On the benefit of using time series features for choosing a forecasting method

Christiane Lemke and Bogdan Gabrys

Bournemouth University - School of Design, Engineering and Computing
Poole House, Talbot Campus, Poole, BH12 5BB - United Kingdom

Abstract. In research of time series forecasting, a lot of uncertainty is still related to the question of which forecasting method to use in which situation. One thing is obvious: There is no single method that performs best on all time series. This work examines whether features extracted from time series can be exploited for a better understanding of different behaviour of forecasting algorithms. An extensive pool of automatically computable features is identified, which is submitted to feature selection algorithms. Finally, a possible relationship between these features and the performance of forecasting and forecast combination methods for the particular series is investigated.

1 Introduction

Extensive empirical studies of the performance of forecasting and forecast combination algorithms, for example conducted by Makridakis and Hibon [1] and Stock and Watson [2], revealed that there is no clear cut winner among the pool of methods investigated which works well for all time series. In a response to the results of the M3 competition [1], Robert J. Hyndman [3] put the future challenges for time series forecasting research into the following words: *"Now it is time to identify why some methods work well and others do not"*.

It is generally acknowledged that different types of time series require different treatment. This brings up the question if characteristics of time series can be used to draw conclusions about which method will work best for forecasting their future values. This work investigates an automatic approach to this problem, since the thorough analysis by experts is often not feasible in practical applications that process a large number of time series in very limited time.

A classic and straightforward classification for time series has been given by Pegels [4]. Time series can thus have patterns that show different seasonal effects and trends, both of which can be additive, multiplicative or non-existent. Gardner [5] extended this classification by including damped trends. Time series do however have many more features that can be taken into account for a potential selection of a method that works best.

Time series analysis in order to find an appropriate ARIMA model has been discussed since the seminal paper of Box and Jenkins [6]. Guidelines are summarised in [7] and rely heavily on examining autocorrelation and partial autocorrelation values of a series. Some publications focus on automatically detecting time series characteristics for model selection: Adya et al. [8] identify 28

possible features of time series that are used for a rule-based forecasting system presented in [9]. This system weights and selects between the forecasting techniques random walk, linear regression, Holt's exponential smoothing and Brown's exponential smoothing. Parameters of the smoothing methods are also determined via rules. This method was submitted to the M3 competition ([1]) but did not provide convincing results.

Vokurka et al. [10] present another rule-based expert forecasting system, which performs automatic preprocessing of the series and automatically determines features of the time series to choose between a simple exponential smoothing, a dampened trend exponential smoothing and a decomposition approach as well as a simple-average combination of these three. This approach was able to improve upon a random walk model and the simple average combination.

The work presented here significantly extends the feature pool that was used in the publications introduced in the previous paragraph. Another focus lies on the functional diversity of the pool of forecasting and combination algorithms. The paper is organised as follows: Section two introduces the methodology of the underlying empirical experiments and justifies the choice of the forecasting and forecast combination algorithms. Section three describes the feature pool and feature selection processes. A relationship between the features and the performance of forecasting approaches is sought in section four. Section five concludes.

2 Underlying empirical experiments

A data set consisting of 111 monthly empirical business time series with 52 to 126 observations has been obtained from a Forecasting Competition conducted in 2006/2007 [11]. The task was to predict 18 future values. In previous work published in [12], experiments on this data set are summarised, using the last 18 observations of the provided time series for an out-of-sample error estimation. The forecast pool consisted of eight forecasting and seven forecast combination algorithms for single-step-ahead prediction as well as twelve forecasting and seven forecast combination algorithms for multi-step-ahead prediction. Where applicable, two approaches for parameter estimation have been considered, namely grid searching for a value that performs best in-sample (*tuned methods*) and setting the parameter to the middle of the parameter range (*untuned methods*).

As a number of the implemented forecasting and forecast combination methods shared the same functional approach, it was considered beneficial to choose just one from every group to reduce the number of class labels and gain clearer insights into which method works best for which time series. In an attempt to obtain a functionally diverse and well-performing method pool, the following methods have been selected:

One-step-ahead forecasting *Taylor's exponential smoothing (Taylor):* A modified dampened trend exponential smoothing was introduced in [13]. A growth rate and the level of the time series are estimated by exponential smoo-

thing and then combined with a multiplicative approach. All parameters are determined by a grid search or set to 0.5.
ARIMA: Autoregressive integrated moving average models (ARIMA) according to Box and Jenkins [6] are models with an autoregressive and a moving average part, fitted to differenced data. The original series as well as its first and second order differences are submitted to the automatic ARMA selection process of a MATLAB toolbox [14], choosing the prediction with the lowest in-sample error. The same process is implemented with undifferenced series only.
Neural network (NN): A feedforward neural network with one hidden layer containing 12 neurons, trained by a backpropagation algorithm with momentum has been implemented. Input variables are 12 lagged values of the time series. These characteristics have been selected based on findings of an extensive review of work using artificial neural networks for forecasting purposes by Zhang et al. [15]. Ten neural networks have been trained and their predictions averaged.
Variance-based combination model (VBW): Weights for a linear combination of forecasts are determined using past forecasting performance ([16]).
Variance-based pooling, three clusters (VBP): Past performance is used to group forecasts into two or three clusters by a k-means algorithm as suggested by Aiolfi and Timmermann [17]. Forecasts of the historically better performing cluster are then averaged to obtain a final forecast.
Regression combination (Regr): In regressing realisations of the target variable on forecasts over past periods, combination weights are estimated by a least squares approach with weights being restricted to be non-negative.

Multi-step-ahead forecasting *Taylor's exponential smoothing (Taylor):* This method is implemented as described for the one-step-ahead problem, but following a direct approach for the multi-step prediction, where n different models are trained directly on the multi-step problem.
ARIMA: An ARIMA model can natively provide multi-step-ahead forecasts, so the single-step method remains unchanged.
Neural network (NN): This was also implemented as described above, obtaining multi-step-ahead predictions by feeding the last forecast back to the model.
Simple average with trimming (SAT): This algorithm averages individual forecasts, only taking the best performing 80% of the models into account.
Variance-based pooling, two clusters (VBP): This is implemented as in the multi-step problem, only using two clusters instead of three.

3 Time series features and their selection

Based on the previous section, a classification task can be formulated as follows: Given a set of time series features, can we predict a) the best performing forecasting method, b) the best performing forecast combination method or c) whether or not combinations work better than individual methods? Each of the three problems can be investigated for single- and multi-step-ahead forecasting.

Table 1 summarises the resulting six problems, for each of which tuned and untuned individual methods as explained in section 2 can be used.

One-step-ahead	
best forecasting method:	3 classes: Taylor, ARIMA, NN
best combination method:	3 classes: VBW, VBP
best general approach:	2 classes: individual method or combination
Multi-step-ahead	
best forecasting method:	3 classes: Taylor, ARIMA, NN
best combination method:	2 classes: SAT, VBP
best general approach:	2 classes: individual method or combination

Table 1: Classification tasks, abbreviations referring to methods introduced in section 2.

Based on the publications cited above and a book by Makridakis et al. [7], a number of features listed in table 2 have been identified.

descriptive statistics	
abbreviation	description
slope	trend (absolute value of the slope of linear regression line)
std	standard deviation of de-trended series
stdrate	ratio between the standard deviation of the first and second half of the de-trended series
skew	skewness of series
kurt	kurtosis of series
sign	sign change measure (counting sign changes of de-trended series divided by its length)
length	length of series
pred	predictability measure according to [18]
nonlin	nonlinearity measure also according to [18]
frequency domain	
abbreviation	description
ff[1-3]	frequencies at which the three maximal values of the power spectrum occur
ff[4]	maximum value of the power spectrum of the fourier transform of the series
ff[5]	number of peaks not lower than 60% of the maximum peak
autocorrelations	
abbreviation	description
acf[1-12]	autocorrelations at lags 1-12
pacf[1-12]	partial autocorrelations at lags 1-12

Table 2: Feature pool

Including irrelevant features in a machine learning algorithm can cause degrading performance of the resulting model [19]. The use of redundant attributes may have the same effect. This is why one automatic and one judgemental feature selection algorithm have been used on the complete feature pool in order to generate a suitable subset of features. Judgementally, the following six features have been selected:

• The intuitive sign change measure, to capture volatility.

• The length of the series, as the number of observations available for training might influence the performance of methods.

• The nonlinearity measure, to quantify predictability of a series.

• The maximum value of the power spectrum of the fourier transform of the series, to identify a strong higher- or lower frequent component

• Partial autocorrelations at lag one and twelve, to capture nonstationarity and yearly seasonality if present.

The automatic method called "Subset Selection" was proposed in [20] and is implemented in the Weka collection of machine learning algorithms [19]. It belongs to the so-called filter methods, which are known for fast and efficient selection of features in a preprocessing step, independent of a learning algorithm. The quality of a feature subset is measured by two components: the individual predictive power given by correlation values and the level of intercorrelation among them. Searching the feature space is done using a Best First algorithm with an empty feature set as a starting point. All possible expansions are then evaluated and the best one is picked to be expanded again.

Using a ten-fold cross validation and selecting features that have been chosen in at least five of the ten calculations, tables 3 and 4 list the features selected for each of them.

Tuned methods	
class label	selected features
best individual forecast	slope, skew, nonlin, acf[1-11], pacf[1,3-4, 6-8, 10-11]
best combination	slope, std, acf[1,9-11], pacf[1,3,8,10-11]
individual vs combination	acf[11], pacf[6,10-11]
Untuned methods	
class label	selected features
best individual forecast	acf[4], pacf[3,8]
best combination	pacf[5,11]
individual vs combination	pacf[2-3,10]

Table 3: Features automatically selected for one-step-ahead forecasting

The tables show that the automatic approach generally chooses completely different features for each of the twelve sub-problems identified. Consequently it can be concluded, that there is no obvious feature that helps to decide for a suitable algorithm in every case. Appearing in seven cases, partial autocorrela-

Tuned methods	
class label	selected features
best individual forecast	skew, acf[7], pacf[5-6]
best combination	std, pacf[2,4,8]
individual vs combination	skew, acf[6], pacf[6]
Untuned methods	
class label	selected features
best individual forecast	ff[1-2,4], pacf[4-7,10]
best combination	std, pacf[2,5-9]
individual vs combination	acf[6], pacf[6]

Table 4: Features automatically selected for multi-step-ahead forecasting

tion at lag six is the feature that gets selected most, indicating that seasonality might be a factor that is important to many of the decisions.

4 Results

Decision trees have been selected as a simple machine learning method giving easily interpretable results. They are built in Matlab, choosing the minimum-cost-tree after a ten-fold crossvalidation. In the figures, the leaf to the left of a node represents the data that fulfils its condition, the leaf to the right hand side represents data that does not. The numbers following the methods in the leafs denote the number of times this particular method performed best on the data subset.

4.1 One-step-ahead

For one-step-ahead tuned forecasting methods, the trees in Figure 1 are created, both having a misclassification cost of 48.6%. Both essentially say the same thing: neural networks work better with yearly seasonality. This is not too surprising since yearly differences have not been taken for the ARIMA model, which works best with purely stationary data. No tree was built for combination methods, both feature sets suggest the regression method with a misclassification cost of 54.9%. The same occurs for the question of whether to use individual methods or combinations, combinations are suggested at a cost of 35.1%.

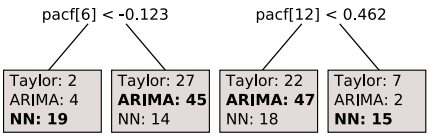

Fig. 1: Tree for tuned forecasting methods, left: automatically selected features, right: judgemental feature selection

Figure 2 shows minimum cost trees for untuned individual methods. The subset selection feature set suggests a neural network if the autocorrelation at lag four is below a certain number and an ARIMA model if it is above (cost 38.7%). Like for the tuned individual methods, the judgementally selected feature set suggests a neural network for series with stronger seasonality and an ARIMA model otherwise (cost 37.8%).

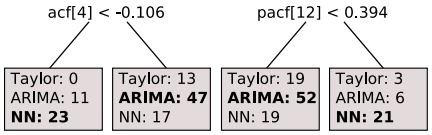

Fig. 2: Tree for untuned forecasting methods, left: automatically selected features, right: judgemental feature selection

For combinations, the subset method suggests the regression method (cost 52.2%), while the judgemental method produces a tree with two nodes (cost: 33.3%) shown in figure 3, which can be read as follows: For longer series, a regression approach seems to work best, while variance-based pooling works better for shorter series with a stronger negative partial autocorrelation at lag one. Variance-based weights are the best option for short series with a positive or small negative partial autocorrelation at lag one. It can be suspected that the regression approach that takes all individual methods into account might need more stable individual forecasts than the others, which cannot be provided by series with a smaller training set. The strong dynamic trimming carried out by variance-based pooling works best for more stationary series, while non-stationarity might be handled better with weights calculated based on past variance.

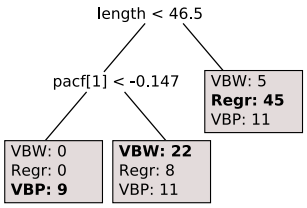

Fig. 3: Tree for untuned combination methods, judgemental feature selection

Comparing individual (fc) and forecast combination (fcc) methods, two trees are shown in figure 4, producing costs of 36.9% and 38.7%, respectively. The tree generated with features based on subset selection is not intuitively readable, having seemingly random partial autocorrelation values as conditions in the nodes. The other tree suggests individual forecasting methods for series with a stronger negative autocorrelation at lag one and combinations otherwise.

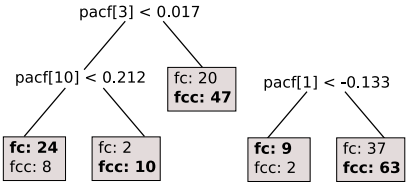

Fig. 4: Tree comparing untuned methods, left: automatically selected features, right: judgemental feature selection

4.2 Multi-step-ahead

Trees for tuned multi-step-combination models with quite high misclassification costs (59.0% and 57.4%) are shown in figure 5. The tree generated by subset selected features suggests methods depending on the partial autocorrelation at lag 4. Judgementally selected features produce a tree that suggests an ARIMA method for low-frequent zigzag and a neural network for a higher-frequent one.

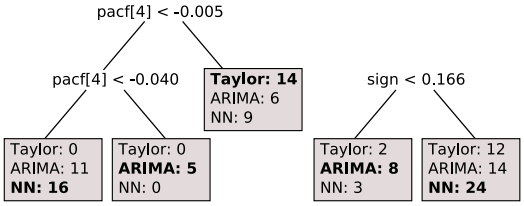

Fig. 5: Tree for tuned forecasting methods, left: automatically selected features, right: judgemental feature selection

For combination of tuned methods, the subset feature selection proposes simple average with trimming (cost 45.9%). Judgemental selection produces the tree shown in figure 6 with a cost of 40.9%, suggesting simple average with trimming for series with weaker seasonality and variance-based pooling otherwise. This might be explained by some methods not being capable of handling seasonality, which are hopefully dynamically removed from the combination in the variance-based pooling approach. Comparing individual to combination methods, both feature selection algorithms suggest individual methods (cost 29.5%).

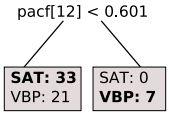

Fig. 6: Tree for tuned combination methods, judgemental feature selection

For untuned multi-step-ahead methods, a one-leaf tree is generated for most cases, suggesting a neural network as an individual method (cost: 55.7%), variance-based pooling as a combination (cost: 44.2%) and individual forecasts over combinations (cost 29.5%). Only judgemental feature selection for individual methods produces an actual tree (cost 47.5%) which is shown in figure 7. It suggests using neural networks for series with lesser nonstationarity indicated by the autocorrelation at lag one. On the other side of the tree, a neural network is again suggested for seasonal series, while series with lower seasonality and a higher nonlinearity measure are better predicted with Taylor's or the ARIMA method.

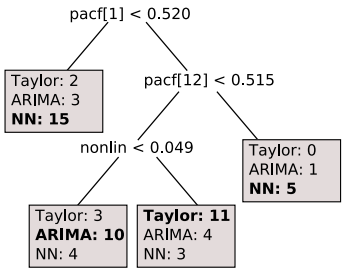

Fig. 7: Tree for untuned forecasting methods, judgemental feature selection

5 Conclusions

This paper investigates an automatic approach to use time series features for choosing a method that will work well for their forecasting. It extends the feature pool of previous work as well as the diversity of methods used as class labels. Both a judgemental and an automatic approach to feature selection have been employed. As a first interesting result, the automatic feature selection approach selected different features for every sub-problem, indicating that there is no obvious feature that always affects the performance of forecasting methods.

Summarising the results presented in section four, it can be seen that characteristics of time series can in some cases give an indication about which method might work best for forecasting its future values. Looking at features in the nodes of the trees, the partial autocorrelation at the lags one and twelve are often present, indicating that nonstationarity and seasonality of a series are important factors for choosing a prediction method. However, the seasonality issue also shows the importance of data preprocessing, because some of the differences in performances of the methods might not occur if quarterly, yearly or any other seasonality had been removed from the series in a preprocessing step.

However, not every sub-problem produced a decision tree that could easily be interpreted. This suggests that it could be beneficial to further extend the feature pool and selection of methods in future work, or that there must be

other mechanisms than just the characteristics of the time series that decide about success or failure of a forecasting method.

References

[1] S. Makridakis and M. Hibon. The M3-Competition: Results, Conclusions and Implications. *International Journal of Forecasting*, 16(4):451–476, 2000.

[2] J.H. Stock and M.W. Watson. A Comparison of Linear and Nonlinear Univariate models for Forecasting Macroeconomic Time Series. In R.F. Engle and H. White, editors, *Cointegration, causality and forecasting. A festschrift in honour of Clive W.J. Granger*, pages 1–44. Oxford University Press, 1999.

[3] Commentaries on the M3-Competition, October-December 2001.

[4] C.C. Pegels. Exponential Forecasting: Some New Variations. *Management Science*, 15(5):311–315, 1969.

[5] E. S. Gardner. Exponential Smoothing: The State of the Art. *Journal of Forecasting*, 4(1):1–28, January-March 1985.

[6] G.E.P. Box and G.M. Jenkins. *Time Series Analysis*. Holden-Day San Francisco, 1970.

[7] S.G. Makridakis, S.C. Wheelwright, and R.J. Hyndman. *Forecasting: Methods and Applications*. John Wiley, New York, 3rd edition, 1998.

[8] M. Adya, F. Collopy, J.S. Armstrong, and M. Kennedy. Automatic identification of time series features for rule-based forecasting. *International Journal of Forecasting*, 17(2):143–157, 2001.

[9] M. Adya, J.S. Armstrong, F. Collopy, and M. Kennedy. An application of rule-based forecasting to a situation lacking domain knowledge. *International Journal of Forecasting*, 16:477–484, 2000.

[10] R.J. Vokurka, B.E. Flores, and S.L. Pearce. Automatic feature identification and graphical support in rule-based forecasting: a comparison. *International Journal of Forecasting*, 12(4):495–512, 1996.

[11] NN3 Forecasting Competition [Online], 2006/2007. Available online: http://www.neural-forecasting-competition.com/ [13/06/2007].

[12] C. Lemke and B. Gabrys. Do we need experts for time series forecasting? In *Proceedings of the 16th European Symposium on Artificial Neural Networks, Bruges*, pages 253–258, 2008.

[13] J. W. Taylor. Exponential Smoothing with a Damped Multiplicative Trend. *International Journal of Forecasting*, 19(4):715–725, October-December 2003.

[14] Delft Center for Systems and Control - Software [Online], 2007. Available online: http://www.dcsc.tudelft.nl/Research/Software [13/06/2007].

[15] G. Zhang, B.E. Patuwo, and M.Y. Hu. Forecasting with Artificial Neural Networks: The State of the Art. *International Journal of Forecasting*, 14:35–62, 1998.

[16] P. Newbold and C.W.J. Granger. Experience with Forecasting Univariate Time Series and the Combination of Forecasts. *Journal of the Royal Statistical Society. Series A (General)*, 137(2):131–165, 1974.

[17] M. Aiolfi and A. Timmermann. Persistence in Forecasting Performance and Conditional Combination Strategies. *Journal of Econometrics*, 127(1-2):31–53, 2006.

[18] T. Gautama, D.P. Mandic, and M.M. Van Hulle. A novel method for determining the nature of time series. *IEEE Transactions on Biomedical Engineering*, 51, 2004.

[19] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.

[20] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.