Several novel evaluation measures for rank-based ensemble pruning with applications to time series prediction

Zhongchen Ma, Qun Dai *, Ningzhong Liu
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Abstract

Ensemble pruning is a desirable and popular method to overcome the deficiency of high computational costs of traditional ensemble learning techniques. Among various of ensemble pruning methods, rank-based pruning is conceptually the simplest and possesses performance advantage. While four evaluation measures for rank-based ensemble pruning specifically for time series prediction are proposed by us in this paper. The first one, i.e. Complementarity measure for time series prediction (ComTSP), is properly modified from Complementarity measure (COM) for classification. The design idea of ComTSP is, if the error made by the subensemble for a pruning sample is larger than that by the candidate predictor to a certain extent, it is assumed that the predictor is complementary to the subensemble. And the predictor which minimizes the error rate of subensemble on the pruning set will be selected at each selection step. The second one, i.e. Concurrency thinning for time series prediction (ConTSP), is correctly transformed from Concurrency measure (CON) for classification. With ConTSP, a predictor is rewarded for obtaining a good performance, and rewarded more for obtaining a good performance when the subensemble performs badly. A predictor is penalized when both the subensemble and itself perform poorly. The measure ReTSP-Value is specifically designed for Reduce Error (RE) pruning for time series prediction. However, ReTSP-Value and ComTSP have the same flaw that, they could not guarantee the remaining predictor which supplements the subensemble the most will be selected. The cause of this flaw is that the predictive error in time series prediction is directional. It is not reasonable for these measures to take reducing error as the only goal while ignore the error direction. While our finally proposed measure ReTSP-Trend overcomes this defect, taking into consideration the trend of time series and the direction of forecasting error. It could indeed guarantee that the remaining predictor which supplements the subensemble the most will be selected. The comparison experiments on four benchmark financial time series datasets show that the measure ReTSP-Trend outperforms the other measures, which can remarkably improve the predictive ability and promote the generalization capability of the pruned ensembles for time series forecasting.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Time series can be defined as a set of sequential observations, of a variable of interest, recorded over a predefined period of time. In general, time series involves a subject of research interest in various areas of knowledge such as: economy (stock prices, unemployment rate, and industrial production), epidemiology (rate of cases of an infectious diseases), medicine (electrocardiogram, and electroencephalogram), and meteorology (temperature, wind velocity and pluviometric precipitation) (Neto et al., 2009). Financial time series forecasting is one of the most active areas in time series prediction. A major challenge confronted with speculators, investors and businesses is how to accurately forecast price movements in financial and commodity markets (Abu-Mostafa & Atiya, 1996). While many factors might influence the trend of a stock market, including political events, general economic conditions, and trader's expectations (Abu-Mostafa & Atiya, 1996), and consequently, it is a challenging task to predict the stock market trend, due to its high volatility and noisy environment.

Motivated by that an ensemble of individual predictors usually performs better than a single predictor, lots of papers (Khashi & Bijari, 2012; Kim & Kim, 1997; Lai, Yu, Wang, & Wei, 2006; Qian & Rasheed, 2010) investigate the use of ensemble methods to improve financial time series forecasting performance. However...
an important shortcoming of ensemble methods is that, in many problems of practical interest, many constituent predictors are needed for the ensemble to achieve good generalization performance. While, obviously, large ensembles require more storage spaces and take longer to make predictions (Hernández-Lobato, Martínez-Muñoz, & Suárez, 2011).

Ensemble pruning is a competitive approach to alleviate the above problems of traditional ensemble methods. Moreover, there exists another benefit that, the generalization performance of the pruned ensemble may be even better than the original ensemble consisting of all the given individual learners. This approach has shown to be effective in classification and regression problems. It can achieve better performance than, or nearly the same level of performance as, the entire ensemble in these tasks (Banfield, Hall, Bowyer, & Kegelmeyer, 2005; Caruana et al., 2004; Margineantu & Dietterich, 1997; Martínez-Muñoz & Suárez, 2006; Martínez-Muñoz & Suárez, 2007; Martínez-Munoz & Suárez, 2004; Prodromidis & Stolfo, 2001; Zhou & Tang, 2003). However, there are few works studying the performance of ensemble pruning technique on time series forecasting problems. But it could be foreseen that the ensemble pruning technique could perform well on this task. As first of all, ensemble pruning technique could be used to enhance the robustness and accuracy of time series forecasting model. Second, time series forecasting is similar to regression problems, and according to the certification given by Zhou, Wu, and Tang (2002), it can be concluded that pruned ensemble could be better than the original entire ensemble in a regression task.

Ensemble pruning methods could be organized into four categories: rank-based methods, clustering-based methods, optimization-based methods and others which do not fall into any of the previous categories (Tsoumakas, Partalas, & Vlahavas, 2009). Among the four categories of ensemble pruning methods, rank-based methods are conceptually the simplest. They order the remaining models in the original ensemble according to an evaluation measure, and incorporate the model which ranks the first into the selected subensemble at each selection step, and this procedure will be executed iteratively until the size of subensemble reaches the expectation. The main issue among the methods of this category is the evaluation measure used for models ranking. In Partalas, Tsoumakas, Hatzikos, and Vlahavas (2008), the authors found that using just the best single model performs quite well and outperforms most of the ensemble pruning methods, apart from the pruning method using Root-Mean-Square-Error (RMSE) as the evaluation measure, when applied to water quality prediction in their work. It could be found that the measure RMSE used for ensemble pruning in time series prediction task has its design prototype in classification task. It corresponds to the Reduce-Error (RE) pruning measure used in classification task.

Inspired by this discovery, we propose four evaluation measures taken those for ensemble pruning in classification task as prototypes and apply them to financial time series forecasting task. Specifically, our proposed four evaluation measures are: Complementarity measure for time series prediction (ComTSP); Concurrency thinning for time series prediction (ConTSP); and ReTSP-Value and ReTSP-Trend for the RE pruning for time series prediction (ReTSP).

As can be clearly identified, ComTSP is specifically modified for time series prediction task with its prototype being Complementarity measure (COM) for classification task. However, we modify the definition of COM appropriately according to the requirement of time series forecasting problems. If the error made by the subensemble for a specific pruning sample is larger than that by the candidate model for a certain degree, it is assumed that the candidate model is complementary to the subensemble. The candidate predictor which can minimize the error rate of subensemble on the selection dataset will be selected at each selection step. However, we found ComTSP has its inherent defect that it cannot guarantee the most complementary predictor will be selected.

And ConTSP is specifically modified from Concurrency (CON) measure in classification task for time series prediction task. A learner is rewarded for obtaining a good performance, and rewarded more for obtaining a good performance when the subensemble performs badly. A learner is penalized in the event both the subensemble and learner perform badly.

The RE pruning approach cannot be directly applied to time series prediction task, either. Since the way of estimating the predictive error is different in time series prediction task. Actually, there exists several error estimating criterions. Our proposed evaluation measure ReTSP-Value is similar to the evaluation measure Root Mean Square Error (RMSE), which has been used in Partalas et al. (2008). However, the predictive error in time series prediction task is directional. It is not very reasonable to only focus on decreasing the value of forecasting error while ignore its direction. This can be understood as a consideration to the diversity of ensemble, while the ensemble diversity here for time series prediction task is apparently different from that for classification task. The proposed measure ReTSP-Value also has the same defect as ComTSP that, it could not guarantee the remaining learner which supplements the subensemble the most will be selected. While our finally proposed measure ReTSP-Trend overcomes this defect, taking into consideration the direction of forecasting error. It could indeed guarantee that the remaining learner which supplements the subensemble the most will be selected into the subensemble at each selection step.

And another contribution of this work consists in, a smart time window size selection procedure is proposed based on ensemble learning paradigm. Moreover, we carry out investigation to study whether there exists great difference among different sizes of original homogenous ensembles with respect to their predictive performance after pruning. And the homogeneous models of the original ensemble are generated with support vector regression (SVR) learning algorithm.

Experimental results demonstrate that using the measure ReTSP-Trend to order the learners in the ensemble has powerful advantages over the other measures, which could significantly improve the predictive accuracy of the pruned ensembles for time series prediction task. And it is demonstrated through experiments that there did not exist any great differences among different sizes of the original ensemble with respect to their predictive performance after pruning.

The rest of this paper is organized as follows. Section 2 presents a theoretical analysis on ensemble pruning for time series forecasting task. The details of the four proposed measures for rank-based ensemble methods are presented in Section 3. Section 4 gives a presentation of the data-driven time window size selection procedure based on ensemble pruning methods. Section 5 takes a review of support vector regression (SVR) learning algorithm. Section 6 describes the setup of empirical study and Section 7 reports and discusses the experimental results. Finally, Section 8 summarizes this paper.

2. Theoretical analysis on ensemble pruning for time series forecasting task

If we consider an equidistant sampled time series \( \{ x_t \}_{t=1,...,N} \), we can construct a \( m \)-dimensional state space vector \( \gamma_t \) in the form

\[
\gamma_t = (x_{t-(m-1)}, x_{t-(m-2)}, \ldots, x_t)
\]

\( x_{(t+1)} = f(\gamma_t) \)  

(1)

(2)
where $s$ is called the horizon of prediction, $m$ denotes the time window size, and the function $f: \mathbb{R}^m \rightarrow \mathbb{R}$ is called the fitting function.

We just concentrate on one-step-ahead prediction, i.e., $s = 1$. This problem could be regarded as a function approximation problem. Suppose we use an ensemble comprising $Q$ component models to approximate a function $f: \mathbb{R}^m \rightarrow \mathbb{R}$, and the predictions of the component models are combined through weighted averaging, where a weight $w_i (i = 1, 2, \ldots, Q)$ satisfying both Eqs. (3) and (4) is assigned to the $i$th component model $f_i$:

$$0 \leq w_i \leq 1$$

$$\sum_{i=1}^{Q} w_i = 1$$

The output variable of the ensemble is determined according to Eq. (5)

$$\hat{f} = \sum_{i=1}^{Q} w_i f_i$$

For convenience, here we assume that all the component models have equal weights, i.e.,

$$w_i = 1/Q (i = 1, 2, \ldots, Q)$$

In other words, here we assume that the component predictions are combined via simple averaging. Then Eq. (5) becomes

$$\hat{f} = \frac{1}{Q} \sum_{i=1}^{Q} f_i$$

It is possible to get a smaller yet better ensemble through ensemble pruning. Next, we introduce Zhou et al.’s analyses in the following.

3. Rank-based ensemble pruning methods and the proposed evaluation measures for time series prediction

The objective of investigation about ensemble pruning methods is to design a procedure that can select the subensemble with the lowest generalization error, which has been proven to be a NP-complete problem (Tamon & Xiang, 2000). In order to simplify the search in the space of subensembles, it is assumed that the best subensemble of size $u - 1$ is included in the best subensemble of size $u$ for rank-based pruning methods, and we can construct a sequence of best subensembles of increasing size by including one learner a time (Martínez-Munoz & Suárez, 2004). The learner which should be incorporated into the subensemble has to be decided according to an evaluation measure for rank-based ensemble pruning methods.

It is too simplistic to use the predictive performance of individual models, which cannot achieve satisfying results (Partalas, Tsoumakas, & Vlahavas, 2009). In Kuncheva and Whitaker (2003), the authors have shown in their works that neither the accuracy of the base learners nor their diversity are by themselves sufficient to identify effective ensembles. A good ensemble evaluation measure needs to take into account both accuracy and diversity. In order to identify subensembles with a good generalization performance, it is necessary to take into account the complementarity of the learners (Martínez-Munoz, Hernández-Lobato, & Suárez, 2009).

In this work, some evaluation measures for rank-based ensemble pruning methods for time series prediction task have been proposed by us. These measures would be shown in the following subsections.

Before describing the evaluation measures which we propose in detail, it is useful to introduce some notations. The input of the support vector regression (SVR) learning algorithm consists in a set of instances $Z_{\text{train}} = \{ (x_i, y_i), i = 1, \ldots, N_{\text{train}} \}$. Each instance is characterized by a feature vector $x_i$ and a predicted value $y_i$. The objective of the learning algorithm is to induce from the training dataset $Z_{\text{train}}$, a hypothesis $h(x)$ that predicts the predicted value of a new example characterized by the vector of attributes $x$.

Ensemble methods generate a variety of hypotheses (in training phase) that are pooled to produce a final prediction by either weighted voting or unweighted voting, stacking, or some other combination methodologies (in prediction phase). The result of combining the predictions of the base learners in an ensemble $E = \{ h_i(x) \}_{i=1}^{N_{\text{sel}}}$ using equally weighted voting equals:

$$H_{\text{w}}(X) = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

The evaluation measures for rank-based pruning technique make use of a selection set composed of $N_{\text{sel}}$ labeled examples, i.e. $Z_{\text{test}} = \{ (x_i, y_i), i = 1, \ldots, N_{\text{test}} \}$, to guide the order of aggregation. Base learners that are expected to perform best when combined are aggregated first. From the subensemble $S_{u-1}$ of size $u - 1$, the subensemble $S_u$ of size $u$ is constructed by incorporating a single learner selected from the remaining models in the original ensemble which are not included in $S_{u-1}$.

The three evaluation measures for rank-based pruning technique, i.e. Complementarity (COM), Concurrency (CON) and Reduce Error (RE), were specifically designed for classification task in Martínez-Munoz and Suárez (2004) and Banfield et al. (2005). Clearly, they could not be directly applied to time series prediction task. These measures have their original designing intention for classification task. The measure Complementarity (COM) is designed for the reason of incorporating the classifier that shifts the decision of the subensemble toward the correct classification the furthest. And the motive for designing the measure CON is similar with that for measure COM. While in this work, we make necessary and appropriate transformations to them so as to make...
them better suited for time series prediction task, and we also analyze whether the designing intention of the transformed measures could remain consistent with that for their original classification task.

3.1. Complementarity measure for time series prediction (ComTSP)

In classification task, the pruning procedure based on measure COM incorporates at each selection step the base learner whose performance is most complementary to that of the current subensemble (Martinez-Munoz & Suárez, 2004). As in RE pruning method, the first classifier incorporated is the one with the lowest error on the selection set \( Z_{sel} \). Subensemble \( S_{u-1} \) is obtained from \( S_{u-1} \) by incorporating the classifier that has the highest classification accuracy on the set of examples that are misclassified by \( S_{u-1} \):

\[
s_u = \arg\max_k \sum_{(x,y) \in Z_{sel}} I(y = h_k(x) \text{ and } H_{S_{u-1}}(x) \neq y)
\]

(15)

where \( h_k \in E \setminus S_{u-1} \), \( I(*) \) is an indicator function (\( I(\text{true}) = 1 \) and \( I(\text{false}) = 0 \)), and \( s_u \) represents the serial number of the currently selected base learner in the original ensemble.

However, we modify Eq. (15) into Eq. (16) appropriately according to the requirement of time series forecasting problems. We set a threshold to judge whether the candidate learner can defeat subensemble \( S_{u-1} \) over an example in the selection dataset \( Z_{sel} \). That is to say, if the error made by the subensemble \( S_{u-1} \) for a specific pruning sample is larger than that by the candidate model for a certain degree, namely, their error difference is bigger than a prespecified threshold, it is assumed that the candidate model is complementary to the subensemble. The candidate learner which can decrease the error rates of subensemble \( S_{u-1} \) on selection dataset \( Z_{sel} \) as far as possible will be selected at the current selection step. In this way, subensemble \( S_u \) is obtained from \( S_{u-1} \) by incorporating the base learner that has the highest forecasting accuracy on the set of pruning examples that are underperformed by \( S_{u-1} \):

\[
s_u = \arg\max_k \sum_{(x,y) \in Z_{sel}} I(y - H_{S_{u-1}}(x) > y - h_k(x) > \text{threshold})
\]

(16)

where \( h_k \in E \setminus S_{u-1} \), \( \text{threshold} \) is a prespecified constant, \( I(*) \) is an indicator function (\( I(\text{true}) = 1 \) and \( I(\text{false}) = 0 \)), and \( s_u \) represents the serial number of the currently selected base learner in the original ensemble.

In Eq. (15), the quantity maximized can be thought as the amount by which the classifier under considered shifts the decision of the ensemble toward the correct classification. However, Eq. (16) cannot guarantee to incorporate at each iteration the learner whose performance is most complementary to that of the selected subensemble. But the learner is complementary to that of the selected subensemble to a certain degree. We use an example to illustrate it.

Suppose we have got a subensemble: SE, two learners: \( h_1, h_2 \). And there is only one example \( (x,y) \) in the selection dataset. We use the simple mean as the integration method. The values which were predicted for this example by \( h_1, h_2 \), SE are respectively \( \rho_1, \rho_2 \) and \( \rho_{se} \). And they meet:

\[
y - \rho_1 > 0, \quad y - \rho_{se} > 0, \quad y - \rho_2 < 0
\]

(17)

\[
|y - \rho_{se}| - |y - \rho_1| > \text{threshold}, \quad |y - \rho_{se}| - |y - \rho_2| < \text{threshold}
\]

(18)

According to Eq. (16), the learner \( h_1 \) should be incorporated to the selected subensemble. However, it is clear that the performance of learner \( h_2 \) is more complementary to that of the selected subensemble rather than learner \( h_1 \) in the case of Eq. (19).

\[
2y - (\rho_1 + \rho_{se}) > |2y - (\rho_2 + \rho_{se})|
\]

(19)

Obviously, the learner \( h_2 \) should be incorporated to the selected subensemble rather than learner \( h_1 \) and the case shown in Eq. (19) exists indeed in practical applications.

The example shows that the measure ComTSP proposed by us has its inherent drawback. However, this problem will be solved with another evaluation measure, i.e. RetSP-Trend, which will be proposed by us in Section 3.3.

3.2. Concurrency thinning for time series prediction (ConTSP)

The measure CON was proposed in Banfield et al. (2005) for classification task based on the performance of both the subensemble and the candidate learner with regard to a selection set \( Z_{sel} \). With the measure CON, a candidate classifier is rewarded for obtaining a correct decision, and rewarded more for obtaining a correct decision when the subensemble is incorrect. A candidate classifier is penalized in the event where both the subensemble and the candidate classifier are incorrect. The subensemble \( S_u \) is obtained by incorporating the base learner that has the highest value of measure CON on the selection dataset \( Z_{sel} \):

\[
s_u = \arg\max_k \text{CON}_k
\]

(20)

where \( s_u \) represents the serial number of the currently selected base learner in the original ensemble. The measure CON obtained by candidate learner \( h_k \) can be calculated as follows:

\[
\text{CON}_k = \sum_{(x,y)\in Z_{sel}} (2I(y=H_{S_{u-1}}(x) \text{ and } y = h_k(x)) + I(y = H_{S_{u-1}}(x) \text{ and } y = h_k(x)) - 2I(y=H_{S_{u-1}}(x) \text{ and } y=\hat{h}_k(x)))
\]

(21)

where \( h_k \in E \setminus S_{u-1} \), \( \text{threshold} \) is a prespecified constant, and \( I(*) \) is an indicator function (\( I(\text{true}) = 1 \) and \( I(\text{false}) = 0 \)).

For a classification task, it is doubtless to judge whether a classifier makes a correct or incorrect decision. However, the procedure can be applied to time series prediction task by reasonable transformation. More specifically, a learner is rewarded for obtaining a good performance, and rewarded more for obtaining a good performance when the subensemble performs badly. A learner is penalized in the event both the subensemble and learner perform badly. We use a threshold to judge whether a learner performing well or performing bad. In this way, subensemble \( S_u \) is obtained by incorporating the base learner that has the highest value of measure ConTSP on the selection dataset \( Z_{sel} \):

\[
s_u = \arg\max_k \text{ConTSP}_k
\]

(22)

where \( s_u \) represents the serial number of the currently selected base learner in the original ensemble. The measure ConTSP obtained by candidate learner \( h_k \) can be calculated as follows:

\[
\text{ConTSP}_k = \sum_{(x,y)\in Z_{sel}} (2I(y = H_{S_{u-1}}(x) > \text{threshold}) + I(y = H_{S_{u-1}}(x) < \text{threshold}) - 2I(y = H_{S_{u-1}}(x) > \text{threshold}) + I(y = h_k(x) > \text{threshold}) + I(y = h_k(x) < \text{threshold}) - 2I(y = h_k(x) > \text{threshold}))
\]

(23)

where \( h_k \in E \setminus S_{u-1} \), \( \text{threshold} \) is a prespecified constant, and \( I(*) \) is an indicator function (\( I(\text{true}) = 1 \) and \( I(\text{false}) = 0 \)).

3.3. Reduce Error (RE) pruning for time series prediction (RetSP)

The evaluation measure RE was proposed by Margineantu and Dietterich for classification task initially in Margineantu and Dietterich (1997). The classifier incorporated into the subensemble at each selection step is selected by estimating its classification error on the selection dataset. The RE procedure is greedy,
therefore, after obtaining the resultant top $T$ individual learners, Margineantu and Dietterich used backfitting search to improve the ensemble. However, backfitting is time-consuming and it could not improve the generalization ability significantly for parallel ensemble generation methods. So, the authors of Martínez-Muoz et al. (2009) abandoned the backfitting procedure for RE in their investigation.

The RE pruning procedure is simple in theory and in practice for classification task. However, it cannot be directly employed to time series prediction task. The way to estimate the classification error is doubtless in classification task. However, it is different to estimate the predictive error in time series prediction task. Actually, there exists several error estimating criterions in time series prediction. To choose an appropriate evaluation measure among them, an in-depth analysis for this method is required, along with which the detail of our proposed two evaluation measure for Reduce Error pruning for time series prediction, i.e. RetSP-Value and RetSP-Trend, is given as below.

Given a selection set $Z_{sel}$ of size $N_{sel}$, the signature vector $c^{(i)}$ of predictor $i$ is defined as the $N_{sel}$ dimensional vector whose $i$th component is

$$
    c^{(i)} = \langle h_i(x_i) - y_i \rangle^2, \quad (x_i, y_i) \in Z_{sel}
$$

The $i$th component $c^{(i)}$ is equal to 0 if the forecasting value of the $i$th predictor exactly matches to the $i$th example in $Z_{sel}$, and it denotes the squared error made by the $i$th predictor for the $i$th example in $Z_{sel}$. The ensemble signature vector $c_{en}$ is defined as the sum of the signature vectors of all the predictors in the ensemble. And the average ensemble signature vector is defined as:

$$
    \langle c \rangle = T^{-1} \sum_{i=1}^{T} c^{(i)}
$$

(25)

In time series prediction task, the $i$th component of $\langle c \rangle$ is the margin of the $i$th example, defined as the accuracy of the ensemble prediction for this example. The $i$th example is fitted better by the ensemble if the $i$th component of $\langle c \rangle$ is smaller. In consequence, the objective is to select a subensemble whose average signature vector is as close as possible to the origin $o$ of coordinates.

The first predictor that is incorporated into the subensemble is the one that reduces the distance from vector $\langle c \rangle$ to $o$ the most. In particular, the predictor selected in the $u$th selection step is indexed with:

$$
    s_u = \arg\min_k \left( o, T^{-1} \left( c^{(k)} + \sum_{i=1}^{u-1} c^{(i)} \right) \right)
$$

(26)

where $h_k \in E_i/S_{u-1}$, $d(u, v)$ denotes the usual Euclidean distance between points $u$ and $v$, and $s_u$ represents the serial number of the currently selected base learner in the original ensemble.

This evaluation measure is similar to the evaluation measure Root Mean Square Error (RMSE), which has been used in Partalas et al. (2008). In their work, it can obtain the best performance despite its simplicity compared to other greedy ensemble selection methods for regression task. The measure RMSE represents the size of error for prediction values. It could be used for classification task for granted, in the reason of the discreteness of class label. However, for time series forecasting task, the predictive error can be positive or negative, namely, the predictive error in time series prediction task is directional. Therefore, it is not very reasonable to only focus on decreasing the value of forecasting error while ignoring its direction. This can also be understood as a consideration to the diversity of ensemble, while the ensemble diversity here for time series prediction task is apparently different from that for classification task. Inspired by the above analysis, we re-define the signature vector $c^{(i)}$ of predictor $i$ as the $N_{sel}$ dimensional vector whose $i$th component equals:

$$
    c^{(i)} = \langle h_i(x_i) - y_i \rangle, \quad (x_i, y_i) \in Z_{sel}
$$

(27)

This method starts with selecting into the empty subensemble the individual learner whose predictive performance on the selection set is the best. Then, the remaining individual learners are sequentially put into the subensemble, such that the Euclidean distance from vector $c$ to $o$ is as small as possible in each selection round. The signature vector $c^{(i)}$ defined by Eq. (27) could guarantee that the remaining learner which supplements the subensemble the most would be selected into it. We use an example to illustrate it in the following.

Suppose we have got a subensemble: SE, two base learners: $h_1$, $h_2$. And there is only one example $(x, y)$ in the selection dataset. We use simple mean as the integration method. The values which were predicted for this example $(x, y)$ by $h_1$, $h_2$ and SE are respectively $\rho_1$, $\rho_2$ and $\rho_{SE}$. And they meet:

$$
    y - \rho_1 > 0, \quad y - \rho_2 > 0, \quad y - \rho_{SE} < 0
$$

(28)

$$
    \left| y - \left( \frac{\rho_1 + \rho_2}{2} \right) \right| > \left| y - \left( \frac{\rho_2 + \rho_{SE}}{2} \right) \right|
$$

(29)

It is obvious that the performance of learner $h_2$ is more complementary to that of the subensemble than learner $h_1$. However, if the signature vector $c^{(i)}$ of predictor $i$ is constructed according to Eq. (24), learner $h_1$ will be incorporated into the subensemble in the reason of Eq. (30).

$$
    d \left( \langle o, T^{-1} \left( (\rho_{SE} - y)^2 + (\rho_1 - y)^2 \right) \rangle \right) < d \left( \langle o, T^{-1} \left( (\rho_{SE} - y)^2 + (\rho_2 - y)^2 \right) \rangle \right)
$$

(30)

Therefore, the signature vector construction method of Eq. (24) has its inherent defect. However, the signature vector construction method of Eq. (27) could overcome this defect for the reason of Eq. (31).

$$
    d \left( \langle o, T^{-1} \left( (\rho_{SE} - y) + (\rho_1 - y) \rangle \right \rangle \right) > d \left( \langle o, T^{-1} \left( (\rho_{SE} - y) + (\rho_2 - y) \rangle \right \rangle \right)
$$

(31)

Obviously, according to Eq. (31), learner $h_2$, which is more complementary to the subensemble, will be incorporated into the subensemble rather than learner $h_1$.

The above example indicates that the signature vector $c^{(i)}$ defined by Eq. (24) only considers the magnitude of forecasting error and it is not so reasonable for time series forecasting task. While the signature vector $c^{(i)}$ defined by Eq. (27) takes into consideration the direction of forecasting error besides the magnitude of forecasting error. It could guarantee that the remaining learner which supplements the subensemble the most will be selected into the subensemble. In this paper, we name our proposed evaluation measure RetSP using signature vector defined by Eqs. (24) and (27) as measure RetSP-Value and RetSP-Trend, respectively.

### 4. Smart time window size selection procedure based on ensemble pruning methods

To form the input data, we have to determine how large the time window size (TWS) should be, for the reason that a narrow time window size is doubtless in classification task. However, it is different to estimate the performance of learner $h_2$ is more complementary to that of the subensemble than learner $h_1$. However, if the signature vector $c^{(i)}$ of predictor $i$ is constructed according to Eq. (24), learner $h_1$ will be incorporated into the subensemble in the reason of Eq. (30).

$$
    d \left( \langle o, T^{-1} \left( (\rho_{SE} - y)^2 + (\rho_1 - y)^2 \right) \rangle \right) < d \left( \langle o, T^{-1} \left( (\rho_{SE} - y)^2 + (\rho_2 - y)^2 \right) \rangle \right)
$$

(30)

$$
    d \left( \langle o, T^{-1} \left( (\rho_{SE} - y) + (\rho_1 - y) \rangle \right \rangle \right) > d \left( \langle o, T^{-1} \left( (\rho_{SE} - y) + (\rho_2 - y) \rangle \right \rangle \right)
$$

(31)

Obviously, according to Eq. (31), learner $h_2$, which is more complementary to the subensemble, will be incorporated into the subensemble rather than learner $h_1$.

The above example indicates that the signature vector $c^{(i)}$ defined by Eq. (24) only considers the magnitude of forecasting error and it is not so reasonable for time series forecasting task. While the signature vector $c^{(i)}$ defined by Eq. (27) takes into consideration the direction of forecasting error besides the magnitude of forecasting error. It could guarantee that the remaining learner which supplements the subensemble the most will be selected into the subensemble. In this paper, we name our proposed evaluation measure RetSP using signature vector defined by Eqs. (24) and (27) as measure RetSP-Value and RetSP-Trend, respectively.

To form the input data, we have to determine how large the time window size (TWS) should be, for the reason that a narrow time window could lead to omission of important information, while a wide, useless window may cause interfering noise. Ideally, for a given problem, the size of the time window should be adapted to the context. This can be done by using recurrent neural networks (RNNs), which would be learned by a gradient-based learning algorithm, such as backpropagation through time (BPTT) algorithm (Assaad, Boné, & Cardot, 2008).
However, in this work, we propose a smart way to solve this problem based on ensemble learning paradigm. First, numbers of models are trained by using training datasets obtained with different time window sizes. In other words, we did not use a fixed number of past values to feed into a single model. Instead, we use a number of time window sizes to form different training datasets, based on which numbers of different models could be trained. And then, we select suitable models by employing ensemble pruning method, rather than select suitable time window size for a single model in advance. This method makes the time window size selection procedure no longer necessary. According to this idea, a flow diagram of our proposed ensemble pruning method specifically designed for time series forecasting is illustrated in Fig. 1.

5. Support vector regression learning algorithm

The homogeneous models of the original ensemble used for the research of pruning methods in this work are derived from different executions of support vector regression (SVR) learning algorithm by using different parameter values. Now we give a review of SVR learning algorithm (Basak, Pal, & Patranabis, 2007).

Given a training set of $N$ examples $(x_i, y_i)$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, where $\mathbb{R}^d$ denotes the space of the input samples. The aim of regression is to find a function which can not only approximate the training data well, but also can accurately predict the value of $y$ for future data $x$.

In general, the approximating function in SVR takes the following linear form:

$$ f(x) = w^T x + b $$

(32)

where $w \in \mathbb{R}^d$, $b \in \mathbb{R}$, $(x, y)$ is an example in training set and $f(x)$ is fitted value for $x$.

Now the question is to determine $w$ and $b$ from the training data by minimizing the regression risk, $R_{\text{reg}}$, which is defined as

$$ R_{\text{reg}}(f) = \Omega[f] + C \sum_{i=1}^{N} \Gamma(y_i - f(x_i))^2 $$

(33)

where $\Omega[f]$ is the structure risk used to control the smoothness or complexity of the function, $\Gamma(*)$ is a loss function that measures the empirical risk, and $C$ is a pre-specified trade-off value. Generally, in SVR, $\Omega[f]$ takes the form of $||w||_1$ in $\ell_1$ – SVR or $\frac{1}{2} w^T w$ in $\ell_2$ – SVR.

The empirical loss function adopts the form of an $\epsilon$ insensitive loss function, which is defined as follows:

$$ \Gamma_\epsilon(y_i - f(x_i)) = \begin{cases} 0, & \text{if } |y_i - f(x_i)| < \epsilon \\ |y_i - f(x_i)| - \epsilon, & \text{otherwise} \end{cases} $$

(34)

The complete optimization of SVR (or more precisely, the optimization of $\ell_2$ – SVR) can be written as follows:

$$ \min_{w, b, \xi_1, \xi_2} \frac{1}{2} w^T w + C \sum_{i=1}^{N} (\xi_1^i + \xi_2^i) $$

s.t. $y_i - (w^T x_i + b) \leq \epsilon + \xi_1^i$,

$(w^T x_i + b) - y_i \leq \epsilon + \xi_2^i$,

$\xi_1^i \geq 0$, $\xi_2^i \geq 0$, $i = 1, \ldots, N$,

(35)

(36)

(37)

(38)

where $\xi_1^i$ and $\xi_2^i$ are corresponding positive and negative errors at the $i$th point, respectively.

The above optimization problem can be solved by the optimization becomes a quadratic programming problem and furthermore, the linear regression model can be extended into the non-linear one by using Mercer’s kernel. There are many kernel functions in machine learning, we introduce two popular kernel functions, the radial basis function kernel (RBF kernel) and the polynomial kernel. And they are respectively defined as

RBF kernel function: $k(x, x_i) = \exp(-\gamma ||x - x_i||^2)$, $\gamma > 0$

(39)

Polynomial kernel function: $k(x, x_i) = (x^T x_i + r)^d$

(40)

where $d$ is the degree of polynomial kernel function, $r \geq 0$ is a constant trading off the influence of higher-order vs. lower-order terms in the polynomial.

---

**Fig. 1.** A flow diagram of ensemble pruning method for time series forecasting.
6. Experimental setup

In the remaining part of this paper, there are two main issues needed to be addressed: (1) to verify the effectiveness of the proposed evaluation measures, i.e. ComTSP, ConTSP, ReTSP-Value and ReTSP-Trend, on financial time series forecasting; (2) to demonstrate whether there exists great difference among different sizes of original homogenous ensembles with respect to their predicting performance after pruning procedure.

To address these two issues, we implement experiments on four benchmark financial time series datasets. We will describe the four benchmark financial time series datasets and the experimental setup in detail in this section. And we will report the experimental results in the next section.

6.1. Experimental data and data pre-processing

The research data in this study consists of four typical stock indices: (1) Dow Jones Industrial Average (^DJI); (2) GlaxoSmithKline plc (GSK); (3) Hang Seng Index (^HSI); (4) Johnson Outdoors Inc. (JOUT). The historical data are collected daily and are obtained from Yahoo Finance. The entire dataset covers the period from January 1, 1996 to December 31, 2012.

We verified the performance of rank-based pruning methods with $k$-fold cross-validation approach by computing mean values. Yet, time series could be problematic for cross-validation. In the forecasting domain, recent patterns should have higher importance when compared with older ones. When the data are not independent, cross-validation becomes more difficult as leaving out an observation does not remove all the associated information due to the correlations with other observations. An approach that is sometimes more principled for time series is forward chaining (stats.stackexchange.com). Therefore, the similar 5-fold cross-validation approach as adopted in Assaad et al. (2008) is also used here, where the procedure would be something like this:

For each fold of cross-validation, the entire dataset is divided into three periods. The first period, which is assigned to in-sample estimation, is used for network training, i.e. the training set. The second period is reserved for ensemble pruning, i.e. the selection dataset. The third period, which is assigned to out-of-sample evaluation, is used for testing purpose, i.e. the testing set.
Fold 1:

The first period is from January 1, 1990 to December 31, 2010; the second period is from January 1, 2011 to December 31, 2011; while the third period is from January 1, 2012 to December 31, 2012.

Fold 2:

The first period is from January 1, 1990 to December 31, 2009; the second period is from January 1, 2010 to December 31, 2010; while the third period is from January 1, 2011 to December 31, 2011.

Fold 3:

The first period is from January 1, 1990 to December 31, 2008; the second period is from January 1, 2009 to December 31, 2009; while the third period is from January 1, 2010 to December 31, 2010.

Fold 4:

The first period is from January 1, 1990 to December 31, 2007; the second period is from January 1, 2008 to December 31, 2008; while the third period is from January 1, 2009 to December 31, 2009.

Fold 5:

The first period is from January 1, 1990 to December 31, 2006; the second period is from January 1, 2007 to December 31, 2007; while the third period is from January 1, 2008 to December 31, 2008.

There are five attributes in the financial time series:

1. The highest value that the stock was negotiated in a certain day, denoted by $H_t$.
2. The lowest value that the stock was negotiated during the same day, denoted by $L_t$.

Fig. 4. Hang Seng Index (HSI).

Fig. 5. Johnson Outdoors (JOUT).
3. The value of the first negotiation of the day, i.e. opening price, denoted by \( O_t \).
4. The value of the last negotiation of the day, i.e. closing price, denoted by \( C_t \).
5. The business volume of the stock during the same day.

While there is a lot of missing data in the volume series, we just use the first four attributes of the five attributes. Among the four attributes, the closing price is the attribute that is really important, since most of the professional investors and financial institutions take action based on its value (Neto et al., 2009). So our forecasting goal is to forecast the daily close price using its past value and the other three time series attributes. We use a single SVR to illustrate it.

\[
\tilde{C}_t = \text{SVR}(C_{t-1}, C_{t-2}, \ldots, C_{t-p}, H_t, L_t, O_t)
\]  
(41)

where \( \tilde{C}_t \) is the \( t \)th prediction of closing price by SVR learning algorithm. \( C(i = t-1, t-2, \ldots, t-p) \) is the \( i \)th observation of the closing price, \( H_t \) is the \( t \)th observation of the highest price, \( L_t \) is the \( t \)th observation of the lowest price, \( O_t \) is the \( t \)th observation of the opening price, and \( p \) is the time window size. Since the attributes of sample sets have different value scales, it is necessary to adjust the scale of each attribute into the range of [0,1]. It ensures that the larger value input attributes do not overwhelm smaller value inputs, and then helps to reduce prediction errors. We use the following two steps to describe normalization method:

step1: each of the attributes \( d_t \) was converted to logarithms returned as:

\[
r_t = \log(d_t)
\]  
(42)

step2: each of the series value \( r_t \) was normalized by the linear interpolation Eq. (43)

\[
r'_t = \frac{r_t - MIN}{MAX - MIN}
\]  
(43)

where: \( r'_t \) = normalized value; \( r_t \) = value to be normalized; \( MIN \) = minimum value of the series to be normalized; \( MAX \) = maximum value of the series to be normalized.

6.2. Experimental methodology

In order to study the difference of predicting performance for various sizes of homogeneous ensembles, the ensemble models in this work are homogeneously derived by SVR algorithm using different parameters. The SVR algorithm is performed by utilizing the toolbox, Libsvm. And, we use the procedure shown in Fig. 1 to construct ensemble models. After that, the four evaluation measures proposed by us, i.e. ComTSP, ConTSP, ReTSP-Value and ReTSP-Trend, are implemented to order the ensemble models according to their performances on the selection dataset. With that, those top ranking models are selected to construct the pruned subensemble. Finally, predictions are made for the testing dataset by combining the forecasting results of the selected models in the pruned ensemble. For simplicity, the simple-average method is employed here as the combination method. And, the performance measurements introduced in the next section are used to gauge the performance of the pruned ensembles obtained with different evaluation measures.

To investigate the effect produced by different sizes of the original homogeneous ensemble on the pruning performance, we construct four kinds of ensembles with the size of 1600, 800, 400 and 200, respectively. The different sizes of ensemble are constructed based on different internal model parameters of \( \varepsilon - \text{SVR} \) algorithm and different time window sizes. Trade-off (C) and kernels are crucial internal parameters for \( \varepsilon - \text{SVR} \) algorithm. Suppose use \( k_1 \) time window sizes, \( k_2 \) trade-offs, \( k_3 \) kernels, we can construct \((k_1 \times k_2 \times k_3)\) \( \varepsilon \)-SVR models. In this work, we set \( \varepsilon = 0.1 \), and two kernels (polynomial kernels and RBF kernels) are used.

Table 1 describes the values of parameters that we use to train base SVR models within the four kinds of ensembles with the size of 1600, 800, 400, 200, respectively. As demonstrated in Section 4 that, the time window size selection procedure is an important issue for time series prediction task, therefore, we use the smart time window size selection procedure based on ensemble pruning method to choose suitable time window size within a reasonable range \([1:20]\). Because, by trial and error, we find that it is large enough for the value of 20 as the maximum value of time window size. The internal model parameters of \( \varepsilon \)-SVR listed in Table 1 are chosen according to our experience from a mass of experiments using \( \varepsilon \)-SVR learning algorithm as the time series prediction model. Moreover, there are some not very good internal model parameters used in our experiments, aiming at enhancing the diversity of the ensemble system, which are also listed out in Table 1.

With respect to the thresholds for ComTSP and ConTSP, the appropriate thresholds values are obtained with trial and error, as shown in Table 2. Additionally, we implement two baseline methods, corresponding to two extreme pruning scenarios, for comparison purposes. The first one selects the best single model (BSM) in the ensemble, according to the performance of the models on the selection dataset. The other one retains all models of the ensemble (ALL).

6.3. Performance measurement

A performance measurement is necessary to appropriately evaluate the predictive performance of pruned ensemble obtained with different ensemble pruning measures. The performance measurements are defined on the basis of prediction error, which is established as the difference between the real value of the series (target or objective of the prediction) and the predicted value (the output of the ensembles). Therefore, they are presented by the following equation:

\[
e_t = (\text{target}_t - \text{output}_t)
\]  
(44)

where \( \text{target}_t \) is the desired output of the prediction model at time \( t \), and \( \text{output}_t \) is the output of the neural network model at time \( t \). Based on the prediction error, three performance measurements used to evaluate the predictive performance of the pruned ensembles are described below.

6.3.1. RMSE—Root Mean Square Error

Root mean square error (RMSE) is the most common metric used to analyze ensemble performance and it is defined by the equation:

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (\text{target}_t - \text{output}_t)^2}
\]  
(45)

where \( N \) denotes the number of data values of the testing time series. Obviously, the lower the value of RMSE, the better is the result of the prediction. Even though RMSE is quite common as a performance measurement, it does not provide a complete and convincing evidence about the accuracy of the predictive model. Therefore, other two metrics are also used to evaluate the performance of the proposed models.

6.3.2. MAPE—mean absolute percentage error

The measure \( \text{MAPE} \) describes the errors in percentages which is an advantage in relation to the RMSE measure, since it does not depend on the values or the scale of the time series, which simplifies its usage. MAPE is defined as:
The appropriate thresholds of ComTSP and ConTSP obtained with trial and error.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation measure</td>
<td>DJI</td>
</tr>
<tr>
<td>ComTSP</td>
<td>0.0035</td>
</tr>
<tr>
<td>ConTSP</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\text{MAPE} = \frac{1}{N} \sum_{t=1}^{N} \left( \frac{\text{target}_t - \text{output}_t}{\text{output}_t} \right)
\]

Clearly, the lower the value of MAPE, the closer is the desired results from the predicted ones.

### 6.3.3. POCID-prediction on change in direction

The measurement POCID (Neto et al., 2009) demonstrates the percentage of the number of correct decisions when predicting whether the value of the time series will increases or decreases in the next time interval. POCID is defined as:

\[
\text{POCID} = 100 \frac{\sum_{t=1}^{N} D_i}{N}
\]

having the value of \( D_i \) determined by:

\[
D_i = \begin{cases} 
1 & \text{if } (\text{target}_t - \text{target}_{t-1})/\text{output}_{t-1} > 0 \\
0 & \text{otherwise}
\end{cases}
\]

The values that POCID assumes are in between 0 and 100, so that, the closer the values are to 100, the better is the model prediction. This measurement is important when applied to the stock market, because a correct prediction on the direction of the series of the stock quotation affects directly the financial gains and losses on the investment.

### 7. Experimental results and discussions

In the following tables, we present the forecasting performances of BSM, ALL and rank-based pruning methods using ComTSP, ConTSP, ReTSP-Value and ReTSP-Trend as their evaluation measures, respectively. The rank-based ensemble pruning technique could reduce ensemble storage spaces and a small fraction of ensemble models would be selected in the final ensemble. And

<table>
<thead>
<tr>
<th>MAPE</th>
<th>Ensemble size</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0.0577</td>
</tr>
<tr>
<td>400</td>
<td>0.0785</td>
</tr>
<tr>
<td>800</td>
<td>0.1018</td>
</tr>
<tr>
<td>1600</td>
<td>0.1253</td>
</tr>
</tbody>
</table>

In the following tables, we present the forecasting performances of BSM, ALL and rank-based pruning methods using ComTSP, ConTSP, ReTSP-Value and ReTSP-Trend as their evaluation measures, respectively. The rank-based ensemble pruning technique could reduce ensemble storage spaces and a small fraction of ensemble models would be selected in the final ensemble. And
by experiments, we found that when the number of predictors of a pruned ensemble was substantially reduced to 10–40, the ensemble could give a good predicting value. It can be clearly seen from Tables 3–10 that, using the evaluation measure ReTSP-Trend to order the learners in the ensemble has powerful advantages over the other measures, which could significantly improve the predictive accuracy of the pruned ensembles for time series prediction task. And, as investigation results showed, there did not have any great differences among different sizes of the original ensemble with respect to their predictive performance after pruning. The best performance was not always obtained with largest size ensemble. It can be concluded that the generalization accuracy could not make a significant improvement with the increase of ensemble size for a homogeneous ensemble created by using different parameters of SVR learning algorithm. On the other hand, the large ensemble size could be more robust than the small one, which could attribute to that different time window sizes could enhance the diversity of ensemble. This shows the advantage of the smart time window size selection procedure on time series forecasting task.

For RMSE and MAPE performance measurements, the subensemble obtained by rank-based pruning methods using ComTSP and ConTSP as evaluation measures could not always outperform

<p>| Table 8 | The prediction error (MAPE) on the test set of GlaxoSmithKline plc (GSK) time series. It showed four kinds of ensemble size to compare its influence. The boldface indicates the algorithm which performed best on this time series. |</p>
<table>
<thead>
<tr>
<th>MAPE</th>
<th>Ensemble size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>BSM</td>
<td>0.0326</td>
</tr>
<tr>
<td>Ensemble</td>
<td>ALL</td>
</tr>
<tr>
<td>Pruning</td>
<td>RetSP-Value</td>
</tr>
<tr>
<td>Method</td>
<td>RetSP-Trend</td>
</tr>
<tr>
<td>ComTSP</td>
<td>0.0253</td>
</tr>
<tr>
<td>ContSP</td>
<td>0.0315</td>
</tr>
</tbody>
</table>

<p>| Table 9 | The prediction error (MAPE) on the test set of Hang Seng Index (^HSI) time series. It showed four kinds of ensemble size to compare its influence. The boldface indicates the algorithm which performed best on this time series. |</p>
<table>
<thead>
<tr>
<th>MAPE</th>
<th>Ensemble Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>BSM</td>
<td>0.0288</td>
</tr>
<tr>
<td>Ensemble</td>
<td>ALL</td>
</tr>
<tr>
<td>Pruning</td>
<td>RetSP-Value</td>
</tr>
<tr>
<td>Method</td>
<td>RetSP-Trend</td>
</tr>
<tr>
<td>ComTSP</td>
<td>0.0389</td>
</tr>
<tr>
<td>ContSP</td>
<td>0.0597</td>
</tr>
</tbody>
</table>

<p>| Table 10 | The prediction error (MAPE) on the test set of Johnson Outdoors Inc. time series (JOUT). It showed four kinds of ensemble size to compare its influence. The boldface indicates the algorithm which performed best on this time series. |</p>
<table>
<thead>
<tr>
<th>MAPE</th>
<th>Ensemble size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>BSM</td>
<td>0.1073</td>
</tr>
<tr>
<td>Ensemble</td>
<td>ALL</td>
</tr>
<tr>
<td>Pruning</td>
<td>RetSP-Value</td>
</tr>
<tr>
<td>Method</td>
<td>RetSP-Trend</td>
</tr>
<tr>
<td>ComTSP</td>
<td>0.0801</td>
</tr>
<tr>
<td>ContSP</td>
<td>0.0784</td>
</tr>
</tbody>
</table>

<p>| Table 11 | The prediction error (POCID) on the test set of Dow Jones Industrial Average (^DJI) time series. It showed four kinds of ensemble size to compare its influence. The boldface indicates the algorithm which performed best on this time series. |</p>
<table>
<thead>
<tr>
<th>POCID</th>
<th>Ensemble size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>BSM</td>
<td>64.3679</td>
</tr>
<tr>
<td>Ensemble</td>
<td>ALL</td>
</tr>
<tr>
<td>Pruning</td>
<td>RetSP-Value</td>
</tr>
<tr>
<td>Method</td>
<td>RetSP-Trend</td>
</tr>
<tr>
<td>ComTSP</td>
<td>61.8298</td>
</tr>
<tr>
<td>ContSP</td>
<td>61.1152</td>
</tr>
</tbody>
</table>

<p>| Table 12 | The prediction error (POCID) on the test set of GlaxoSmithKline plc (GSK) time series. It showed four kinds of ensemble size to compare its influence. The boldface indicates the algorithm which performed best on this time series. |</p>
<table>
<thead>
<tr>
<th>POCID</th>
<th>Ensemble size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>BSM</td>
<td>53.2065</td>
</tr>
<tr>
<td>Ensemble</td>
<td>ALL</td>
</tr>
<tr>
<td>Pruning</td>
<td>RetSP-Value</td>
</tr>
<tr>
<td>Method</td>
<td>RetSP-Trend</td>
</tr>
<tr>
<td>ComTSP</td>
<td>54.0394</td>
</tr>
<tr>
<td>ContSP</td>
<td>54.2108</td>
</tr>
</tbody>
</table>

<p>| Table 13 | The prediction error (POCID) on the test set of Hang Seng Index (^HSI) time series. It showed four kinds of ensemble size to compare its influence. The boldface indicates the algorithm which performed best on this time series. |</p>
<table>
<thead>
<tr>
<th>POCID</th>
<th>Ensemble size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>BSM</td>
<td>65.9985</td>
</tr>
<tr>
<td>Ensemble</td>
<td>ALL</td>
</tr>
<tr>
<td>Pruning</td>
<td>RetSP-Value</td>
</tr>
<tr>
<td>Method</td>
<td>RetSP-Trend</td>
</tr>
<tr>
<td>ComTSP</td>
<td>69.0997</td>
</tr>
</tbody>
</table>

<p>| Table 14 | The prediction error (POCID) on the test set of Johnson Outdoors Inc. time series (JOUT). It showed four kinds of ensemble size to compare its influence. The boldface indicates the algorithm which performed best on this time series. |</p>
<table>
<thead>
<tr>
<th>POCID</th>
<th>Ensemble size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200</td>
</tr>
<tr>
<td>BSM</td>
<td>59.0952</td>
</tr>
<tr>
<td>Ensemble</td>
<td>ALL</td>
</tr>
<tr>
<td>Pruning</td>
<td>RetSP-Value</td>
</tr>
<tr>
<td>Method</td>
<td>RetSP-Trend</td>
</tr>
<tr>
<td>ComTSP</td>
<td>58.5431</td>
</tr>
<tr>
<td>ContSP</td>
<td>58.1437</td>
</tr>
</tbody>
</table>

<p>| Table 15 | Diebold-Mariano test results between ConTSP and ComTSP evaluation measures on four financial time series. |</p>
<table>
<thead>
<tr>
<th>Diebold-Mariano value</th>
<th>Time series</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>^DJI</td>
</tr>
<tr>
<td>1-Fold cross validation</td>
<td>-344.3964</td>
</tr>
<tr>
<td>3-Fold cross validation</td>
<td>-45.9716</td>
</tr>
<tr>
<td>4-Fold cross validation</td>
<td>-1.9992</td>
</tr>
<tr>
<td>5-Fold cross validation</td>
<td>-49.4551</td>
</tr>
</tbody>
</table>
the best single model (BSM) in the original ensemble. This can attribute to that using these measures to enhance diversity of subensemble is not suitable. According to these experimental results, it can be concluded that the diversity-based measurement should also consider the direction of predictive error for time series forecasting task rather than only consider its magnitude.

For POCID performance measurement, Tables 11–14 show that none of the evaluation measures for rank-based pruning could always obtain the best results. However, the measures ComTSP and ConTSP could perform the best sometimes. This phenomenon appears which can attribute to that the good POCID results are obtained at the cost of accuracy. And when the ensemble size reaches adequately big, the measure ConTSP could perform very well on the four datasets with regard to POCID measurement. It is indicated that the rank-based pruning method using ConTSP as evaluation measure could give a relatively correct prediction on time series trend in the case of large ensemble size, whereas the measure ReTSP-Trend has its relative advantage in trend prediction of time series in the case of small ensemble size.

Then, the Diebold-Mariano test (Diebold & Mariano, 2002) is used to compare predictive accuracy of these ensemble pruning methods on the five test datasets obtained by 5-fold cross-validation approach which has been mentioned in Section 6.1 for each time series. As the improvements with the increase of ensemble size to the predictive accuracy of a homogeneous ensemble is weak, we set the ensembles size identically as 200, and then evaluate the differences among the evaluation measures of rank-based ensemble pruning methods to check whether the difference is significant. And, at last, the ComTSP vs. ReTSP-Trend which performs relatively better in the above two groups are chosen and grouped together to check the difference between them. The Diebold-Mariano test results are shown in Tables 15–17.

The numbers displayed in bold in above tables represent that they cannot reject the null hypothesis, i.e. the difference between the predictive accuracy of the two corresponding algorithms is not significant at 5% significance level in this time of Diebold-Mariano test. From the above contrastive analysis, most of the Diebold-Mariano test results can reject the null hypothesis, which means that the difference between the predictive accuracy of the two corresponding algorithms is significant at 5% significance level. Therefore, it can be concluded that the predictive accuracy differences are significant among these four evaluation measures of rank-based ensemble pruning methods. And as a whole, it can be seen that ReTSP-Trend performs the best and it defeats the other three measures.

Additionally, we also show the forecasting results by random walk model in Table 18.

It has been found from previous research works that it is difficult to beat the random walk model in real time series forecasting task. And we can see from Table 18 that the random walk model could make excellent prediction performance according to RMSE and MAPE performance measurement for all these time series. However, its performance measured by POCID is worse than all the ensemble pruning methods which we proposed in this work. The POCID measurement demonstrates the percentage of the number of correct decisions when predicting whether the value of the time series will increase or decrease in the next time interval. For the investors, the predicted information of increase or decrease of a stock is used to decide a stock operation (buy/sell). Thus, the random walk model is less believable when decide a stock operation in real economic activity.

What is more, it has been suggested in Kilian and Taylor (2003) that the difficulty of beating the random walk model in real time does not reflect a problem with the forecast model based on economic fundamentals, rather it is a natural consequence of the small time span of data available for empirical work. And, the authors also found that the random walk model is significantly less accurate at longer horizons. Therefore, we will make multiple-step-ahead prediction procedure on the financial time series prediction task to compare the performance of our proposed ensemble pruning methods with the random walk model in our future work.

Next, we use figures to visualize the predictive performance of the pruned ensemble resulted from these pruning evaluation measures. As has been mentioned in Section 6.1, 5-fold cross-validation approach has been adopted in our experiments. Therefore, the forecasting results based on the test dataset obtained by the fifth fold cross-validation procedure are reported in detail in the following Figs. 2–5.

It can be clearly observed from the above figures that the reason why ComTSP and ConTSP could defeat ReTSP-Trend over POCID performance measurement is that they sacrifice their predictive accuracy for the POCID result. However, the measure ReTSP-Trend shows its great advantage over other measures for rank-based ensemble pruning technique, which can be observed from the figures quite obviously.

8. Conclusions

Rank-based ensemble pruning methods are very popular and conceptually the simplest. In this work, we propose four evaluation measures for rank-based ensemble pruning methods, specifically for time series prediction task, i.e. Complementarity measure for time series prediction (ComTSP); Concurrency thinning for time series prediction (ConTSP); and ReTSP-Value and ReTSP-Trend for
RE pruning for time series prediction (ReTSP). However, ComTSP and ReTSP-Value have their inherent defect that they cannot guarantee the most complementary predictor will be selected. The reason of this defect is that the predictive error in time series prediction is directional. It is not very reasonable for these measures to take reducing error as the only objective yet neglect the error direction.

However, our finally proposed measure ReTSP-Trend overcomes this defect, taking into consideration the forecasting error direction. It could indeed guarantee that the remaining predictor which supplements the subensemble the most would be selected. Results of comparative experiments based on four benchmark financial time series datasets demonstrate that the measure ReTSP-Trend possesses prominent advantages over the other three measures, which markedly enhances the generalization capability of the pruned ensembles. Moreover, ReTSP-Trend also outperforms the BSM and ALL methods, according to our experimental results.

Therefore, it is our main contribution of this work that, our proposed evaluation measure of ReTSP-Trend considers the direction of predictive error of the models in the ensemble with respect to time series forecasting task for the first time. This is substantially different from conventional ensemble pruning methods, which only consider the magnitude of predictive error of models in the ensemble. The proposed evaluation measure of ReTSP-Trend for rank-based ensemble pruning methods possesses pretty good performances when applied on time series forecasting task. And, most importantly, the method can reduce ensemble storage spaces, make the forecasting procedure speedy and make the forecasting results credible.

Another contribution of this paper is that, we propose a smart time window size selection procedure based on ensemble learning paradigm. This strategy can enhance the diversity of the ensemble for time series forecasting task. And the ensemble pruning methods with the smart time window size selection procedure could make a great accuracy-diversity trade-off.

Improvements to the robustness and accuracy of ensemble system with ensemble pruning methods have direct impact on forecasting performance, and further, the decision making procedure that the forecast supports. The proposed ensemble pruning method in this work can make good forecasting results on stock prices forecasting. They have extensive financial applications, for example: forecasting bankruptcy and business failure, foreign exchange rate and others. What is more, they can have practical implications for multiple forecasting applications, such as: climate change forecasting, earthquake prediction, political forecasting, telecommunications forecasting, weather forecasting, tourism forecasting, and economic forecasting, etc.

As for future research directions of our work, one attractive future work direction concerns enhancing the prediction accuracy of change trend by designing some evaluation measures which are based on prediction measures considering direction of change, such as POCID, for rank-based ensemble pruning methods. Another appealing direction concerns the development of some better integration methods of ensemble, which might improve the performance of rank-based ensemble pruning methods from another important aspect. Finally, considering the high volatility and noisy environment in practical time series forecasting applications, static approaches that are forced to select a fixed subensemble prior to seeing an unknown instance may have theoretical disadvantage compared to dynamic ones. Therefore, another promising future research direction concerns developing dynamic ensemble pruning methods instead of the static ones.

Acknowledgements

This work is supported by the National Science Foundation of China under the Grant Nos. 61100108 and 61375021, and is supported by the Natural Science Foundation of Jiangsu Province of China under the Grant No. SBK201322136, and is supported by “the Fundamental Research Funds for the Central Universities”, No. NZZ2013306, and Qing Lan Project. And we would like to express our appreciation for the valuable comments from reviewers and editors.

References