Intelligent forecasting for financial time series subject to structural changes

Jae Joon Ahn\textsuperscript{a}, Suk Jun Lee\textsuperscript{a}, Kyong Joo Oh\textsuperscript{a,∗} and Tae Yoon Kim\textsuperscript{b}

\textsuperscript{a}Department of Information and Industrial Engineering, Yonsei University, Seoul, South Korea
\textsuperscript{b}Department of Statistics, Keimyung University, Daegu, South Korea

Abstract. This paper is mainly concerned about intelligent forecasting for financial time series subject to structural changes. For example, it is well known that interest rates are subject to structural changes due to external shocks such as government monetary policy change. Such structural changes usually make prediction harder if they are not properly taken care of. Recently, Oh and Kim (2002a, 2002b) suggested a method that could handle such difficulties efficiently. Their basic idea is to assume that different probabilistic law (and hence different predictor) works for different situations. Their method is termed as two-stage piecewise nonlinear prediction since it is comprised of establishing various situations empirically and then installing a different probabilistic nonlinear law as predictor on each of them. Thus, for its proper prediction functioning, it is essential to identify the law dictating the financial time series presently. In this article we propose and study a mixing approach for better identification of the presently working probabilistic law.

Keywords: Structural change, two-stage prediction, mixing approach, classifier, intelligent forecasting

1. Introduction

Many financial time series tend to experience structural changes after they follow a new different probabilistic law. Interest rates are a typical example of these since they often show structural changes due to external shocks such as government monetary policy change. Of course, such structural changes make predictions harder if they are not properly taken care of [7,13,15]. Based on intelligent approach, Oh and Kim [10,11] proposed two-stage piecewise nonlinear prediction method to handle the structural changes efficiently. Indeed, they propose to establish various situations as groups first and then install a predictor on each of them. In their algorithm, groups and their training data are primarily built on time domain and then refined on state space. For practical implementation of Oh and Kim’s algorithm, the present situation is to be identified as one of the groups given in the algorithm and then the corresponding predictor is to be applied. Thus, it is quite critical to identify the present situation properly for their algorithm to be successful since failing to do so would bring malfunctioning to the entire algorithm easily. The main objective of this article is to suggest an algorithm that improves efficiency of the \textit{between-group classifier} whose basic function is to identify the present situation for proper prediction. In fact, a mixing approach is proposed which installs various between-group classifiers and mixes them by a specified
Once $\hat{f}$ is established, it is applied back to $Z_i$ for $i = 1, \ldots, n-1$ to refine the groups on state space.

2. Two-stage prediction with mixing approach

Oh and Kim [10,11] proposes the two-stage piecewise nonlinear method for prediction of financial time series subject to structural changes. Stage One consists of two steps: (i) construction of groups on time space with change points (ii) training a between-group classifier to refine the groups and their training data on state space for Stage Two. Stage Two is to install a predictor by training it on each refined group. A key idea behind this two-stage piecewise nonlinear prediction is to decompose prediction errors (i.e. within-group errors and between-group errors) to improve efficiency of prediction. Notice that between-group error is handled by between-group classifier, say $\hat{f}$, and within-group error by within-group predictor, say $g$. Therefore, it is important to have an accurate $\hat{f}$ that identifies the group for the prediction for this method to work. Otherwise, the two-stage prediction would easily fail completely. Oh and Kim [10,11] simply employed backpropagation neural network (BPN) for $\hat{f}$ which has its own limitations such as overfitting and local optimum problem. To better the accuracy of $\hat{f}$, we propose mixing approach that employs three different between-group classifiers and then mixes them according to a specified rule to produce a better $\hat{f}$. In fact, we employ discriminant analysis (DA), logistic regression (LR), and BPN for $\hat{f}$ (see Remark 2 below). For mixing rule, voting method (VM), arithmetic mean (AM), and geometric mean (GM) are considered (see Remark 3 below).

For more concrete description of the method, consider a time series $X_1, X_2, \ldots, X_n$. Then the time series is said to have a change-point at $k^*$ if \{($X_1, X_2, \ldots, X_{k^*}$) and ($X_{k^*+1}, X_{k^*+2}, \ldots, X_n$) have common distribution functions $F_1(x)$ and $F_2(x)$ respectively, where $F_1(x) \neq F_2(x)$ for some $x$. Suppose that there are $R$ change points $1 < k_1^* < k_2^* < \cdots < k_R^* < n$, which means that $R+1$ different probabilistic laws are governing $R+1$ different groups for the given time series. These change points are estimated by Pettit test [1], and as a result we have $1 < k_1^* < k_2^* < \cdots < k_R^* < n$ as their estimates. To obtain training data set based on $X_1, X_2, \ldots, X_n$, let $G_i$ denote the group to which $X_i$ belongs. Then

$$G_i = \begin{cases} 
1, & \text{if } 1 < i \leq \hat{k}_1^* \\
2, & \text{if } k_1^* < i \leq k_2^* \\
\vdots & \\
R+1, & \text{if } \hat{k}_R^* < i \leq n.
\end{cases} \quad (1)$$

Suppose that we have input vector $Z_i = (z_{i1}, z_{i2}, \ldots, z_{im})$ and output $G_{i+1}$ for each $i = 1, \ldots, n-1$. Then

$$T_{n-1} = \{(Z_i, G_{i+1}) : i = 1, \ldots, n-1\} \quad (2)$$

forms a desired training data set, which produces a between-group classifier $\hat{f} : R^m \to \{1, 2, \ldots, R+1\}$.

Once $\hat{f}$ is established, it is applied back to $Z_i$ for $i = 1, \ldots, n-1$ to refine the groups on state space,
i.e., to construct a refined training data set as
\[ T_{n-1}^{(1)} = \{ (Z_i, \hat{G}_{i+1}) : \hat{G}_{i+1} = \hat{f}(Z_i), i = 1, \ldots, n-1 \} \]  
(3)
for Stage Two. In fact when \( \hat{f} \) is applied back to input data \( Z_i \), it re-classifies the group to which \( X_{i+1} \) belongs, say \( \hat{G}_{i+1} \), by calculating the (posterior) probability
\[ \Delta(Z_i) = \{ \delta_j(Z_i) = P(G_j|Z_i) : j = 1, \ldots, R + 1 \}. \]  
(4)
Note that \( \Delta(Z_i) \) in Eq. (4) is output of \( \hat{f} \) leading to \( \hat{G}_{i+1} \). To improve the accuracy of \( \hat{f} \), we employ three different classification methods (discriminant analysis \( \hat{f}_D \), logistic regression \( \hat{f}_L \), and BPN \( \hat{f}_N \), respectively) and mix them according to a specified rule to obtain the final \( \hat{f} \). Indeed, when being applied to \( Z_i \) for \( i = 1, \ldots, n-1 \) each classifier trained to \( T_{n-1} \) respectively produces its own output.
\[ \Delta_D(Z_i) = \{ \delta_{D,j}(Z_i) : j = 1, \ldots, R + 1 \} \]
\[ \Delta_L(Z_i) = \{ \delta_{L,j}(Z_i) : j = 1, \ldots, R + 1 \} \]
\[ \Delta_N(Z_i) = \{ \delta_{N,j}(Z_i) : j = 1, \ldots, R + 1 \}, \]  
(5)
One needs a specific rule for mixing the three different sets of outputs in Eq. (5). Thus, if a proper mixing rule is specified, say \( H_j(Z_i) \) for \( j = 1, \ldots, R + 1 \), then we have
\[ \hat{f}(Z_i) = \hat{G}_{i+1} = r \text{ if } \arg \max_{j=1,\ldots,R+1} H_j(Z_i) = r. \]  
(6)
This procedure produces \( T_{n-1}^{(1)} \) in Eq. (3) as desired. Possible candidates for mixing rule \( H_j(Z_i) \) are discussed in Remark 3 below.

To generate prediction at Stage Two, a within-group predictor is to be trained to each group on \( T_{n-1} \) via the following model:
\[ X_{i+1} = \begin{cases} 
g_1(Z_i) + \varepsilon_{1,i+1}, & \text{if } \hat{G}_{i+1} = 1 \\
g_2(Z_i) + \varepsilon_{2,i+1}, & \text{if } \hat{G}_{i+1} = 2 \\
\vdots & \text{if } \hat{G}_{i+1} = R + 1 \\
g_{R+1}(Z_i) + \varepsilon_{R+1,i+1}, & \text{if } \hat{G}_{i+1} = R + 1 
\end{cases} \]  
(7)
where \( g_j(j = 1, 2, \ldots, R + 1 \) is the mapping function between \( X_{i+1} \) and \( Z_i \) assuming \( \hat{G}_{i+1} = j \), and \( \varepsilon_{j,i+1} \) is the iid error. Predictor \( \hat{g}_j \) is obtained from the refined training data set \( \{ (Z_i, \hat{G}_{i+1}) : \hat{G}_{i+1} = \hat{f}(Z_i) = j, i = 1, \ldots, n-1 \} \). Then for prediction at the present time \( t \), the predicted value for \( t+1 \) is given by
\[ \hat{X}_{t+1} = \hat{g}_j(Z_i) \text{ if } \hat{G}_{t+1} = j(j = 1, \ldots, R + 1) \]  
(8)
where \( \hat{G}_{t+1} = \hat{f}(Z_i) = j \).

Assuming there is one change-point (and hence \( R = 1 \)) in the given time series dataset, Figure 1 shows the basic architecture of Stage One and Stage Two.

**Remark 1.** Mathematically and intuitively, it is not hard to argue that mixing approach could register a better efficiency. Indeed, assuming the underlying two groups \( T \) and \( T^c \), it is easy to show that when two between-group classifiers \( C_1 \) and \( C_2 \) are independent,
\[ P(T|C_1 = T) \leq P(T|C_1 = T, C_2 = T) \text{ if } P(C_2 = T|T^c) \leq P(C_2 = T|T) \]  
(9)
where \( P(T|C_1 = T) \) is the conditional posterior probability that the true state is \( T \) given that \( C_1 \) classifies it as \( T \), and other conditional probabilities are defined in a similar fashion. Then, Eq. (9) implies that classification reliability in terms of posterior probability could be improved by employing an additional independent classifier whose correct classification rate exceeds error classification rate. Of course, \( P(C_2 = T|T) - P(C_2 = T|T^c) \) (i.e. the difference between correct classification rate and error classification rate) determines the amount of improved efficiency. Verification of (9) may be done as follows. By Bayesian Theorem and independence of \( C_1 \) and \( C_2 \)

\[
P(T|C_1 = T, C_2 = T) = \frac{P(C_1 = T, C_2 = T|T)P(T)}{P(C_1 = T, C_2 = T|T)P(T) + P(C_1 = T, C_2 = T^c|T)P(T^c)}
\]

\[
= \frac{P(C_1 = T|T)P(C_2 = T|T)P(T) + P(C_1 = T|T^c)P(C_2 = T|T^c)P(T^c)}{P(C_1 = T|T)P(C_2 = T|T)P(T) + P(C_1 = T|T^c)P(C_2 = T|T^c)P(T^c)}
\]

\[
\geq \frac{P(C_1 = T|T)P(C_2 = T|T)P(T) + P(C_1 = T|T^c)P(C_2 = T|T^c)P(T^c)}{P(C_1 = T|T)P(T) + P(C_1 = T|T^c)P(T^c)} = P(T|C_1 = T).
\]

Fig. 1. Training two different predictors on each of two groups.
At the above the inequality holds by the given condition \( P(C_2 = T|T^C) \leq P(C_2 = T|T) \).

**Remark 2.** Our mixing approach employs three classifiers, discriminant analysis (DA), logistic regression (LR), and BPN. DA is a statistical method derived from assuming linearity of the underlying model while BPN is an artificial intelligence (AI) algorithm based on nonlinearity of the underlying model. LR is a statistical model bearing attributes of both nonlinearity and linearity. The three classifiers appear to be a set of reasonable choices because more often than not they tend to behave rather independently since they independently represent the three different classes of classification methods currently available. For technical details of DA, LR and BPN including other classifiers, one may refer to [4,6,14] where \( \delta_{D,j}(Z_i), \delta_{L,j}(Z_i), \) and \( \delta_{N,j}(Z_i) \) in (5) are also derived and explained in detail. Also note that BPN is preferred to other neural network for practical application mainly because of its stability (see also Chatfield [3] and Refenes [20]).

**Remark 3.** The considered possible mixing rules are voting method (VM), arithmetic mean (AM), and geometric mean (GM). VM is given by

\[
H_{VM,j}(Z_i) = \frac{\eta[\Delta_{N,j}(Z_i)] + \eta[\Delta_{D,j}(Z_i)] + \eta[\Delta_{L,j}(Z_i)]}{3} \quad (10)
\]

where \( \eta(\Delta_j(Z_i)) = 1 \) if \( \arg \max_{k=1,\ldots,R+1} \delta_k(Z_i) = j \) and 0 otherwise. AM and GM are respectively given by

\[
H_{AM,j}(Z_i) = \frac{\delta_{N,j}(Z_i) + \delta_{D,j}(Z_i) + \delta_{L,j}(Z_i)}{3} \quad (11)
\]

\[
H_{GM,j}(Z_i) = \sqrt[3]{\delta_{N,j}(Z_i)\delta_{D,j}(Z_i)\delta_{L,j}(Z_i)}. \quad (12)
\]

Then, the final output of mixing classifier is yielded by the rule at (6). Note that these mixing rules are practiced in empirical studies of Section 3. One may refer to [32] for their technical details. To explain the motivation behind the choice of the above mixing rules, consider the case where \( \delta_{D,j}(Z_i) = 0.9, \delta_{L,j}(Z_i) = 0.4, \) and \( \delta_{N,j}(Z_i) = 0.4. \) This case is rather hard to make a clear cut decision since it is likely to produce an inconsistent classification result. Using the mixing rules above, we have \( H_{VM,j}(Z_i) = 0.33, H_{AM,j}(Z_i) = 0.57, H_{GM,j}(Z_i) = 0.53 \) and hence in the order of AM, GM, VM the mixing rule favors the class \( j \) given \( Z_i. \) Now it is easy to see that VM ignores classifiers LR and BPN completely, AM takes both into account seriously and GM acts in-between (recall that arithmetic mean is always greater than or equal to geometric mean). In fact, our empirical study of Section 3 verifies that GM could be an optimal choice in real situation. For advantages or disadvantages of other various mixing rules, one may refer to section 4 of [32] where it is claimed that GM performs best when the individual posterior probabilities are estimated correctly (see “Product Rules” on p. 37 there).

**Remark 4.** The mixing approach is a computationally intensive method since training various between-group classifiers requires much time and space computationally. Thus, it is necessary to find an appropriate set of classifiers for the mixing approach. In this context, Eq. (9) might be of help because it provides two criteria, independence and accuracy, for selection of a specific classifier. Indeed, in practice, one may remove the classifiers that are dependent. Note that independence of classifiers may be checked by comparing \( T_n^{(1)}(T^{(1)} \mid Z_i) \) that is yielded by each classifier. In other words, if the between-group classifiers
produce similar $T_{n-1}^{(1)}$'s, then it indicates possible dependency among them which signifies that mixing approach would not be beneficial. Refer to Eq. (15) of Section 3 for this issue.

**Remark 5.** Our mixing approach could be considered as a mixture of boosting and a special case of ensemble method. It is boosting in the sense that different predictor is trained on different group to enhance accuracy, and it is a special case of ensemble method since it combines different models (i.e. DL, LR BPN) to obtain additional diversity of information. See also [32]. Note that our mixing approach is proposed and employed here mainly because it is hard to add correct diversity for time series data (e.g., interest rates) through usual re-sampling schemes such as boosting or bagging. See also [33].

### 3. Empirical studies

Over the past several decades, statistical models have been used extensively to trace the interest rates with quality accuracy. When they are applied to forecasting, however, numerous studies underscore their limitations due to difficulties caused by structural changes [7,13,15,24,25]. Recently, several studies point out that AI approaches, such as fuzzy theory [26] and neural networks [27], can better the predictability of the interest rates [16]. In the meantime, most of the prediction algorithms introduced so far tend to fix on sole use of analysis tool whether it is a conventional statistical method or an AI technique. In this section two sets of empirical studies are carried out, i.e., prediction of the US Treasury bill rates with 3 month’s maturity (3 month T-bills) and prediction of the US Treasury bill rates with 30 year’s maturity (30 year T-bills). The former demonstrates that synergy or mixing effect may be created by employing mixing approach while the latter reports the case where the mixing approach fails to function. In fact, these two examples together not only prove validity of (9) but also provide a useful guideline for employing mixing approach. Note that the US T-bill rates are well known to be subject to structural changes [5,12,22,23].

The first empirical experiment data set is the 3 month T-bill rates from Jan., 1961 to May, 1999 (see Fig. 2(a)). Its training dataset runs from January 1961 to December 1986 and testing dataset from January 1987 to May 1999. The second empirical experiment data set is the 30 year T-bill rates from Jan., 1977 to Dec, 1991 (see Fig. 2(b)). Its training dataset runs from January 1977 to December 1986 and testing dataset from February 1987 to May 1999. For our empirical experiments, input variable $Z_i = (z_1, \cdots, z_m)$ is built on the basis of Fisher’s theory that nominal interest rates are comprised of expected real interest rates and anticipated inflation [8,9,18,28]:

\[
\text{Nominal Interest Rates} = \text{Expected Real Interest Rates} + \text{Anticipated Inflation}
\]

Since it is also well known that money supplies and real GNP growth have a sizable impact on interest rates [17], input variables selected are change rate of consumer price index (CPI), difference between the nominal interest rates and the anticipated inflation (ERIR), M2, and industrial production index (IPI) which represent anticipated inflation, expected real interest rates, money supplies and real GNP growth, respectively. The list of input variables is given in Table 1. Note that the change rate at time $i + 1$ in the data set means

\[
r_{i+1} = \frac{y_{i+1} - y_i}{y_i}
\]

where \(\{y_i : i = 1, 2, \ldots\}\) is a given sequence of rates.
In order to measure synergy effect or mixing effect, seven prediction algorithms are experimented and their performances are compared. While one algorithm, labeled pure NN, implements a single predictor (i.e., training predictor \( \hat{g} \) without establishing \( \hat{f} \)), the other algorithms employ two-stage forecasting.
Table 1
Description of input variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBILL</td>
<td>US T-bills</td>
<td>Output</td>
</tr>
<tr>
<td>M2</td>
<td>Money stock</td>
<td>Input</td>
</tr>
<tr>
<td>CPI</td>
<td>Consumer price index</td>
<td>Input</td>
</tr>
<tr>
<td>ERIK</td>
<td>Expected real interest rates</td>
<td>Input</td>
</tr>
<tr>
<td>IPI</td>
<td>Industrial production index</td>
<td>Input</td>
</tr>
</tbody>
</table>

Table 2
Results of US 3 month T-bills

(a) Numerical evaluations of the seven algorithms for US 3 month T-bills via metrics in (14)

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>MAE</th>
<th>MAPE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure_NN</td>
<td>0.3775</td>
<td>0.3178</td>
<td>6.493</td>
</tr>
<tr>
<td>DA_NN</td>
<td>0.4346</td>
<td>0.3289</td>
<td>6.066</td>
</tr>
<tr>
<td>LR_NN</td>
<td>0.4044</td>
<td>0.3098</td>
<td>5.744</td>
</tr>
<tr>
<td>BPN_NN</td>
<td>0.3948</td>
<td>0.3024</td>
<td>5.615</td>
</tr>
<tr>
<td>VM_NN</td>
<td>0.3992</td>
<td>0.3036</td>
<td>5.679</td>
</tr>
<tr>
<td>AM_NN</td>
<td>0.3465</td>
<td>0.2696</td>
<td>5.326</td>
</tr>
<tr>
<td>GM_NN</td>
<td>0.3242</td>
<td>0.2449</td>
<td>4.741</td>
</tr>
</tbody>
</table>

(b) Pairwise t-test results for the 7 algorithms

<table>
<thead>
<tr>
<th></th>
<th>DA_NN</th>
<th>LR_NN</th>
<th>VM_NN</th>
<th>BPN_NN</th>
<th>AM_NN</th>
<th>GM_NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure_NN</td>
<td>0.817</td>
<td>1.561</td>
<td>1.592</td>
<td>1.794*</td>
<td>3.342***</td>
<td>5.425***</td>
</tr>
<tr>
<td>DA_NN</td>
<td>4.147***</td>
<td>4.118***</td>
<td>4.511***</td>
<td>2.548**</td>
<td>3.684***</td>
<td></td>
</tr>
<tr>
<td>LR_NN</td>
<td>0.671</td>
<td>1.648</td>
<td>1.701*</td>
<td>3.175***</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM_NN</td>
<td>1.183</td>
<td>1.245</td>
<td>2.766***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPN_NN</td>
<td>1.069</td>
<td>2.661***</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AM_NN</td>
<td>2.614***</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GM_NN</td>
<td>2.614***</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

***Significant at 1%; **Significant at 5%; *Significant at 10%.

which involves training both \( \hat{f} \) and \( \hat{g} \). Indeed VM_NN, AM_NN, and GM_NN are to mix outputs of the three between-group classifiers and the algorithms based on sole use of BPN, LR, or DA for \( \hat{f} \) are labeled as BPN_NN, LR_NN, DA_NN, respectively. Note that these seven algorithms are applied to the two sets of data separately.

For the 3-month T-bill rates data set, assuming there is one change-point (and hence two groups on time domain) in the training dataset, Pettitt test is applied once. The change-point is detected at February 1973, i.e., the first period runs from January 1961 to January 1973 and the second period from February 1973 to December 1986. When one looks at their means (4.44 versus 8.31) and variances (1.77 versus 8.31), it is easy to see that group 1 is characterized as low and stable while group 2 as high and varying. To compare the performances of the 7 algorithms, the actual values of the interest rates and the corresponding predicted values by each algorithm are shown for the testing period in Fig. 3. It is easy to see from Fig. 3 that Pure_NN performs poorly on the average while GM_NN and AM_NN reports improved performances over Pure_NN especially on the circled regions A and B in Fig. 3.

Numerical evaluations of the seven algorithms are given in Table 2(a), using various metrics

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (p_t - o_t)^2}, \quad \text{MAE} = \frac{1}{n} \sum_{t=1}^{n} |p_t - o_t|, \quad \text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \frac{|p_t - o_t|}{o_t} \cdot 100 \quad (14)
\]

where \( p_t \) is a predicted value and \( o_t \) is an observed value at \( t \). Since scaling is quite important for
Fig. 3. Plot (a) shows the actual vs. predicted values for testing period of US 3 month T-bills and plots (b) and (c) magnify the circled regions A and B in (a).
Table 3
Results of US 30 year T-bills

(a) Numerical evaluations of the seven algorithms for US 30 year T-bills via metrics in (14)

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>MAE</th>
<th>MAPE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure_NN</td>
<td>0.4198</td>
<td>0.5564</td>
<td>9.437</td>
</tr>
<tr>
<td>DA_NN</td>
<td>0.1020</td>
<td>0.2611</td>
<td>4.405</td>
</tr>
<tr>
<td>LR_NN</td>
<td>0.1066</td>
<td>0.2639</td>
<td>4.458</td>
</tr>
<tr>
<td>BPN_NN</td>
<td>0.1236</td>
<td>0.2871</td>
<td>4.869</td>
</tr>
<tr>
<td>VM_NN</td>
<td>0.1020</td>
<td>0.2611</td>
<td>4.405</td>
</tr>
<tr>
<td>AM_NN</td>
<td>0.1135</td>
<td>0.2778</td>
<td>4.686</td>
</tr>
<tr>
<td>GM_NN</td>
<td>0.1280</td>
<td>0.2930</td>
<td>4.956</td>
</tr>
</tbody>
</table>

(b) Pairwise t-test results for the 7 algorithms

<table>
<thead>
<tr>
<th></th>
<th>DA_NN</th>
<th>LR_NN</th>
<th>VM_NN</th>
<th>BPN_NN</th>
<th>AM_NN</th>
<th>GM_NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure_NN</td>
<td>9.77***</td>
<td>9.72***</td>
<td>9.77***</td>
<td>10.33***</td>
<td>9.76***</td>
<td>9.95***</td>
</tr>
<tr>
<td>DA_NN</td>
<td>0.92</td>
<td>–</td>
<td>5.91***</td>
<td>6.51***</td>
<td>6.41***</td>
<td></td>
</tr>
<tr>
<td>LR_NN</td>
<td>0.92</td>
<td>4.27***</td>
<td>3.20***</td>
<td>6.93***</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM_NN</td>
<td>5.91***</td>
<td>6.51***</td>
<td>6.41***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPN_NN</td>
<td>3.23**</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AM_NN</td>
<td></td>
<td></td>
<td>4.38***</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*** Significant at 1%.

measuring error for the interest rates prediction, a sensible comparison metric would be MAPE. Then it is obvious that mixing algorithms, GM_NN and AM_NN (particularly GM_NN), are remarkably superior to the others in Table 2(a). Also refer to Table 2(b) where pairwise t-test results for the hypothesis $H_0$: two classifiers are equivalent in the difference of their means of the absolute errors, e.g., $E|y_i,A - o_i,A| = E|y_i,B - o_i,B|$ for given classifiers A and B – are shown. Significance test result with small p-values means that the two classifiers A and B perform with significantly different level of accuracy, which together with Table 2(a) strongly suggests that the mixing approach improves the performance of the algorithm based on single classifier. The best performance by GM_NN appears due to the facts that GM acts in-between among AM, VM and GM, or that GM performs best when the individual posterior probabilities are estimated correctly, as discussed in Remark 3 above.

For the 30 year T-bill rates data set, we go through the same prediction and evaluation procedure. Indeed assuming there is one change-point in the training period, Pettitt test is applied once. The change-point is detected at February 1986, i.e., the first period runs from January 1977 to January 1986 and the second period from February 1986 to May 1999. When one looks at their means (10.85 versus 8.00) and variances (3.81 versus 0.68), it is easy to see that group 1 is characterized as high and varying while group 2 as low and stable. In Fig. 4, the actual values of the interest rates and the corresponding predicted values by each of 7 algorithms are shown for the testing period. It is easy to see from Fig. 4 that Pure_NN performs poorly on the average and the mixing algorithms (GM_NN, AM_NN, VM_NN) fail to report improved performances especially on the circled regions C and D. As done in Table 2, numerical evaluations of the seven algorithms are given in Table 3, which shows ineffectiveness of the mixing algorithms overall. Note that Table 3 suggests that the best performer VM_NN is almost equivalent to DA_NN.

Empirical studies in this section confirms Eq. (9) of Remark 1. Especially, it is to be noted that independence of classifiers play a critical role for effective implementation of mixing approach. Indeed, improved performance of mixing approach for 3 month T-bill rates data and its poor performance for 30 year T-bill rates data are certainly due to the independence among classifiers. As mentioned in Remark 1,
Fig. 4. Plot (a) shows the actual vs. predicted values for testing period of US 30 year T-bills data, and plots (b) and (c) magnify the circled regions A and B in (a).
this might be checked by similarity among $T_{n-1,N}^{(1)}, T_{n-1,D}^{(1)}$, and $T_{n-1,L}^{(1)}$. For this, we have calculated the following rate:

$$\lambda_{D,N,L} = \frac{\text{the number of inconsistent data among } T_{n-1,N}^{(1)}, T_{n-1,D}^{(1)}, \text{and } T_{n-1,L}^{(1)}}{\text{the total number of } T_{n-1}^{(1)}}$$

(15)

which is $15/215 = 6.97\%$ for 30 year T-bill rates and $48/311 = 15.43\%$ for 3 month T-bill rates. This shows that the level of independence possessed by the three classifiers for 3 month T-bill rates exceeds the level of independence possessed by the three classifiers for 30 year T-bill rates.

4. Concluding remarks

This paper proposes and studies a mixing approach to improve two-stage piecewise nonlinear prediction method suggested by Oh and Kim [10,11] for prediction of time series subject to structural changes. It is shown that mixing various between-group classifiers may enhance predictability of interest rates significantly if the mixed classifiers perform independently with quality accuracy. For empirical experiments, interest rates prediction is considered and seven algorithms which employ different mixing rules are compared. Empirical experiments are done for two different data sets, which clearly demonstrate that the independence among classifiers (and hence type of data) plays a significant role for achieving the desired mixing or synergy effect of mixing algorithm. For independence check among classifiers we proposed $\lambda$.

Acknowledgment

T.Y. Kim’s research is supported by KRF C00045 in 2007.

References