Selection of Time Series Forecasting Models based on Performance Information

Patrícia Maforte dos Santos, Teresa Bernarda Ludermir and Ricardo Bastos Cavalcante Prudêncio
Centro de Informática, Universidade Federal de Pernambuco
PO Box: 7851
50732-970, Recife – PE Brazil
Phone: +55 8132718438
E-mail: {pms,tbl,rbcp}@cin.ufpe.br

Abstract

In this work, we proposed to use the Zoomed Ranking approach to rank and select time series models. Zoomed Ranking, originally proposed to generate a ranking of candidate algorithms, is employed to solve a given classification problem based on performance information from previous problems. The problem of model selection in Zoomed Ranking was solved in two distinct phases. In the first phase, we selected a subset of problems from the instances base that were similar to the new problem at hand. This selection is made using the k-Nearest Neighbor algorithm, whose distance function uses the characteristics of the series. In the second phase, the ranking of candidate models was generated based on performance information (accuracy and execution time) of the models in the series selected from the previous phase. Our experiments using the Zoomed Ranking revealed encouraging results.

Keywords: time series forecasting, meta-learning, ranking.

1. Introduction

A time series is a set of observations of a phenomenon ordered in time [2]. Examples of time series include the monthly home electric energy consumption registered for a period of one year or the diary sales of a product in the course of one month. Time series analysis is an identification process for the characteristics, patterns and important properties of a series used to describe in general terms a generator phenomenon [6]. Forecasting is, without doubt, one of the main goals in time series analysis.

Time series forecasting has been used in many real world problems, reducing the risks that arise from uncertainties, as well as helping in planning and decision-making, since the effectiveness of a decision obviously depends on the events that follow afterward.

A number of models have been developed in the literature regarding time series forecasting. Therefore, selecting the most adequate model for forecasting a given series can be a difficult task that depends on the candidate models and the characteristics of the series.

Moreover, there are many methods that facilitate the selection of classification algorithms in the Meta-Learning[1] area. Many of these methods suggest a single algorithm or a group of algorithms that are expected to perform well in the given problem [8] [3]. Thus, the need to obtain a more informative solution arises. We believe that providing a ranking of the candidate algorithms is a more informative and flexible solution. A further advantage of having a ranking of algorithms is to allow the user to select either a single algorithm or more than one in accordance with the available resources (i.e. time and hardware).

In this context, there are currently several methods that generate algorithm rankings based on their past performance information. Some of these methods use accuracy as performance information [4] [12], while

[1] There are many interpretations that can be attributed to the term ‘meta-learning’. In general, meta-learning is the study of how learning systems can increase their efficiency through their own expertise.

others use accuracy and time [14]. However, these methods do not take into account the characteristics that define the problem. Given a new problem, a static ranking is generated based on all available performance information, without regardless of the problem at hand. Zoomed Ranking [15] [5] emerges to solve the drawbacks of these other methods in the selection of classification algorithms.

We propose a Meta-Learning approach, which acquires knowledge from the input data to select time series models. This solution leads to the integration of two distinct areas of knowledge: Time Series Forecasting and Meta-Learning.

2. Zoomed Ranking for selection of models

The Zoomed Ranking approach is composed of two distinct phases. In the first phase, namely the zooming phase, a subset of problems (training series) similar to the new problem at hand is selected from the base of instances. This selection is made using the k-Nearest Neighbors algorithm, whose distance function uses the characteristics of the problem to select the k from previously processed cases that are most similar to the new problem. In the second phase, the ranking of the candidate models is generated based on their performance information (accuracy and execution time) regarding the problems selected from the previous phase.

2.1. Selection of most similar series

As shown in the previous section, the selection of series that are most similar to the new problem is performed in the zooming phase. This phase receives this name because, given a space of previously processed problems, we focus only on the neighborhood of the series that represent the problem at hand. The relevance of a processed series regarding the new series at hand is measured using the set of the meta-attributes of the series.

In this work, the meta-attributes used were the length of the series, basic tendency, percentage of turning points and first auto-correlation coefficient. Some important criteria were added to the definition of these meta-attributes. The first was the choice of characteristics that could be reliably identified, thus avoiding subjective analysis, such as visual inspection on the series graphics. Another aspect is the fact that other authors in the time series literature should have previously used the meta-attributes. Finally, it is necessary to use a manageable number of attributes to avoid spending an excessive amount of time on the model selection process.

The distance between the series is measured using the unweighted L_1 norm [1]:

$$dist\left(v_{x,s_i}, v_{x,s_j}\right) = \sum_x \frac{\left|v_{x,s_i} - v_{x,s_j}\right|}{\max_{k \neq i}\left(v_{x,s_k}\right) - \min_{k \neq i}\left(v_{x,s_k}\right)} \quad (1)$$

where s_i and s_j are the series and $v_{x,si}$ is the value of the meta-attribute x on the series s_i.

The range of possible values for normalization divides this distance. The distance function is used as part of the k-Nearest Neighbors algorithm, the idea of which is simply to select the k nearest cases to a input case given a distance function [10].

2.2. Generation of ranking based on accuracy and time

After selecting the relevant series in the previous section, ranking is generated based on the performance information obtained with the application of the candidate models to these series. As the selected series are similar to the one that represents the new problem, the models are expected to behave similarly when applied to them.

The method of ranking generation used is the Adjust Ratio of Ratios (ARR), which combines information of accuracy and time to generate the ranking of the candidate models [5]. The ARR is defined according Equation 2:

$$ARR_{m_p,m_q}^{s_i} = \frac{\dfrac{SR_{m_p}^{s_i}}{SR_{m_q}^{s_i}}}{1 + AccD * \log\left(\dfrac{1+T_{m_p}^{s_i}}{1+T_{m_q}^{s_i}}\right)} \quad (2)$$

where $SR_{m_p}^{s_i}$ and $T_{m_p}^{s_i}$ represent the success rate and the time of the m_p model on the series s_i, respectively, and AccD, which will be better explained later, is defined by the user and represents the relative importance between accuracy (success rate) and time. The sucess rate was obtained from the error (Mean of the Absolut Error) of the candidate models on the forecasting of the training series, that is, is the inverse of this measure.

The ARR method used the ratio between a benefit and a cost measure to calculate the overall quality of the candidate model. The ratio of success rates $SR_{m_p}^{s_i}$ /

$SR^{s_i}_{m_q}$ can be seen as a measure of advantage of the m_p model in relation to the m_q model, that is, as a benefit. Similarly, the ratio of times $1 + T^{s_i}_{m_p} / 1 + T^{s_i}_{m_q}$ can be seen as a measure of disadvantage of the m_p model in relation to the m_q model, that is, as a cost. The values of $T^{s_i}_{m_p}$ and $T^{s_i}_{m_q}$ were added to 1 in our adaptation of ARR method, because in our case study, which will be explained in details in Section 3, there are a great number of zero values for these measures, making it impossible to use the original formula. Another 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 success rate. Therefore, if a simple ratio of time were used, it would dominate the ARR method. In this way, the effect of this range could be minimized using log ($1 + T^{s_i}_{m_p} / 1 + T^{s_i}_{m_q}$), which yields the order of magnitude of this ratio.

The relative importance between accuracy and time can be obtained with the AccD parameter, as mentioned above. The user gives the value of this parameter and it indicates how much accuracy is compromised in speeding up the model ten fold. Finally, the value 1 is added to yield values that vary around 1, as with the success rate.

The ranking of the candidate models is generated using the ARR method of information aggregating, that is, the value of the ARR method is calculated to each candidate model and the ranking is generated directly from these values. This can be achieved in the following way: First we calculate the geometric mean across all selected series and then we calculate the arithmetic mean across all models, as in Equation 3:

$$ARR_{m_p} = \frac{\sum_{m_q} \sqrt[n]{\prod_{s_i} ARR^{s_i}_{m_p,m_q}}}{m} \qquad (3)$$

where n represents the number of series and m represents the number of models. The ranking is derived directly from this measure applied to each model.

The following definitions are used in the Zoomed Ranking pseudo-code algorithm:

- TEST_SERIES is the set that contains the performance information of the test series;
- TRAINING_SERIES is the set that contains the performance information of the training series;
- MODELS is the set of the candidate models;
- *s* is a test series;
- zooming(*s*) is the application of the zooming phase to the series *s*;
- ArgMinimum (*Distance, n*) is the function that returns the series with the n-th smallest distance;
- Distance[*s1*] contains the distance from the *s* test series to the s1 series of the training set;
- Selected[*n*] contains the n-th most similar series to s. It has Z positions;
- dist(s1, s2) is the *dist* measure of the s1 series relative to the s2 series, which was previously presented in Equation 4;
- Ranking[*m1*] contains the position of m1 model in the recommended ranking;
- ARR (m1) is the ARR measure of the m1 model, which was previously defined in Equation 6;
- recommended_ranking() return the recommended ranking to the *s* test series after the application of zooming(s);
- Z corresponds to the number of selected series in the zooming phase.

```
For all si ∈ TEST_SERIES {
  zooming(si,Z);
  recommended_ranking();
}

zooming(si,Z) {
  Init Distance with zero;
  For all sj ∈ TRAINING_SERIES
         Distance[j] = dist(si, sj);
  For all i ∈ {1, Z}
    Selected = ArgMinimun(Distance,Z);
}

recommended_ranking() {
  For all mk ∈ MODELS
    Ranking[k] = ARR(mk);
}
```

Figure 1. Zoomed Ranking algorithm

2.3. Generation of ranking based on accuracy and time

Our system was implemented according the architecture presented in Figure 2:

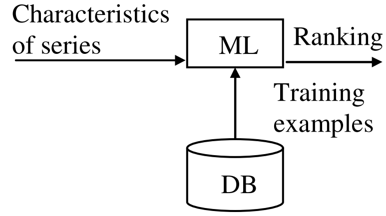

Figure 2. System architecture

In this architecture, the DB module represents the database that contains the meta-examples previously processed by the candidate models. Each meta-example associates a time series (represented by a set of characteristics) to the performance of the candidate models during the series forecasting. The ML module represents the meta-learner, that is, the module responsible for automatically generating knowledge to facilitate the selection of the candidate models. This module receives the characteristics of the series as meta-example input.

In the training phase, the ML module acquires knowledge from the set of meta-examples stored in the DB. This knowledge associates the characteristics of the time series in the training set, which are contained in the meta-examples, with the performance obtained by the candidate models on these series. The knowledge acquired can be refined as new examples are inserted in the base.

In the use phase, given a new test series for forecasting, the ML module uses the characteristics of this series to retrieve the most similar meta-examples contained in the instances base. The performance information contained in the selected meta-examples is used for the ML module to generate the ranking of the candidate models.

3. Experimental results

The base of series used in our experiments was extracted from the M3-Competiton [9], containing 645 yearly time series. From these series, 430 were used for the training of the algorithm and 215 were used for testing. This database is a standard benchmark dataset used to foreacasting[2]. The series are related to economic and demographic domains and represent an adequate sample for performing the experiments [13]. In our experiments, we used the following commonly used candidate models to forecast the M3-Competition time series: Random Walk [6], Holt's Linear Exponential Smoothing [7] and the Auto-Regressive model [2].

The meta-learner of the ML module is an adaptation of the Zoomed Ranking discussed in Section 2. This algorithm was implemented in the C language, using the Microsoft Visual C++[3] tool.

The meta-examples stored in our DB contain the performance information of the candidate models in each training series. This information was obtained in a previous work [12] with the implementation of three candidate models in the Matlab[4].

Our experiments were performed with three values for the *AccD* parameter (1%, 20% and 40%). The 1% value represents the situation in which more importance to accuracy is given. The 40% value represents the inverse situation, that is, more importance is given to time. The 20% value represents an intermediate situation. We used three values for the z parameter (20, 50 and 200), which represents the quantity of series that are selected for the *k-NN* algorithm in the zooming phase. We will denote the use of zooming with z equals k followed by the application of the *ARR* as $Z_k(ARR)$.

Table 1. Ranking without the zooming phase

AccD	1%		20%		40%	
Rank	Algorithm	ARR	Algorithm	ARR	Algorithm	ARR
1	Holt's Smoothing	1.47	Random Walk	1.09	Random Walk	2.19
2	Auto Regressive	1.02	Holt's Smoothing	1.07	Auto Regressive	0.88
3	Random Walk	0.79	Auto Regressive	0.93	Holt's Smoothing	0.82

Table 1 shows the ranking generated for *ARR*, that is, the ranking generated using all available series in the database. This result can be seen as a standard result that will be used to compare the improvement of the *ARR* with the use of the zooming phase. It can be seen that when more importance is given to accuracy, Holt's Smoothing model presents the best results, as it is the slowest model. However, when more importance is given to time, the Random Walk model tops the ranking for being the fastest model.

Table 2. Ranking generated for Z_{20}(ARR)

AccD	1%		20%		40%	
Rank	Algorithm	ARR	Algorithm	ARR	Algorithm	ARR
1	Holt's Smoothing	1.19	Holt's Smoothing	1.36	Random Walk	2.53
2	Random Walk	1.00	Random Walk	0.86	Auto Regressive	0.97
3	Auto Regressive	0.73	Auto Regressive	0.78	Holt's Smoothing	0.65

Table 2 presents the result of $Z_{20}(ARR)$ which show a small variation in the position of the models regarding the rankings of the Table 1. We observe that, even with the application of the zooming phase, the Holt's Smoothing model continues in the top of the ranking when is given more importance to the accuracy. The same occurs with the Random Walk model, which continues in the top of the ranking when is given more

2 International Institute of forecasters: http://www.ms.ic.ac.uk/iif/data/m3comp/m3comp.htm

3 Microsoft Visual C++ is a registered trademark of the Microsoft Corporation. We used the 6.0 version of this tool.

4 Matlab, the Language of Technical Computing, is a tool from The MathWorks, Inc. We used the 6.1 version of this tool.

importance to the time, that is, to the speed of the model.

Table 3. Ranking generated for Z_{200}(ARR)

AccD	1%		20%		40%	
Rank	Algorithm	ARR	Algorithm	ARR	Algorithm	ARR
1	Holt's Smoothing	1.74	Random Walk	1.52	Random Walk	3.63
2	Random Walk	1.43	Holt's Smoothing	0.85	Auto Regressive	1.09
3	Auto Regressive	0.66	Auto Regressive	0.59	Holt's Smoothing	0.49

When we increase the quantity of selected series in the zooming phase, as can be seen in Table 3, the results are similar to previous results. There is a small variation in just the ranking where *AccD* equals 20%. However, *AccD* equal to 20% is only an intermediate value. Thus, no greater importance is given either to the time or the precision.

4. Evaluation of the recommended ranking

The evaluation of the recommended ranking for Zoomed Ranking was performed with the use of Spearman's rank correlation coefficient [11], which measures the distance from the recommended ranking to the ideal ranking. The ideal ranking corresponds to the correct ordering of the candidate models for a given test series. For each test series, an ideal ranking is generated from the performance information of the candidate models in this series. Thus, the ideal ranking of the s_i series is constructed for the ordering of the m_p model regarding each m_q model, as in Equation 4:

$$\sum_{m_q} ARR^{s_i}_{m_p,m_q} / m \qquad (4)$$

where m is the number of models and $ARR^{s_i}_{m_p,m_q}$ is calculated based on the performance information of the models in the test series.

Spearman's rank correlation coefficient is defined according to Equation 5:

$$r_s = 1 - \frac{6\sum_{i=1}^{m}(rr_i - ir_i)^2}{m^3 - m} \qquad (5)$$

where rr_i and ir_i are the ordering of the m_i model on the recommended and ideal ranking, respectively, and m is the number of models.

This measure is normalized for the value $m^3 - m$ to generate more significant values.

The value 1 represents perfect agreement between rankings, whereas –1 represents perfect disagreement. The value 0 means that the rankings are not related. In this way, values near to 1 indicate that the ranking have many agreement positions and values near to –1 indicates that the rankings have many disagreement positions.

Table 4. Mean of Spearman's coefficients

	z = 20	z = 50	z = 200	Sem zooming
AccD	r_s médio	r_s médio	r_s médio	r_s médio
1%	0.102	0.058	0.123	0.065
20%	0.219	0.181	0.205	0.237
40%	0.391	0.426	0.549	0.563

Each recommended ranking was compared with its corresponding ideal ranking using Spearman's rank correlation coefficient. Table 4 displays the means of all Spearman's rank correlation coefficients calculated for each of the tested configurations. We observe that the results obtained when more importance is given to time (*AccD* = 1%) are better than the results obtained without the application of zooming. Such results are due to the way we had chosen the attributes of the series. The attributes were selected taking precision into account meanly.

5. Conclusions and Further Work

In this work, we proposed the use of a Meta-Learning approach that acquires knowledge from the input data to select time series models. We can point out the contributions of our work in two different areas, namely Meta-Learning and Time Series Forecasting. The Meta-Learning approach used was an adaptation of the Zoomed Ranking method originally proposed for the selection of classification algorithms. This method uses the performance information of the candidate models on the previously processed series in the generation of the rankings for these models. However, the rankings are generated from the information on the series most similar to the new series at hand, which are selected in the zooming phase. This selection is made using the *k-NN* algorithm.

The results were obtained with the variation of the k parameter, which represents the number of neighbors of the *k-NN* algorithm, and of the *AccD* parameter, which represents the relative importance between accuracy and time. These results are compared with the standard *ARR*. The results obtained are better than the results of the standard *ARR* in many configurations. This difference becomes even greater when more importance is given to accuracy. This occurs because the meta-attributes that describe the series were selected taking accuracy into account.

Another improving of this work could be making the choice of the parameter AccD more automated. In this process, the user could simply give the percentage that represents this parameter to the system.

In future work, we could obtain better results using another base of cases with a greater number of instances and one that it does not present many values equal to zero for the execution time of the models, as occurred with the base used in the present work. We could also avoid zero execution time by using other forecasting models for which a greater difference among the speeds can be observed.

It is possible to use other methods in the zooming phase, such as a neural network, to measure the similarity between the series, substituting the KNN method.

Other methods of ranking generation will be implemented and the results compared with Spearman's rank correlation coefficient used in Zoomed Ranking.

6. References

[1] Atkeson, C. G., A. W. Moore, and S. Schaal, *Localy Wighted Learning*, Vol. 11, pp. 11-74, Kluwer, 1997.

[2] Box, G. E. & Jenkins, G. M., 'Time Series Analysis: Forecasting and Control', *Holden-Day*, San Francisco, CA, 1970.

[3] Brazdil, P., C. Soares, 'Characterizing the applicability of classification algorithms using meta-level learning', In: F. Bergadano and L. de Raedt, editors: *Proceedings of the ECML-94*, pages 83-102, Springer-Verlag, 1994.

[4] Brazdil, P., C. Soares, 'A Comparison of Ranking Methods for Classification Algorithm Selection', *Machine Learning: Proceedings of 11th European Conference on Machine Learning (ECML2000)*, R.López de Mántaras and E.Plaza (Eds.), LNAI 1810, Springer Verlag, 2000.

[5] Brazdil P., C. Soares and J. Pinto da Costa, 'Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results', *Machine Learning*, 2003, 50(3) 251-277, Kluwer.

[6] Harvey, A. C., *Time Series Models*, MIT Press, Cambridge, MA, 1993.

[7] Holt, C. C., Modigliani, F., Muth, J. F. and Simon, H. A., *Planning Production Inventories and Work Force*, Prentice-Hall, Englewood Cliffs, NJ, 1960.

[8] Kalousis, A. and Theoharis, T., 'NOEMON: Design, Implementation and Performance Results of an Intelligent Assistant for Classifier Selection', *Intelligent Data Analysis*, Vol. 3, no. 5, pp. 319-337, 1999.

[9] Makridakis, S. and Hibon, M., 'The M3-Competition: Results, Conclusions and Implications', *International Journal of Forecasting*, Vol. 16, n. 4, pp. 451-476, 2000.

[10] Mitchell, T., *Machine Learning*, MacGraw Hill, New York, NY, 1997.

[11] Neave, H and P. Worthington, *Distribution-Free Tests*, Routledge, 1992.

[12] Prudêncio, R. B. C and Ludermir, T. B. ' Meta-learning approaches to selecting time series models, Neurocomputing, in press, 2004.

[13] Shah, C., 'Model Selection in Univariate Time Series Forecasting Using Discriminant Analysis', *International Journal of Forecasting*, Vol. 13, pp. 489-500, 1997

[14] Soares, C. M. O. P., 'Ranking Classification Algorithms on Past Performance', M.Sc. Dissertation, Faculdade de Economia, Universidade do Porto, 1999.

[15] Soares, C., P. Brazdil, 'Zoomed Ranking: Selection of Classification Algorithms based on Relevant Performance Information', In: D. Zighed, J. Komorowski, J. Zytkow editors: *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2000)*, Springer, pp. 126-135, 2000.