APPLYING PROPOSITIONAL LEARNING TO TIME SERIES PREDICTION

Luis Torgo

LIACC
R.Campo Alegre, 823 - 2o.
4100 PORTO
PORTUGAL
Telf. : (+351) 2 600 16 72 - Ext. 115
Fax : (+351) 2 600 3654
e-mail : ltorgo@ncc.up.pt

Keywords : Time series prediction, propositional learning systems, introduction of new attributes, empirical evaluation.

Abstract

In this paper we present some work on trying to apply propositional learning systems to the problem of time series prediction. We describe the difficulties of trying to adapt classification systems to prediction problems. Some techniques to perform this adaptation are presented. These techniques are based on the introduction of new attributes that convey relevant information about the temporal behavior of a time series. These strategies are evaluated on some time series problems. Results indicate that the predictive performance strongly depends on the used set of attributes and that the best set is domain-dependent. Automatic strategies for finding the optimal set of attributes are sketched out. Experiments were carried out in order to compare our performance to the one obtained by other types of methods in the same data sets. The results compare reasonably to other methods, although indicating that more work needs to be done before machine learning-based systems can compete with other techniques on prediction problems.

1. INTRODUCTION

The symbolic machine learning (ML) field has defined some problems as the fundamental goals of its research. Among them are classification problems, clustering, discovery, etc. The problem of prediction is clearly not among them. There are few works in the ML field on this subject. Among the few exceptions are some papers on predicting discrete sequences of symbols [4, 10]. However, these works do not address the particular case of time series prediction, so they have a quite different scope from the work presented here.

The field of time series prediction is considered a key problem in several scientific areas and has been receiving increasing attention from several academic and non-academic institutions. Several fields related to machine learning like Neural Nets and Genetic Algorithms have already been working on this subject (see for instance [11, 13 and 18]). It is our believe that the symbolic machine learning area could add something to this scientific topic.

The main goal of this paper is to try to adapt an existing propositional learning system to the problem of time series prediction. We used the regression tree learner M5 [14, 15] in our experiments because it is able to perform numeric classification. In time series we want to predict future values of a

numeric variable. This was one of the reasons for choosing the M5 system. M5 builds a decision tree with linear models in the leaves acting like a kind of piecewise linear regression model.

The main motivation for trying to use symbolic systems on these tasks is that they make it easy to include background knowledge to help the learning task. We argue that this inclusion could improve performance in prediction problems. Nevertheless, at the present stage of our work we still do not explore this issue.

The main difficulty of this adaptation is that prediction problems are different from classification problems that are the target problems addressed by the majority of ML systems. On prediction problems we have issues like time dependencies among data (autocovariance and autocorrelation of a stochastic process [7]) which are not directly mapped into the propositional learning framework. Our work is currently concentrated on this adaptation. We outline a technique based on feature selection that enables the use of ML systems on prediction. Feature selection is needed because if we use few features to describe the data we might be loosing important information on the temporal behavior of the data. On the contrary if we learn with too many features we may overfit the noise. The results of our experiments indicate that the choice of attributes is relevant to the obtained performance. This conclusion is in accordance to other work [3, 6] on testing the sensibility of tree learners to the used set of attributes.

2. ADAPTING ML SYSTEMS TO PREDICTION PROBLEMS

We face two basic problems when trying to use ML systems on prediction tasks. First the variable being predicted in a time series is usually real valued. Most ML systems are not able to deal with numeric classes. Secondly, the time series model assumes that there exists some form of correlation between successive values of the variable being predicted. Assuming that this is relevant information we need to enable ML systems to learn theories that somehow relate class values to previous class values.

2.1 REPRESENTING A TIME SERIES IN A PROPOSITIONAL LANGUAGE

In classical propositional learning, examples are labeled with class values and are described using a set of attributes. The systems learn theories that in some way explain the observed examples of each class. The resulting theories are general descriptions of the classes. These descriptions are logical relations between the attributes' values.

There is an obvious mapping between the variable being predicted in a time series and the class labels of classification problems. They are the goal of learning. We still have the problem that usually classes are a set of finite discrete labels while the values of the predicted variable are a set of possible infinite continuous values. Numeric classes can be a problem to some learning systems. To overcome this difficulty we either use learning systems that cope with numeric classes or we discretize the class values before the learning phase. Learning systems that deal with numeric classes have the advantage of not loosing any information in a process of discretization. The main disadvantage is that few can do that. Some examples are the regression trees like CART [2], M5 [14, 15] and RETIS [8, 12]. In this paper we follow the strategy of using a learning system (M5) that deals with numeric classes.

In order to use inductive learning systems we have to decide which attributes (features) should we use to describe each observed instance of the goal variable. On section 2.2.1 we will see that we can introduce attributes that store relevant information about previous values of the goal variable.

2.2 TIME DEPENDENCIES

One basic assumption for predicting future values of a time series is that any value of the goal variable depends in some way on the previous values. The usual goal of classical linear time series models is to find a mathematical model that fits the observed past values. The resulting model is usually some kind of relation [17] of the next value to the previous ones.

We need to enable ML systems to learn theories that relate class values to previous values of the class. This somehow runs out of the scope of usual ML problems where we learn relations of class labels to attribute values.

There is an easy solution to this problem. We can pre-process the time series data creating attributes that contain previous class values. With this simple coding scheme we can learn relations among successive values of the goal variable.

2.2.1 Introducing New Attributes

The number of previous values of the goal variable (lagged values [7]) that we include as attributes is an important question as it will determine the type of models learned by the systems. If we include too many previous values we may introduce irrelevant information that can act as a kind of noise to the learning algorithm.

In the experiments described in this paper we have tried several values for the lag.

The introduction of attributes need not be restricted to information on previous class values. We can also include other relations between previous values of the goal variable. For instance if we want to capture information on trends we can add attributes calculated as differences of previous class values. This could be taken further by adding attributes with the value of differences of differences and so on.

Of course many other possible relations can be considered. In our experiments we limit ourselves to previous class values, differences, differences of differences and weighted averages of previous values. This does not mean that we think these relations will be sufficient to deal with any time series problem. They were just a first trial on the issue of feature selection. The next step is doing a kind of automatic testing of the relevance of these new attributes to guide the decision about which ones should be used in the learning process. This issue is discussed in more detail on section 2.2.2.

2.2.2 Selective Attribute Introduction

As said before we have tried attributes expressing different types of relations among previous class values. Using a basic a set of arithmetic functions we can define a kind of feature space. It is our goal to develop an algorithm that searches this space looking for the *ideal* feature set that guarantees the best performance in the given data. This is motivated by our experiments that showed the relation between the used features and the predicting accuracy. They also showed that the *best* set of attributes is domain dependent.

Recently, very interesting works [3, 6, 12] on the subject of deciding which attributes are relevant for a given learning task where presented. These techniques could be applied in our problem to help the decision of which attributes to use for a given time series prediction problem.

3. EXPERIMENTS

We had two main objectives when planning the experiments. The first was to compare and evaluate the different alternatives in pre-processing that were referred in the previous section. The second goal was to evaluate the use of ML systems in prediction problems when compared to other techniques.

3.1 THE DATA SETS

To attain the goals of the experiments we needed data sets for which we could find results of other systems. We decided to use some data sets of a competition held out in DGOR (Deutsche Gesellschaft für Operations Research) [16] in 1982. These problems had the advantage of being well described and presenting lots of results of 12 different competitors. The main disadvantage of these data sets was that they were quite small. In effect the ones we used had around 100 examples for learning. This small number of examples can limit the learning performance of propositional learners, in particular when lots of attributes are present (which is the case in some of our trials where lots of attributes are introduced).

From the 15 time series problems that were used in the DGOR competition we had access to 5. Their names were ZR03, ZR04, ZR06, ZR11 and ZR15. These data sets were basically industry data sets (products sales, etc.).

These datasets needed to be pre-processed according to the methods referred in section 2.2.1. We compared the following strategies of introducing new attributes. We always tried data sets constructed by using 1 to 5 lagged variables. We named these trials with the index "_tN" where N is the used time lag. For instance "ZR03_t4" is the data set built from problem ZR03 using 4 attributes which represent the last four class values. We also tested some variants with differences. They have the suffix "...dN" where N is the level of differences used. For instance "ZR06_t5d2" uses 5 lagged variables and it as also 4 attributes of first level differences and 3 of differences of differences (so it uses 12 attributes). We also included in the tests some variants with weighted averages of previous class values (the ones with "sm" in their name). We did not exhaust all possible combinations. We heuristically chose some of them. For instance the trial "ZR03_smt5d2" uses the same attributes as "ZR03_t5d2" but it has an extra attribute whose value is calculated as $(5*y_{t-1} + 4*y_{t-2} + 3*y_{t-3} + 2*y_{t-4} + y_{t-5})/15$. This is a kind of weighted moving average linear model [7]. There where also other variants with different weights on the previous values of the variable.

3.2 USED LEARNING ENGINE

In our experiments we have decided to use M5 [14, 15]. M5 is a propositional learning system that is able to deal with numeric classes and learns model trees [14]. Model trees differ from regression trees [2] in that they have linear models in their leaves instead of average values. The algorithm for building model trees is similar to the one used in building decision trees. However, M5 is a kind of hybrid algorithm in the sense that it combines instance-based learning with model trees. It can learn an instance-based model of the data, a model tree or a combination of both. In its default mode the system decides which methodology to use for a given problem. In our experiments we used all default parameter values.

3.3 THE EXPERIMENTS

The goal of these experiments was to compare the results of the ML learned models to other types of systems not belonging to the symbolic learning field.

We compare our models to the 12 methods used in the DGOR competition. This comparison was made using three statistics of errors :- the mean square error (MAE), root mean squared error (RMSE) and Theil's coefficient (TU). They are calculated using the following formulas :

$$MAE = \frac{\sum_{i=1}^{N} \left| observation_i - prediction_i \right|}{N} \qquad RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(observation_i - prediction_i \right)^2}{N}}$$

$$TU = \sqrt{\frac{\sum_{i=1}^{N} \left(observation_i - prediction_i \right)^2}{\sum_{i=1}^{N} \left(observation_i - observation_{i-1} \right)^2}}$$

The goal of the experiments was to predict the value of the goal variable for the next 12 time periods. Errors were calculated on the basis of those 12 predictions. In resume for each of our 5 data sets we constructed the data sets of examples for each of the candidate attribute introduction strategy and after learning, we tried to predict those 12 future values using the resulting learned model.

The 12 methods used in the DGOR competition included smoothing average variants, Kalman filters, Box-Jenkins [1] methods, etc. In the table of results we refer to them as m1 to m12.

We summarize the results of the experiments in table 1. This table shows the results on the MAE statistic. The results are ordered showing the best model at the top.

ZR03		ZR04		ZR06		ZR11		ZR15	
m5	9.9	ZR04_t2	0.97	m10	3.28	m8	208	m11	26.8
m12	12.2	ZR04_t3	0.98	m11	3.38	m1	217	m4	28.8
m3	14.8	ZR04_t4	0.98	m7	3.73	m7	222	m8	30.9
m8	16.9	ZR04_t5	0.98	m5	3.97	m11	243	m5	36.5
m11	17.9	ZR04_t2d1	0.98	m12	3.98	m6	251	m6	37.3
m7	19.7	ZR04_t3d1	0.98	m3	4.1	m3	255	m7	40
m9	22.3	ZR04_t4d1	0.98	m8	4.15	m2	257	m1	40.4
m4	25.1	ZR04_t5d1	0.98	m1	4.26	m9	292	ZR15_t1	42.26
m10	25.9	ZR04_t3d2	0.98	m2	4.87	m12	292	m10	44.2
ZR03_t1	26.48	ZR04_t4d2	0.98	m9	5.33	m5	295	m12	45.8
m2	26.5	ZR04_t5d2	0.98	m4	5.65	m10	308	m3	46.8
m1	29.4	ZR04_sm3	0.98	ZR06_t3d1	6.64	m4	317	ZR15_t2d1	48.2
ZR03_t3d1	30.69	ZR04_t4sm4w	0.98	ZR06_t2d1	6.71	ZR11_t4d2	422.27	ZR15_t5	48.44
ZR03_t3d2	30.69	ZR04_t4sm4wv	0.98	ZR06_sm3	6.71	ZR11_t4sm4w	432.85	m2	48.6
ZR03_t2	31.63	m4	1	m6	6.79	ZR11_t4sm4wv	440.84	ZR15_t4	48.74
ZR03_t2d1	32.18	ZR04_t4sm4	1	ZR06_t5	6.83	ZR11_t4sm4	466.3	ZR15_t3d1	48.74
ZR03_smt5d2	32.67	m2	1.08	ZR06_t3d2	6.87	ZR11_t1	471.88	ZR15_t4d1	48.74
ZR03_t4d1	34	m10	1.08	ZR06_t4d1	6.88	ZR11_t2	477.8	ZR15_t5d1	48.74
ZR03_t5d1	34	m11	1.09	ZR06_t5d1	6.98	ZR11_t2d1	477.8	ZR15_t4d2	48.74
ZR03_t3	34.53	ZR04_smt5d2	1.16	ZR06_t5d2	6.98	ZR11_sm3	478.37	ZR15_t5d2	48.74
ZR03_t5d2	34.55	ZR04_t1	1.25	ZR06_t4	7.29	ZR11_t3d1	518.93	ZR15_sm3	48.74
ZR03_t5	35.09	m6	1.29	ZR06_t4sm4wv	7.29	ZR11_t4d1	526.02	ZR15_smt5d2	48.74
ZR03_t4d2	35.27	m5	1.38	ZR06_t3	7.53	ZR11_t5d2	546.57	ZR15_t4sm4	48.74
ZR03_t4	35.45	m12	1.44	ZR06_t4d2	7.58	ZR11_t5d1	579.36	ZR15_t4sm4w	48.74
ZR03_sm3	35.47	m8	1.45	ZR06_t4sm4	7.63	ZR11_t3d2	585.25	ZR15_t4sm4wv	48.74
ZR03_t4sm4wv	35.55	m3	1.54	ZR06_smt5d2	8.17	ZR11_t4	591.29	ZR15_t2	49.84
ZR03_t4sm4w	35.56	m7	1.57	ZR06_t4sm4w	8.19	ZR11_smt5d2	769.3	ZR15_t3	54.01
ZR03_t4sm4	36.81	m9	1.89	ZR06_t2	8.5	ZR11_t3	827.32	m9	58.6
m6	37.9	m1	1.9	ZR06_t1	8.96	ZR11_t5	841.71	ZR15_t3d2	64.82

Table 1. Summary of comparative results with other fields' methods.

The ordering for the other statistics is similar so we omit the tables for space reasons.

As we can see from table 1 the results of our methods vary a lot from domain to domain. They range from the surprisingly good results of M5 on ZR04 to the very bad results on ZR11. This variation also occurs with the other methods as there seem to be no clear winner on all problems. However, our results are not generally ranked in the best positions. This indicates that further research is needed before this ML-based models are ready for this kind of competitions. We expect that by tuning some of the learning parameters of M5 we can improve these results. Automatic feature selection is another technique that can improve our rankings. We should also try other learning algorithms like RETIS.

We have tried to understand the reason for such bad outcome in the ZR11 data. We observed that M5 was using a branch of its learned model tree whose label was not a regression formula but an average value (which can happen in M5). The system was making almost always the same prediction for all 12 values of this data set. This value was an average value which of course would not allow the system to go outside of the scope of the values in the learning set. This clearly indicates that further investigation is needed on either improving the behavior of M5 through parameter tuning, or develop other systems that do not have this behaviour.

4. CONCLUSIONS

We have shown that through some simple pre-processing techniques we can use propositional learning systems like M5 to deal with prediction problems. These systems could bring to the field of time

series prediction the easy incorporation of background knowledge to improve prediction accuracy. This is a key issue that we hope to elaborate in future work.

Although no extensive variations where tried the results of ML-learned models compare reasonably to other more sophisticated methods.

The results of the experiments carried out are very domain-dependent. They are also influenced by the set of used attributes. Automatic methods for controlling attribute introduction are needed. We outlined a strategy for doing this based on work on feature set selection. With variants of these methods we hope to control the introduction of new attributes.

The analysis of the results of these experiments could lead to the development of a symbolic learner more adapted to prediction tasks.

Acknowledgments

I would like to thank Prof. Pavel Brazdil for is support as well as for reading previous versions of this paper. I would also like to thank Prof. Ross Quinlan for supplying the M5 system.

References

[1]. Box,G., Jenkins,F. (1976) : *Time Series Analysis : Forecasting and Control*. Oakland, CA. Holden-Day, 1976.

[2]. Breiman,L. , Friedman,J.H., Olshen,R.A. & Stone,C.J. (1984): *Classification and Regression Trees*, Wadsworth Int. Group, Belmont, California, USA, 1984.

[3]. Caruana,R., Freitag,D. (1994): Greedy Attribute Selection. In *Proceedings of 11th IML (International Machine Learning Conference)* . Cohen,W. & Hirsh,H. (eds.). Morgan Kaufmann, 1994.

[4]. Dietterich,T., Michalski,R. (1985) : Discovering Patterns in Sequences of Events. In *Artificial Intelligence, 25-2*. North-Holland, 1985.

[5]. Gershenfeld,N., Weigend,A. (1993) : The Future of Time Series: Learning and Understanding. In Time Series Prediction : Forecasting the Future and Understanding the Past. Weigend,A. & Gershenfeld,N. (eds.). Addison Wesley. 1994.

[6]. John,G., Kohavi,R., Pfleger,K. (1994): Irrelevant Features and the Subset Selection Problem. In *Proceedings of 11th IML (International Machine Learning Conference)* . Cohen,W. & Hirsh,H. (eds.). Morgan Kaufmann, 1994.

[7]. Judge,G.G, Hill,R.C., Griffiths,W.E, Lutkepohl,H & Lee,T. (1988): *Introduction to the Theory and Practice of Econometrics (2nd edition)*. John Wiley & Sons, 1988.

[8]. Karalic, A.. (1991): The Bayesian Approach to Tree-Structured Regression. In *Proceedings of ITI-91*. Cavtat, Croatia, 1991.

[9]. Karalic, A..(1992): Employing Linear Regression in Regression Tree Leaves. In *Proceedings of ECAI-92*. Wiley & Sons, 1992.

[10]. Laird,P., Saul,R. (1994) : Discrete Sequence Prediction and Its Applications. In *Machine Learning,* 15-1 (p. 43-68). Kluwer Academic Publishers, 1994.

[11]. Meyer,L. Packard,N. (1992) : Local Forecasting of High-Dimensional Chaotic Dynamics. In *Nonlinear Modelling and Forecasting*. Casdagli,M. & Eubank,S. (eds.). Addison-Wesley, 1992.

[12]. Moore,A., Lee,M. (1994): Efficient Algorithms for Minimizing Cross Validation Error. In Proceedings of 11th IML (International Machine Learning Conference) . Cohen,W. & Hirsh,H. (eds.). Morgan Kaufmann, 1994.

[13]. Mozer,M. (1993) : Neural Net Architectures for Temporal Sequence Processing. In *Time Series Prediction : Forecasting the Future and Understanding the Past*. Weigend,A. & Gershenfeld,N. (eds.). Addison Wesley. 1994.

[14]. Quinlan, J.R. (1992): Learning with Continuos Classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. Singapore: World Scientific, 1992.

[15]. Quinlan, J.R. (1993): Combining Instance-Based and Model-Based Learning. In *Proceedings of 10th IML (International Machine Learning Conference)*. Utgoff,P.(ed.). Morgan Kaufmann. 1993.

[16]. Schwarze,J., Weckerle,J. (1982) : *Prognoseverfahren im Vergleich*. Braunschweig, 1982.

[17]. Weigend,A., Gershenfeld,N. (1994) : *Time Series Prediction : Forecasting the Future and Understanding the Past*. Addison Wesley. 1994.

[18]. Weigend,A., Huberman,B., Rumelhart,D. (1992) : Predicting Sunspots and Exchange Rates with Connectionist Networks. In *Nonlinear Modelling and Forecasting*. Casdagli,M. & Eubank,S. (eds.). Addison-Wesley, 1992.