, the expectation of L^{gen} is given by

$$E_{(\epsilon_i, D_j^{\alpha n*})}(L^{gen}) = \sum_{i \in I^{pop}} \left((\beta_i^{b^*})^2 + \frac{\rho_i^2}{b} + \sigma_i^2\right), \tag{17}$$

where $\beta_i^{b^*} \triangleq E_{(\epsilon_i, D_j^{\alpha n*})}\left(\hat{y}_i^{b^*} - y_i\right) = \mu_i^{b^*} - r_i$ denotes the bias term, and ρ_i^2/b represents the variance term. Thus, the variance term of the bagging, ρ_i^2/b, can be reduced by the increase of b. Here, we would like to note that in order to reduce the bias term $(\beta_i^{b^*})^2$, we have developed a bagboosting method [13] for the CAN2, but we had few improvements. We think it is because the amount of the bias is not so large. Since we had suffered from a huge computational cost with little improvement of the performance, we abandoned the use of the bagboosting method.

In addition, the above analysis is to see that the variance of the prediction is reduced by the bagging. Here, the reduction of the variance is important for selecting optimal parameter values, such as the input elements $x_{ij} = y_{l_{ij}}(t - \tau_{ij})$ (see Section II-B), and the number N of the units in a CAN2 (see Section II-D.2). Namely, we will select such parameter values so that they can achieve smallest SMAPE for a couple of validation periods, and the SMAPE chages largely from period to period if the variance is big.

III. Experimental Results

Here, we show some experimental results for the reduced dataset consisting of 11 time series. We have run a validation test for $y_{101}(t)$ or the 101th time series. The predictions using the first-order difference for a validation period $T_{101}^{pred} = I[157, 174]$ are shown in Fig. 2, where the initial time $t = 1$ represents January 1978, which is the earliest time of all provided time series. From (a), we can see that both the predictions using the first-order difference, $\widehat{\Delta y}_{101}(t)|_{156}$ and $\widehat{\Delta y}_{101}(t)|_{155}$, predict the first-order difference $\Delta y_{101}(t)$ of the given time series $y_{101}(t)$ very well. However, from (b), we can see that the directly reconstructed time series $\widehat{y}_{101}^d(t)|_{156}$ and $\widehat{y}_{101}^d(t)|_{155}$ are not so good. Namely, we can see that the error of them is bigger than that of $\widehat{\Delta y}_{101}(t)|_{156}$ and $\widehat{\Delta y}_{101}(t)|_{155}$. However, from Fig. 2(c), we can see that the modified predictions $\widehat{y}_{101}^{d(2,m)}(t)$ for $m = 0, 1, 2$ seem to have achieved a good performance. Here, as shown in Fig. 2(d), (e) and (f), we have used the mth order polynomial approximation $\tilde{y}_{101}^{(m)}(t)$ of the original data and a expected final value $y_T = y_{101}(192)$ (which is tuned by a line search to be $y_T = y_T^*$; see Section II-C) for obtaining $\widehat{y}_{101}^{d(2,m)}(t)$. As shown at the comment in Fig. 2(c), the SMAPE as a performance index of $\widehat{y}_{101}^{d(2,m)}(t)$ is smallest for $m = 1$ in this prediction period $I[157, 174]$. However, we can see that every approximation seems reasonable from Fig. 2(d), (e) and (f). Therefore, as one of the solutions, we decide to use $m = 0$ which has the medium trend. Here, we would like to note that the tuned final value $y_T^* = y_{101}(192)$ determine the trend of the forecasting period $I[175, 192]$ for the submission. Actually $y_T^* = 5116, 5192, 4996$ for $m = 0, 1, 2$ in Fig. 2(d), (e), (f), respectively, and y_T^* for $m = 0$ takes the medium value. Further, the tuned final value y_T^* for $m = 3, 4, 5$ are $4801, 5349, 6402$, respectively, which also show that $m = 0$ is reasonable. However, a larger m bigger than 2 often provided unreasonable trend. Thus, we usually compare the predictions only for $m = 0, 1, 2$.

The predictions for the forecasting period $T_{101}^{forecast} = I[175, 192]$ for the submission are shown in Fig. 3. Here, the target values $y_{101}(t)$ are unknown for $t \in T_{101}^{forecast}$. Although both $\widehat{y}_{101}^d(t)|_{174}$ and $\widehat{y}_{101}^d(t)|_{173}$ increase to above 5500 at $t = 192$, we have examined that $\widehat{y}_{101}^d(t)|_{172}$ decreases to below 5000 as the increase of time. Thus, we think that the modified predictions $\widehat{y}_{101}^{d(2,m)}(t)$ for $m = 0, 1, 2$ are reasonable. We can also confirm that $\widehat{y}_{101}^{d(2,m)}(t)$ for $m = 0$ achieves the medium trend. Further, the difference of $\widehat{y}_{101}^{d(2,m)}(t)$ for $m = 0, 1, 2$ is not so large, so the selection

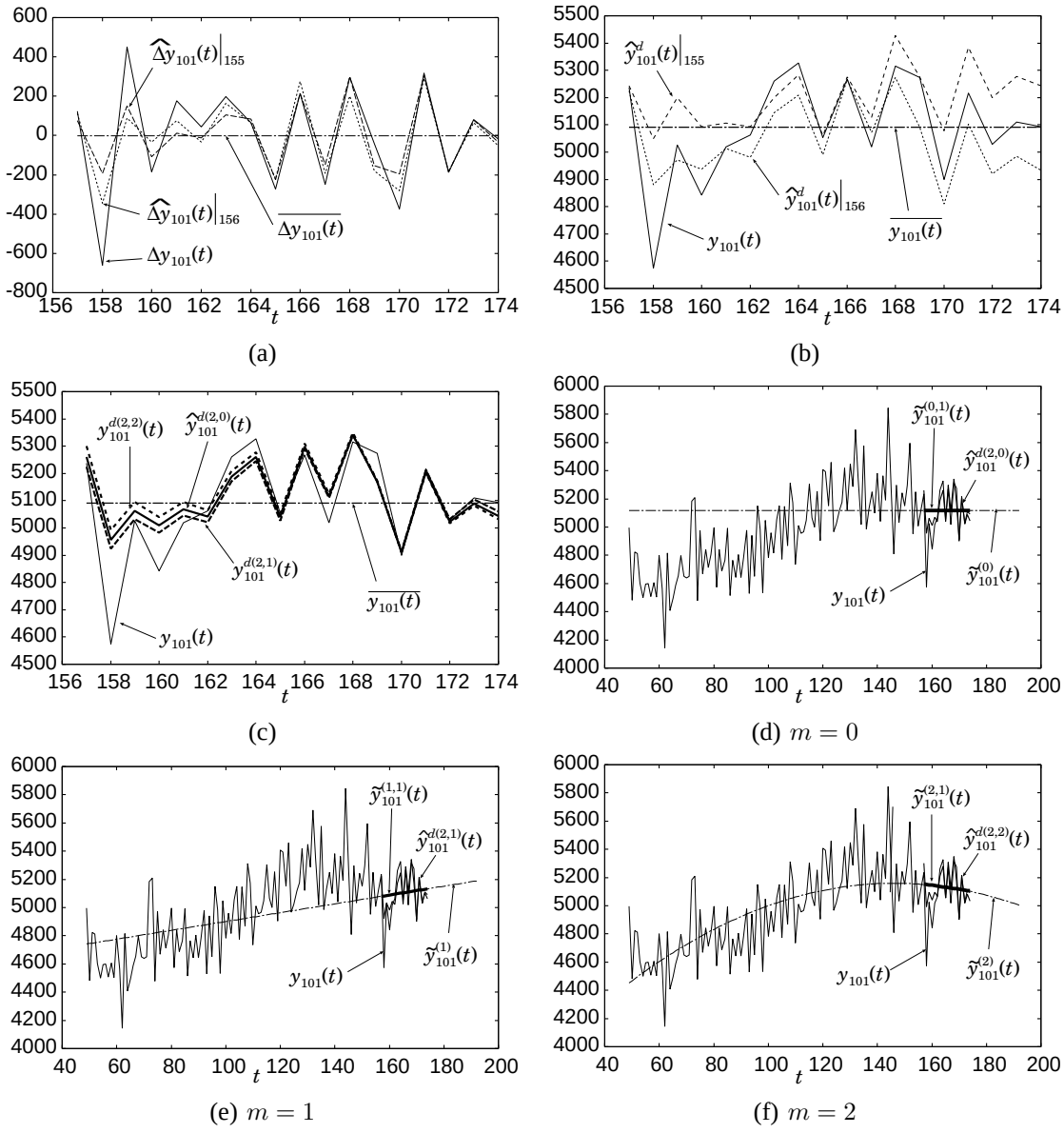

Fig. 2. Example of predictions using the first-order difference for a validation period. (a) Predictions $\widehat{\Delta y}_{101}(t)|_{156}$ and $\widehat{\Delta y}_{101}(t)|_{155}$ of the first-order difference $\Delta y_{101}(t)$. (b) Reconstructed predictions $\hat{y}^d_{101}|_{156}$ and $\hat{y}^d_{101}|_{155}$. (c) The modified predictions $\hat{y}^{d(2,m)}_{101}(t)$ for $m = 0, 1, 2$, where the SMAPE is 0.463, 0.462 and 0.493, respectively. (d),(e) and (f) show the mth order polynomial approximation $\tilde{y}^{(m)}_{101}(t)$ of the trend of $y_{101}(t)$, the first order short-term trend $\tilde{y}^{(m,1)}_{101}(t)$, and the modified predictions $\hat{y}^{d(2,m)}_{101}(t)$ for $m = 0, 1, 2$.

of m may not be so important for this time series, while it seems important for other some time series.

IV. Conclusion

We have described the method which we have used for the time series forecasting competition. The method uses the first-order difference for dealing with the seasonality of the time series. We have developed a method to estimate the trend of time series since the differencing neglects the trend of the time series. We have shown the CAN2 for learning efficient piecewise linear approximation of nonlinear function, and the bagging of the CAN2 for reducing the variance of the prediction by the single CAN2.

Acknowledgment

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] T. Kohonen, *Associative Memory*, Springer Verlag, 1977.

[2] 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.

[3] 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.

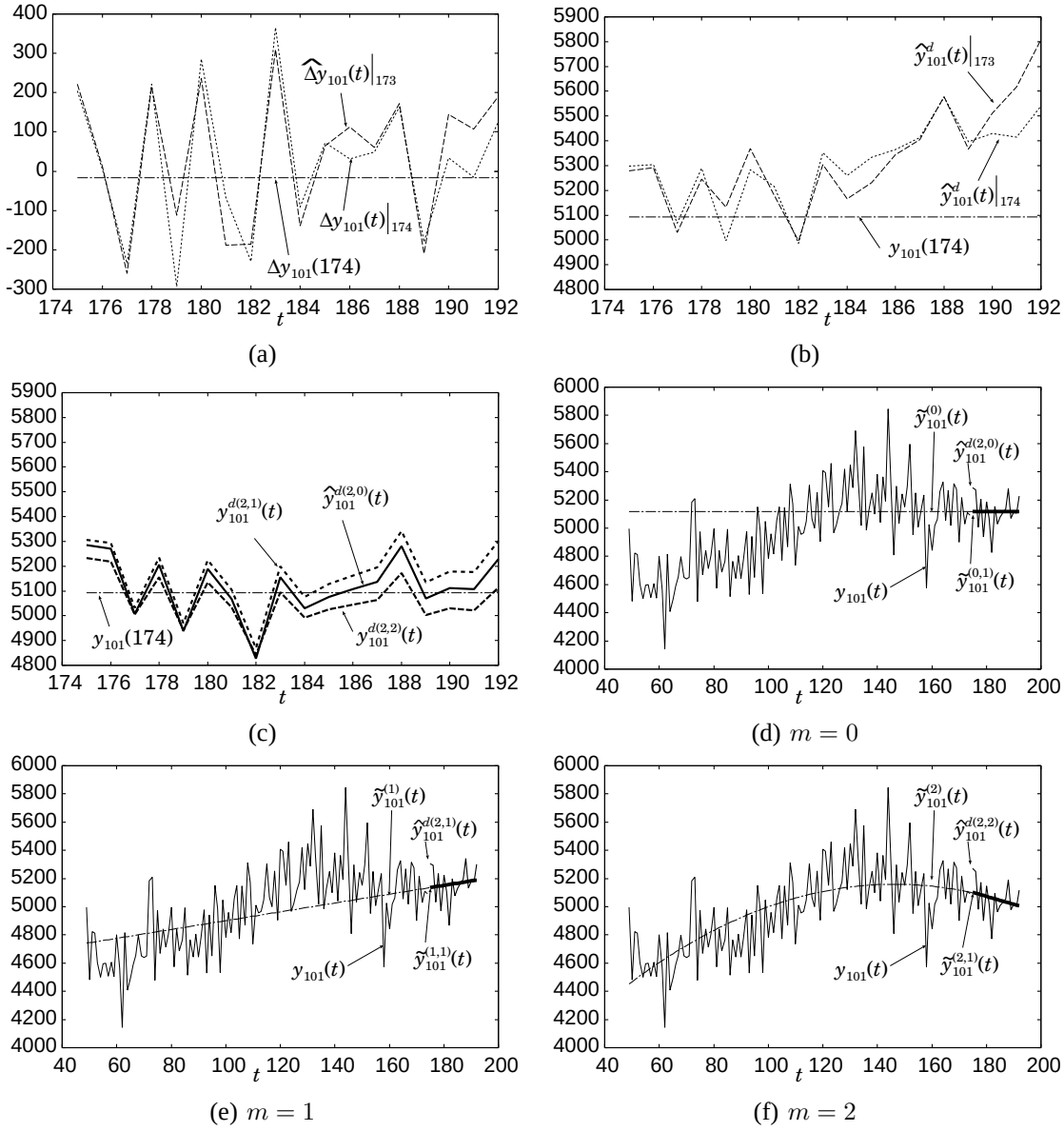

Fig. 3. Example of predictions using the first-order difference for the forecasting period $T_{101}^{forecast} = I[175, 192]$.

[4] S. Kurogi, "Asymptotic optimality of competitive associative nets for their learning in function approximation," *Proc. of ICONIP2002*, vol. 1, pp. 507–511, 2002.

[5] S.Kurogi, "Asymptotic optimality of competitive associative nets and its application to incremental learning of nonlinear functions," *IEICE D-II (in Japanese)*, vol. J86-D-II-2, pp. 184–194, 2003.

[6] S. Kurogi, T. Ueno and M. Sawa, "A batch learning method for competitive associative net and its application to function approximation," *Proc. of SCI2004*, vol. V, pp. 24–28, 2004.

[7] S. Kurogi, T. Ueno and M. Sawa, "Batch learning competitive associative net and its application to time series prediction," *Proc. of IJCNN 2004*, CD-ROM, 25-29 July 2004.

[8] S. Kurogi, S. Tanaka and R. Koyama, "Combining the predictions of a time-series and the first-order difference using bagging of competitive associative nets," *Proc. of ESTSP 2007*, pp. 123–131, 2007

[9] J.D.Farmer and J.J.Sidorowich, "Predicting chaotic time series," *Phys. Rev.Lett.*, vol. 59, pp. 845–848, 1987.

[10] J.H.Friedman, "Multivariate adaptive regression splines," *Ann Stat*, vol. 19, pp. 1–50, 1991.

[11] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24,pp. 123–140, 1996.

[12] J.G. Carney and Padráig Cunningham, "The NeuralBAG algorithm: Optimizing generalization performance in bagged neural networks," *Proceedings of ESANN'1999*, pp. 21–23, 1999.

[13] M.Dettling, "Bagboosting for tumor classification with gene expression data," *Bioinformatics*, vol. 20(18),pp. 3583–3593, 2004.