Ensemble Regression Trees for Time Series Predicitions

Article · January 2007

2 authors, including:

Huanhuan Chen
University of Science and Technology of China

Ensemble Regression Trees for Time Series Predicitions

Huanhuan Chen *Student Member, IEEE* and Xin Yao, *Fellow, IEEE*

***Abstract*— This paper propose to combine Bootstrap sampling and random subspace method for time series forecasting problems. The algorithm and methodology are described for the NN3 forecasting competition. A simple model selection approach based on minimizing in-sample Symmetric Mean Absolute Percent Error (SMAPE) is employed.**

I. INTRODUCTION

Over the last decades, there has been a lot of research directed at understanding and predicting the further. This research gives us many forecasting methods, most of them are replying on statistical models. Although these traditional statistical time series methods perform well on some forecasting problems, they have some inherent limitations due to the assumption of the statistical models. Some researchers tried to employ machine learning techniques to forecast the time series.

Ensemble of multiple learning machines, i.e. a group of learners that work together as a committee, has received a lot of research interests in the machine learning community because it is thought as a good approach to improve the generalization ability [1]. The term "ensemble" can be used to describe the paradigm that brings together a number of learning machines to provide a single output. This technique originates from Hansen and Salamon's work [1], which shows that the generalization ability of a neural network can be significantly improved through ensembling a number of neural networks. Because of the simple and effective properties, neural network ensemble has become a hot topic in machine learning communities and has already been successfully applied to many areas, for example face recognition [2], character recognition [3], image analysis [4], etc.

Bagging [5] and Random feature subspace [6], the well-known ensemble algorithms which have attracted an extensive research interest, employ bootstrap sampling [7] of training data and random selection of feature subsets to promote the diversity and thus improve the performance of ensemble.

This paper will briefly introduce the algorithm that combines Bootstrap Sampling of data and Random Subspace method (BSRS) for time series prediction. The algorithm BSRS is introduced in Section II and the methodology for NN3 competition is presented in Section III. Finally, Section IV concludes the paper.

Huanhuan Chen is with CERCIA, School of Computer Science, University of Birmingham, Birmingham B15 2TT, United Kingdom (phone: 44-121-4143736; email: H.Chen@cs.bham.ac.uk).

Xin Yao is with CERCIA, School of Computer Science, University of Birmingham, Birmingham B15 2TT, United Kingdom (phone: 44-121-4143747; email: X.Yao@cs.bham.ac.uk).

II. BOOTSTRAP SAMPLING OF DATA AND RANDOM SUBSPACE METHOD (BSRS)

Ensembles [1] is a learning paradigm where a collection of learning machines are trained for the same task. There have been many ensemble methods studied in the literatures, such as Bagging [5], Boosting [8], ensemble of features and so on.

Bagging is proposed in [5] based on Bootstrap sampling [7]. In a Bagging ensemble, each base learner is trained on a set of n training samples, drawn uniformly at random with replacement from the original training set of size n. Predictions on new samples are made by simple averaging. For unstable learners such as neural networks or regression trees Bagging works very well. Generally speaking, Bagging can reduce the error due to deduction of variance of the base learner [9].

Apart from randomly sampling the training set, random feature subspace [6] method generates ensemble by adopting different feature subsets for different ensemble members to promote the diversity [6], [10] and most of the existing ensemble feature method claims better results than traditional methods [11], [6].

BSRS is an algorithm that combines Bootstrap sampling and random subspace method to ensemble regression trees. It is consisted of a number of regression trees which grow with the examples, bootstrapped sampled from the training set, with randomly selected feature subset.

III. METHODOLOGY

A. Model Specification

In BSRS algorithm, classification and regression tree (CART) is employed as based classifiers. CART is selected since it performs well and does not require additional parameters to optimize. In this competition, 500 regression trees are used to constitute the ensemble; Each tree grows with the training subset, bootstrapped sampled from the training set, with randomly selected feature subset.

Bagging is based on bootstrap sampling and each base CART is trained on $1 - 1/e \approx 0.632$ training points [7], i.e. randomly sample 63.2% of data to train each component tree. In this algorithm, 70% of feature is randomly selected in training. In total, to grow each CART in BSRS, 63.2% of data is randomly selected; Within the selected data, 70% of feature is randomly selected.

B. Training Set Generation

In the machine learning algorithms, a training set $\{X_i, Y_i\}$ is required to train the corresponding learning machines. For

Algorithm 1 Ensemble of Regression trees, BSRS, For Time Series Prediction

1: **Input:** 111 Time series $TS_i(length_i)$ where $i = 1...111$ is the number of time series involved in the experiments, $length_i$ is the length of the time series i; Tmax is the maximal value of $lag = 12$.
2: **Output:** Forecasted Time series with FTS_i, where $i = 1...111$.
3: **for** $i = 1$ to 111 **do**
4:     **for** $lag = 1$ to $Tmax$ **do**
5:         $[trainX, trainY] = GenerateDataset(TS_i, lag);$
6:         $[Predicted\_Series, Fitted\_Series] = BSRS(TS_i, trainX, trainY);$
7:         $Temp\_Series\{lag\} = [TS_i(1 : lag); Fitted\_Series; Predicted\_Series];$
8:         $Train\_SMAPE(i, lag) = SMAPE(TS_i(lag + 1 : length_i), Fitted\_Series);$
9:     **end for**
10:     $[temp, idx] = min(Train\_SMAPE(i, :));$
11:     $TFS_i = Temp\_Series\{idx\};$
12: **end for**

Fig. 1. The Matlab Pseudocode of our Algorithm for Time Series Prediction

time series prediction problems, since there is only a series available, we need to generate the training set from the series. The parameter needed in the generation is the lag number, which indicates how many previous steps will influence the next step. Firstly, the generation process takes the first lag numbers in the series, $t_1 \cdots t_{lag}$ as input X_1 and t_{lag+1} as Y_1, and then move the window, whose size is lag, to t_2, meaning that $X_2 = [t_2 \cdots t_{lag+1}]$ and $Y_2 = [t_{lag+2}]$, until the Y_i reaches the end of the time series.

C. Model Selection

As we know, different time series forecasting problems have different window size, i.e. lag in the generation of training set. In order to select the parameter, a simple model selection, which selects the lag value that minimizes the Symmetric Mean Absolute Percent Error (SMAPE) across the training time series, is employed in the competition.

$$SMAPE = \frac{1}{n} \sum_{t=lag+1}^{n} \frac{|X_t - F_t|}{(X_t + F_t)/2} \times 100 \qquad (1)$$

where X is the actual time series, F is the forecasted series, n is the number of observations. In the subscript of Equation 1, $lag + 1$ is used instead of 1 since the first lag numbers have been used for generating training set and our algorithm can not forecast these values.

D. Matlab Pseudocode and Explanation

In the following, the pseudocode written in matlab and the detailed explanation is illustrated as follows:

Algorithm: Ensemble of Regression Tree BSRS to Forecast Time Series

For each time series TS_i, where $i = 1 \cdots 111$.

1) For lag from 1 to Tmax, generate training set: trainX and trainY, by TS_i and the lag, i.e. the data from $t - lag$ to t is used to predict the value $t + 1$ (Lines 4-5).
2) Apply BSRS algorithm to compute the forecasted training series $Fitted\_Series$ and the predicted series $Predicted\_Series$, whose length is 18 according to NN3 competition (Line 6).
3) Save the temporary full series $Temp\_Series$ including the forecasted training series and the predicted series (Line 7).
4) Compute the SMAPE of the fitted training series and the training series TS_i, excluded the first lag numbers since they are employed for training (Line 8).
5) Select the model that minimize the SMAPE $Train\_SMAPE(i, :)$ and record the corresponding full forecasted series (Lines 10-11).

IV. Conclusion

In the competition, an ensemble algorithm that combines Bootstrap sampling and random subspace method is proposed for time series forecasting problems. In order to select the lag parameter that used to generate the data set, a simple model selection based on minimizing in sample SMAPE is employed. The algorithm is efficient and simple to implement.

References

[1] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[2] F. J. Huang, T. Chen, Z. Zhou, and H. Zhang. Pose invariant face recognition. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, pages 245–250, Washington, DC, USA, 2000.

[3] L. K. Hansen, L. Liisberg, and P. Salamon. Ensemble methods for handwritten digit recognition. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 333–342, Helsingoer, Denmark, 1992.

[4] K. J. Cherkauer. Human expert level performance on a scientifi c image analysis task by a system using combined artifi cial neural networks. In *Proceedings of AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, pages 15–21, Menlo Park, CA, USA, 1996.

[5] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
[6] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
[7] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, London, U.K., 1993.
[8] R. E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406, 1999.
[9] J. H. Friedman. Bias, variance, 0-1 loss and the curse of dimensionality. Technical report, Stanford University, 1996.
[10] D. W. Optiz. Feature selection for ensembles. In *Proceedings of the 16th International Conference on Artificial Intelligence*, pages 379–384, 1999.
[11] J. O'Sullivan, J. Langford, R. Caruana, and A. Blum. Featureboost: A meta-learning algorithm that improves model robustness. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 703–710, San Francisco, CA, USA, 2000.