Selecting Neural Network Forecasting Models Using the Zoomed-Ranking Approach

Patrícia M. Santos, Teresa B. Ludermir, Ricardo B. C. Prudêncio
Center of Informatics, Federal University of Pernambuco
Pobox 7851 - CEP 50732-970 - Recife (PE) - Brazil
{pms,tbl,rbcp}@cin.ufpe.br

Abstract

In this work, we propose to use the Zoomed-Ranking approach to ranking and selecting Artificial Neural Network (ANN) models for time series forecasting. Given a time series to forecast, the Zoomed-Ranking provides a ranking of the candidate models, by aggregating accuracy and execution time obtained by the models in similar series. The best ranked model is then returned as the selected one. In order to evaluate this proposal, we implemented a prototype to rank three ANN models for forecasting time series from different domains. In the experiments, the rankings of models recommended by Zoomed-Ranking were significantly correlated to the ideal rankings.

1 Introduction

The selection of an adequate model for forecasting a time series is an important aspect to the success of the forecasting process. The traditional approaches to model selection involve in general costly empirical evaluations [1] or require expert knowledge [2], not always easy to acquire. A promising approach to model selection is the use of Machine Learning [3, 4]. In this approach, a learning algorithm predicts the best candidate model, based on descriptive features (e.g. length, basic trend,...) of the time series to forecast.

Considering the above context, in [5], the authors applied the Zoomed-Ranking (ZR) approach [6] not only to select the best model, but also to rank the candidate models. The ZR was originally proposed in the context of the Meta-Learning field [7, 8] to rank and select machine learning algorithms based on features of the training sets. In the model selection problem, each training example for the ZR is related to a time series and stores: (1) the descriptive features of the series; and (2) the forecasting error and the execution time obtained by each candidate model, when used to forecast the series. Given a new time series, the most similar examples are retrieved from the training set. Following, a ranking of models is suggested for the new series by aggregating the forecasting error and execution time obtained by the models in the similar series.

In the current work, we extend the previous research by investigating the use of ZR to select Artificial Neural Network (ANN) forecasting models. Our motivation here is that, although the ANN models have been successfully used to forecast different time series, there is few knowledge to guide non-expert users in applying the ANNs [9].

In order to evaluate the viability of our proposal, we implemented a prototype to rank the following models: (1) the Time Delay Neural Network (TDNN) [10]; (2) the Time-Lagged RBF network [11]; and (3) the SOM with Temporal Activity Diffusion (SOMTAD) network [12]. Experiments were performed on a set of time series collected from a benchmark repository. The obtained results revealed that the rankings of models suggested by ZR were strongly correlated to the ideal rankings observed for the series in a test set.

Section 2 presents some approaches to model selection. Section 3 presents a brief introduction to Meta-Learning and its relationship to the problem of model selection. Section 4 describes the implemented prototype that uses Zoomed-Ranking to rank ANN models. Section 5 shows the experiments and obtained results, and finally, section 6 concludes the paper and presents the future work.

2 Time Series Model Selection

Different time series models have been developed in the literature. However, there is no model considered the most adequate, independently on the time series and the forecasting objectives [2]. A straightforward solution to model selection is to perform an empirical evaluation (e.g. hold-out, cross-validation,...) using the available time series data, and compare the estimated performance obtained by the candidate models [1]. Despite its simplicity, this solution is costly for a large amount of series to forecast or several candidate models to evaluate [9].

A more efficient approach to selecting models is based on the development of expert systems [2], in which rules are designed to relate time series features (e.g. length, basic trend, autocorrelations...) and the candidate models performance. A landmarking work in this approach is the Rule-Based Forecasting system [2]. In this work, an expert system with 99 rules is used to weight and select four forecasting methods. The authors used time series features in the rules to modify the weight associated to each model (see figure 1 as an example). In the experiments performed using the expert system, the improvement in accuracy has shown to be significant. The main limitation of the expert system approach, however, is the difficulty in acquiring knowledge, since that good experts in time series forecasting are expensive and not always available [3]. This limitation may be even more drastic in the case of more complex models.

Rule:	Insignificant Basic Trend - IF NOT a significant basic trend, THEN add 0.05 to the weight on random walk model and subtract it from that on the linear regression model.

Figure 1. Example of rule implemented in the Rule Based Forecasting system.

In order to minimize the above difficulty, a number of authors have used machine learning algorithms to automatically acquire knowledge for model selection [3, 4]. In this solution, each training example is generated from a time series and stores: (1) the features describing the series; and (2) the class attribute, indicating the best candidate model for that series (in general, the best model is defined as the model which obtained the lowest forecasting error in an empirical evaluation). A supervised learning algorithm is then used to associate the time series features to the best models.

The use of learning algorithms as described above reduces the need for experts and facilitates the inclusion of new models in the selection process. The drawback of this solution, however, is that it only considers the forecasting accuracy (i.e. lowest forecasting error) as performance criteria to model selection. As it will be seen, the use of Meta-Learning techniques can overcome this limitation.

3 Meta-Learning

Meta-Learning is a framework developed in the field of supervised machine learning with the aim of selecting the best learning algorithm for a given task (usually classification tasks)[7, 8]. Meta-Learning uses empirical examples to produce a machine learning model (meta-learner) responsible for associating the candidate algorithms (base-learners) with the characteristics of the learning tasks.

In a strict formulation of Meta-Learning [13], each meta-example is related to a single learning task and stores: (1) descriptive features of the learning task, i.e. its *meta-features*; and (2) a class attribute associated to the base-learner which obtained the highest accuracy in the learning task. In this formulation, the meta-learner just becomes a classifier in which the meta-features correspond to predictor attributes for the class associated to the best base-learners. Meta-features are in most cases statistical and information theoretic features of the available datasets (e.g., number of examples, number of attributes, correlations among attributes, class entropy,...). The best base-learner (i.e. the class attribute) in turn is usually defined via a cross-validation experiment.

Different approaches have been proposed in order to add new functionalities in the Meta-Learning process. In [14, 15], for instance, a set of different meta-learners is used not only to predict the best base-learner (as described above) but also to recommend a ranking of base-learners. In this approach, a strict meta-learner is built for each different pair (X, Y) of base-learners. Given a new learning task, the outputs of the meta-learners are collected and then, points are credited to the base-learners according to the outputs. For instance, if ’X’ is the output of meta-learner (X, Y) then the base-learner X is credited with one point. The ranking of base-learners is recommended for the new task directly from the number of points assigned to the base-learners.

The Meta-Regression approach [16] tries to directly predict the accuracy (or alternatively the error) of each base-learner instead of simply predicting the class that corresponds to the best algorithm. The meta-learner in this case may be used either to select the base-learner with the highest predicted accuracy or to provide a ranking of algorithms based on the order of predicted accuracies. In [16], the authors obtained good results when a linear regression model was used to predict the accuracy of 8 different classification algorithms.

In the Zoomed-Ranking approach [6], the authors originally proposed the use of instance-based learning in order to produce rankings of base-learners taking into account accuracy and execution time. In this approach, each meta-example stores the meta-features describing a learning task, as well as the accuracy and execution time obtained by each base-learner in the task. Given a new learning task, the Zoomed-Ranking retrieves the most similar tasks based on the similarity of meta-features. The ranking of base-learners is then recommended for the new task by deploying a multi-criteria measure that combines the total accuracy and execution time obtained by the base-learners in the similar tasks. More recently, the authors provided a deeper investigation of these ideas [17].

As said, the concepts of Meta-Learning were originally evaluated to select algorithms for classification tasks. In recent years, Meta-Learning has been extrapolated to other domains of application [21, 18]. In [9], the authors originally proposed the use of Meta-Learning to the problem of time series model selection. In this context, the time series models correspond to the base-learners, and the model selection is seen as the task of algorithm selection. The meta-features are attributes used to describe the time series being forecasted. When applied to model selection, the Meta-Learning techniques are able not only to select one best model based on accuracy (as seen in section 2), but also to provide rankings of models and consider multiple performance criteria in the model selection process. The viability of this proposal was investigated in a different number of case studies [5, 9, 19, 20, 21].

In [5], the Zoomed-Ranking was applied to rank three linear time series models: the Random Walk, the Holt Exponential Smoothing model and the Linear Autoregressive model. The ranking of models for a given time series was generated taking into account forecasting error and execution time obtained by the models in similar series. In [5], experiments were performed on a set of 215 yearly time series. The obtained results were very promissing when the rankings suggested by Zoomed-Ranking were compared to the correct rankings of models.

4 Zoomed-Ranking for ANN Model Selection

In the current research, we investigated the use of Zoomed-Ranking (ZR) to rank Artificial Neural Networks (ANNs) forecasting models. The motivation here is that, although ANNs represent a powerful approach to forecasting, there is not much knowledge to guide its usage, compared to the existing knowledge that supports the use of simpler linear models [9]. Hence, the investigation of ZR for ANN model selection contributes both to the research on Meta-Learning and to research on ANNs for time series forecasting.

In order to verify the viability of our proposal, we implemented a prototype used to select the following ANN models:

1. TDNN (Time Delay Neural Network) [10]: it corresponds to a feedforward network with time delays in the connections. The input layer receives a fixed time window of the series at hand (i.e. a fixed number of past values of the series), in order to forecast future values of the series;

2. Time-Lagged RBF (Radial Basis Function) [11]: it corresponds to a traditional RBF neural network in which the input layer receives a time window of the series at hand (as in the TDNN model);

3. SOMTAD (SOM with Temporal Activity Diffusion) [12]: it corresponds to a SOM network that creates temporally correlated neighborhoods in the output space.

The candidate ANN models were used to forecast benchmarking time series related to financial, micro and macroeconomic domains, available in the Time Series Data Library (TSDL) repository[1]. In the following sections, we provide a more formal description of the proposed solution, as well as, the details of the implemented prototype.

4.1 Time Series Features

In this proposed solution, a series Z is described by a vector of p features $x = (x^1, ..., x^p)$ in which each $x^j (j = 1, \dots, p)$ corresponds to the value of a different time series feature.

We highlight here that there is no standard set of features to describe time series. However, some criteria can be followed to define the time series features. First, we should prefer features that can be reliably identified, avoiding subjective analysis, such as visual inspection of plots. According to [2], subjective feature extraction is time consuming, requires expertise, and has a low degree of reliability. Another aspect is to choose features that have already been used in the literature to describe the type of time series at hand. Finally, it is interesting to use a manageable number of attributes to avoid adding an excessive amount of time on the model selection process.

Following the above recomendations, we used in the implemented prototype $p = 5$ features to describe the TSDL series:

1. Length of the time series (*L*): number of observations of the series;

2. Basic Trend (*BT*): slope of the linear regression model. As higher this feature value, higher is the global trend of the series;

3. Test of Turning Points (*TP*): Z_t is a turning point if $Z_{t-1} < Z_t > Z_{t+1}$ or $Z_{t-1} > Z_t < Z_{t+1}$. The presence of a very large number or a very small number of turning points indicates that the series is not generated by a purely random process;

4. Average Coefficient of Autocorrelation (*AC*): average of the first 5 autocorrelation coefficients. Large values of this feature suggest a strong correlation between adjacent points in the series;

[1] http://www-personal.buseco.monash.edu.au/~hyndman/TSDL

5. Type of the time series (*TYPE*): it is represented by 3 categories indicating the series domain, *finances, micro-economy* and *macro-economy*.

The first four features are directly computed using the series data and TYPE in turn is an information provided by the TSDL repository. As this set is possibly not optimal, in future implementations we will consider new features.

4.2 Generation of Meta-Examples

The training set for the Zoomed-Ranking is generated from a set of time series, where each series is used to generate a different meta-example.

Formally, let $E = \{e_1, \ldots, e_n\}$ be a set of n meta-examples, in which each meta-example stores: (1) the descriptive features of a time series; and (2) the accuracy and execution time, obtained by K candidate models in the series forecasting. A meta-example $e_i \in E$ is defined as a vector $e_i = (x_i, S_i, T_i)$. In this vector, $x_i = (x_i^1, \ldots, x_i^p)$ is the description of the series; $S_i = (S_i^1, \ldots, S_i^K)$, where $S_i^k(k = 1, \ldots, K)$ is the accuracy of the model k in the i-th series; and $T_i = (T_i^1, \ldots, T_i^K)$, where $T_i^k(k = 1, \ldots, K)$ is the execution time of model k for the i-th series. In the implemented prototype, $S_i = (S_i^1, S_i^2, S_i^3)$ and $T_i = (T_i^1, T_i^2, S_i^3)$ respectively represent the forecasting error and execution time obtained by the models TDNN, RBF and SOMTAD in a single series.

Given a time series used to generate a meta-example, the accuracy and execution time of each model are colected by performing a hold-out experiment using the avaliable time series data. Initially, the time series data is divided in two parts: the fit period and the test period. The test period in our prototype corresponds to the last observation of the series and the fit period corresponds to the remaining data. The fit period is used to train the ANN models. The trained models are then used to generate its individual forecasts for the test period. Finally, each meta-example $e_i = (x_i, S_i, T_i)$ is then composed by: the time series features (x_i) extracted for describing the fit period, the absolute forecasting error (S_i) obtained by the models in the test period and the time execution (T_i) recorded during the ANN models training.

As it will be seen in the section 5, the process described above was applied in 80 different time series, and hence, a set of 80 meta-examples was generated. We highlight that such set of meta-examples actually stores the experience obtained from empirically evaluating the ANN models to forecast a large number of different time series. A Meta-Learning approach (as the ZR) will be able to use this experience to recommend rankings of models for new series only based on the time series features, without the need of performing new empirical evaluations.

4.3 Zoomed-Ranking

The Zoomed-Ranking (ZR) meta-learning approach was used in our prototype to rank the three ANN models. This approach is composed by two distinct phases: the Zooming and the Ranking phases, described here. Given a new time series to forecast, in the Zooming phase, a number of m meta-examples are retrieved from the training set according to a distance function that measures the similarity between time series features. The distance function implemented in the prototype was the L_1-norm defined as:

$$dist(x, x_i) = \sum_{j=1}^{p} \frac{|x^j - x_i^j|}{max_i(x_i^j) - min_i(x_i^j)} \qquad (1)$$

In this equation, x is the description of the input series to be forecasted and x_i is the description of the i-th series in the training set. We used in the implemented prototype the L_1-norm as originally proposed in the ZR approach [6]. In future work, other similarity measures will be investigated.

In the Ranking phase, a ranking of models is suggested, by aggregating the forecasting error and execution time stored in the m retrieved meta-examples. This is performed by deploying the Adjust Ratio of Ratios (ARR) measure [17], as defined in the equation:

$$ARR_{k,k'}^{i} = \frac{\frac{S_i^{k'}}{S_i^k}}{1 + AccD * log(\frac{T_i^k}{T_i^{k'}})} \qquad (2)$$

The metric $ARR_{k,k'}^{i}$ combines forecasting error and execution time, to measure the relative performance of the models k and k' in the series i. The parameter $AccD$ is defined by the user and represents the relative importance between forecasting accuracy and execution time. $AccD$ assumes values between 0 and 1. The lower is the $AccD$ parameter, the higher is the importance given to accuracy relative to execution time.

The ratio of forecasting errors $S_i^{k'}/S_i^k$ can be seen as a measure of advantage of the model k in relation to model k', that is, a measure of relative benefit of model k (the higher is $S_i^{k'}/S_i^k$, the lower is the forecasting error of model k relative to model k'). In turn, the ratio of execution times $T_i^k/T_i^{k'}$ can be seen as a measure of disadvantage of the model k in relation to model k', that is, as measure of relative cost of model k. The ARR measure uses the ratio between a benefit and a cost measure to compute the overall quality of the candidate model k related to k'.

An aspect that should be observed regarding the time ratio is the fact that this measure has a much wider range of possible values than the ratio of accuracy rate. Therefore, if simple ratios of time were used, it would dominate the ARR measure. In this way, the effect of this range could be minimized by using the log of time ratios. We highlight that the use of log of time ratios was also adopted in [6, 17].

Finally, the ranking of models suggested to the input series is generated by aggregating the ARR information across the m retrieved meta-examples and K candidate models, as follows:

$$ARR_k = \frac{\sum_{k' \neq k} \sqrt[m]{\prod_{i \in Zoom} ARR^i_{k,k'}}}{K-1} \qquad (3)$$

In the above equation, the $Zoom$ set represents the m retrieved meta-examples. The geometric mean in ARR is computed across the m retrieved meta-examples and then the arithmetic mean across the candidate models. The ranking is suggested directly from ARR_k (the higher is the ARR_k value, the higher is the rank of model k). The geometric mean was used in order to verify the following property: $ARR_{k,k'} = 1/ARR_{k',k}$. The same would not be true if the arithmetic mean were used.

5 Experiments and Results

In the performed experiments, we collected 80 time series from the Time Series Data Library (TSDL) repository. Hence, a set of 80 meta-examples were generated by applying the procedure described in section 4.2. This set was divided into 60 meta-examples for training and 20 meta-examples for testing the ZR approach.

The experiments were performed for different values of: (1) $AccD$ parameter (0, 0.2, 0.4 and 0.6), which controls the relative importance of accuracy and time; and (2) the parameter m (1, 3, 5, 7, 9, 11 and 13 neighbors), which defines the neighborhood size in the Zooming phase.

In order to evaluate the performance of ZR, we deployed the Spearman Ranking Correlation coefficient (SRC). Given a series i, the SRC coefficient measures the similarity between the recommended ranking of models and the ideal ranking (i.e. the correct ordering of models taking into account the ARR measure computed in the series), as defined bellow:

$$SRC_i = 1 - \frac{6 * \sum_{k=1}^{K} (rr_{k,i} - ir_{k,i})^2}{K^3 - K} \qquad (4)$$

In the equation, $rr_{k,i}$ and $ir_{k,i}$ are respectively the rank of model k in the recommended ranking and the ideal ranking for the series i. SRC_i assumes values between -1 and 1. Values near to 1 indicate that the two rankings have many agreement positions and values near to -1 indicate disagreement between the rankings. In order to evaluate the rankings generated for the 20 series in the test set, we calculated the average of SRC across these series.

The ZR approach was compared to a default ranking method [21], in which the ranking is suggested by aggregating the performance information for all training meta-examples, instead of using only the most similar ones. Despite its simplicity, the default method has been used as a basis of comparison in different case studies in the literature of Meta-Learning [5, 6, 17, 19, 21].

Table 1. Average SRC coefficient across the 20 series in the test set.

	Average SRC			
	AccD = 0.0	**AccD = 0.2**	**AccD = 0.4**	**AccD = 0.6**
m = 1	0.45	0.47	0.50	0.45
m = 3	0.47	0.50	0.52	0.50
m = 5	0.47	0.50	0.52	0.50
m = 7	0.47	0.50	0.52	0.50
m = 9	0.62	0.50	0.52	0.50
m = 11	0.67	0.70	0.67	0.50
m = 13	0.67	0.65	0.62	0.45
Default	0.02	0.05	0.07	0.05

The Table 1 shows the average values of SRC across the test series, considering the ZR approach and the default ranking. As it can be seen, the rankings recommended by ZR were in average more correlated to the ideal rankings when compared to the default method. In fact, the SRC average values for the default ranking are near to zero, indicating neutrality related to the ideal rankings. The ZR in turn obtained SRC values from 0.45 to 0.70, for all different experimental settings, indicating positive correlation to the ideal rankings.

6 Conclusion

In this work, the Zoomed-Ranking approach was used to rank ANN models for time series forecasting. The application of Zoomed-Ranking in the model selection problem is an advance compared to the previous approaches since it is able to automatically acquire knowledge to select models, considering accuracy and execution time as performance criteria.

We evaluated the viability of Zoomed-Ranking for model selection, focused on the selection of ANN models. We implemented here a prototype for ranking and selecting three different ANN models, and performed experiments on a test set of benchmarking time series.

The experiments performed with the implemented prototype revealed that the rankings provided by Zoomed-Ranking were significantly correlated to the ideal rankings. As future work, we highlight the use of new distance functions in the Zooming phase, the inclusion of new candidate models in the forecasting process and more refined experiments to evaluate the proposed solution.

Although the Zoomed-Ranking technique considers forecasting error and execution time as criteria to suggest models, other performance measures could be applied (for instance, different measures of forecasting error). Hence, in a future work, we intend to develop a more general multi-criteria Meta-Learning approach, that will consider a varying number of performance measures. In future work, we also intend to include in the experiments other ANN models for time series forecasting.

Acknowledgments: The authors would like to thank CNPq (Brazilian Agency) for its financial support.

References

[1] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," *International Journal of Forecasting*, vol. 16, pp. 437–450, 2000.

[2] M. Adya, F. Collopy, J. Armstrong, and M. Kennedy, "Automatic identification of time series features for rule-based forecasting," *International Journal of Forecasting*, vol. 17, no. 2, pp. 143–157, 2001.

[3] B. Arinze, "Selecting appropriate forecasting models using rule induction," *Omega-International Journal of Management Science*, vol. 22, no. 6, pp. 647–658, 1994.

[4] A. R. Venkatachalan and J. E. Sohl, "An intelligent model selection and forecasting system," *Journal of Forecasting*, vol. 18, pp. 167–180, 1999.

[5] P. M. dos Santos, T. B. Ludermir, and R. B. C. Prudêncio, "Selection of time series forecasting models based on performance information," in *Proceedings of the 4th International Conference on Hybrid Intelligent Systems*, 2004, pp. 366–371.

[6] C. Soares and P. Brazdil, "Zoomed Ranking - Selection of classification algorithms based on relevant performance information," in *Lecture Notes in Computer Science*, vol. 1910, pp. 126–135, 2000.

[7] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Journal of Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.

[8] R. V. C. Giraud-Carrier and P. Brazdil, "Introduction to the special issue on meta-learning," *Machine Learning*, vol. 54, no. 3, pp. 187–193, 2004.

[9] R. B. C. Prudêncio and T. B. Ludermir, "Selection of models for time series prediction via meta-learning," in *Proceedings of the Second International Conference on Hybrid Systems*. IOS Press, 2002, pp. 74–83.

[10] K. J. Lang and G. E. Hinton, "A time-delay neural network architecture for speech recognition," Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-DS-88-152, Dec. 1988.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan College Publishing Company, 1994.

[12] J. Principe, N. Euliano, and S. Garania, "Principles and networks for self-organization in space-time," *Neural Networks*, vol. 15, pp. 1069–1083., 2002.

[13] R. Leite and P. Brazdil, "Predicting relative performance of classifiers from samples," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[14] A. Kalousis and T. Theoharis, "Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection," *Intelligent Data Analysis*, vol. 3, no. 5, pp. 319–337, 1999.

[15] A. Kalousis, J. Gama, and M. Hilario, "On data and algorithms - understanding inductive performance," *Machine Learning*, vol. 54, no. 3, pp. 275–312, 2004.

[16] H. Bensusan and K. Alexandros, "Estimating the predictive accuracy of a classifier," in *Proceedings of the 12th European Conference on Machine Learning*, 2001, pp. 25–36.

[17] P. Brazdil, C. Soares, and J. daCosta, "Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.

[18] G. Tsoumakas, D. Vrakas, N. Bassiliades and I. Vlahavas, "Lazy adaptive multicriteria planning," in *Proceedings of the 16th European Conference on Artificial Intelligence*, 2004, pp. 693–697.

[19] R. B. C. Prudêncio, and T. B. Ludermir, "Selecting and ranking time series models using the NOEMON approach," *Lecture Notes in Computer Science*, vol. 2714, pp. 654–661, 2003.

[20] R. B. C. Prudêncio, T. B. Ludermir, and F. A. T. de Carvalho, "A modal symbolic classifier to select time series models," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 911–921, 2004.

[21] R. B. C. Prudêncio and T. B. Ludermir, "Meta-learning approaches to selecting time series models," *Neurocomputing*, vol. 61, pp. 121–137, 2004.