Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Theoretical and practical approaches for time series prediction

Alina Bărbulescu, Dana Simian

Abstract

The goal of this paper is to discuss two different modern approaches for modeling and prediction of time series – general regression neural networks and support vector regression. It is known that the performances of different approaches from machine learning field are strongly dependent on data. We apply and evaluate our methods on eight different real meteorological series. In order to increase the SVR performances we develop a method for obtaining a SVR optimal multiple kernel.

1 Introduction

Modeling the time series evolution is of main importance for the prediction of real life processes as temperature, precipitation, earthquakes, due to their economic and social implications, that could be dramatic (drought, disasters, famine etc). The main problems in modeling and prediction of this type of series are their non – linearity, high variability, correlation and/or persistence, making them inappropriate for the use of classical regression methods. Therefore, new methods, belonging to machine learning, artificial intelligence and optimization techniques, as artificial neural networks (ANN), gene expression programming (GEP), support vector regression (SVR) can be successfully used for this aim. It is known that generally a SVM based approach is strong dependent of data type. In [3] we have studied the problem of forecasting the meteorological time series. We used for this purpose an adaptive GEP algorithm, AdaGEP, and a ε-SVR algorithm with RBF kernel and we performed an empirical comparison of these methods on many series of temperature and precipitations from different meteorological stations in the Black Sea region. The comparison revealed that there is not one method with best results for all studied data series, but AdaGEP dominates SVR models in the most cases. It would be ideal to find a method capable to be used with almost the same level of performances for meteorological series obtained from any stations and for all variables taken into account (temperatures, precipitations etc.). This is a difficult task and it is not sure that it can be accomplished.
The aim of this paper is to analyze two models obtained for meteorological time series using General Regression Neural Networks (GRNN) and SVR. We detect models and we evaluate their performances for six different real meteorological series, using DTREG software [6]. We also propose a theoretical approach for improving the performances of SVR. This is a method for the choice of a multiple SVR kernel such that the measure of the prediction error is minimized. We consider two such kinds of measures: Mean Absolute Prediction Error and Mean Squared Error.

2 Problem model

2.1 Time series prediction

The problem we face with is the following one: given a set of measured values for some meteorological characteristics (temperature, precipitations) in a period of time, predict the future values based on the past values. Practical a number *d* of past values is chosen to predict the future one. The choice of *d* is not the object of this article. Usually additional information and experiments are necessary to accomplish this task.

The classical problem of time series prediction is: find a function *f* which predicts future values, in a given prediction horizon *p*, of the series $X_t = (x_t, x_{t-\tau}, ..., x_{t-(d-1)\tau})$, i.e. express $x_{t+p} = f(X_t)$.

The series studied are annual and monthly precipitations and temperature data collected at four main meteorological stations from Dobrudja region, Romania, presented in Table 1.

Series	Station	Type	Variable	Period
CAT	Constanta	Annual	Temperature	01.1965 – 12.2005
TAT	Tulcea	Annual	Temperature	01.1965 – 12.2005
SAT	Sulina	Annual	Temperature	01.1965 – 12.2005
JAT	Jurilovca	Annual	Temperature	01.1965 – 12.2005
CMP	Constanta	Monthly	Precipitation	01.1961 – 12.2009
SMP	Sulina	Monthly	Precipitation	01.1965 – 08.2007
CMT	Constanta	Monthly	Temperature	01.1961 – 12.2009
SMT	Sulina	Monthly	Temperature	01.1961 – 08.2003

Table 1. Data series

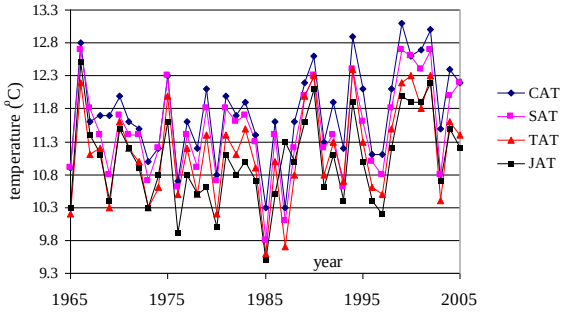

Fig.1. Series of annual temperatures

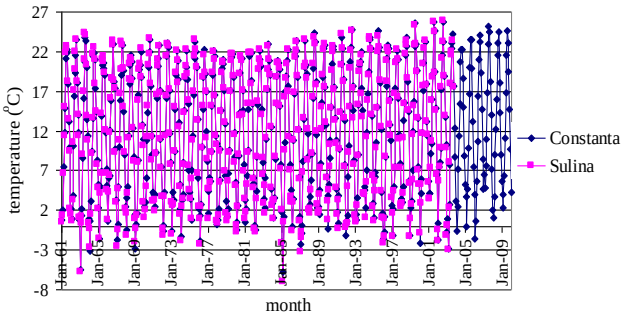

Fig.2. Series of monthly temperatures

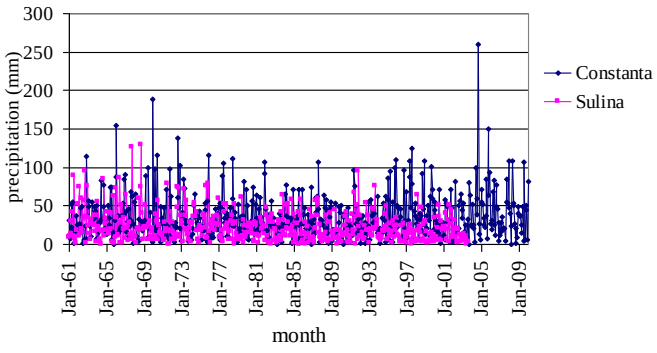

Fig.3. Series of monthly precipitation

The charts of series of annual temperatures are presented in Fig.1, those of monthly temperatures in Fig. 2 and of monthly precipitation in Fig. 3.

2.2 General Regression Neural Networks (GRNN)

GRNN introduced by Specht [14] is a feedforward network that allows the estimation of a dependent vector Y= (Y_i) from an independent vector **X**= (X_j) obtained by measurements. GRNN have the same architecture as the Probabilistic Neural Networks (PNN). The difference is that PNN act on categorical target variables and GRNN act for continuous target variables. That means that PNN performs classification while GRNN performs regression. PNN estimate the probability density function f(**X**,Y) for each class based on the training samples using Parzen or a similar probability density function. In both cases (PNN and GRNN) the operations are organized into a multilayered feed-forward network with four layers: Input layer, Pattern layer, Summation layer, Output layer [13]. In the case of GRNN the input layer contains one neuron for each predictor variable Y_i. The pattern layer contains one neuron for each case in the training data set, and stores the predictor variables values and target values [2]. The summation layer computes the numerator and the denominator of the estimator using two neurons: the numerator and the denominator summation neurons. Output layer contains one neuron that contains the result of division of the values in the numerator and the denominator of the previous layers. Both GRNN and PNN use nonparametric estimators of probability density function. The measure of how well each training sample can represent the position of prediction is the Euclidian or the city block distance between the training sample and the point of prediction [14].

2.3 Support Vector Regression (SVR)

SVR is a category of Support Vector Machines (SVM). SVM represents a powerful tool for solving learning tasks like classification and regression tasks. They are supervised learning methods introduced first by Vapnik [15]. The goal of SVM is to build a model, *f*, which predicts the output of a system depending of a set of variables, using a set of training data for which the output is known. The main characteristic of SVM is that the prediction function is expanded on a subset of support vectors as will be seen in relation (1). Support Vector based algorithms were extended from classification tasks to regression ones using various loss functions [1]. Traditional statistical regression techniques aim to minimize the deviation of *f(x)* from the known outputs for all training examples. The ε - SVR introduced in [15] uses the so called ε – insensitive loss function. It minimizes the generalized error bound instead of minimizing the observed training error, being based on the structural risk minimization principle. ε - SVR is searching for a function *f* that has at most ε deviation from the target outputs on all the training data and is as flat as possible. These requirements lead to a convex optimization problem. Next we present this formulation following the presentation from [15,1].
We consider first the case of a linear function *f:*

$$f(x) = \langle w, x \rangle + b;\ b \in \mathbb{R}, x \in X$$

where we denoted by *X* the space of all input instances and $\langle \cdot, \cdot \rangle$ represents the dot product in *X*.

Suppose that the training data are denoted by $(x_i, y_i) \subset X \times \mathbb{R}$, *i* =1, ..., *m*.

To take into account the possibility of an infeasible convex optimization problem we introduce the slack variables ξ, ξ^* and the problem formulation becomes:

$$\text{minimize } \left\{ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \right\},$$

subject to the constraints:

$$\begin{cases} y_i - \langle w, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

The dual optimization problem is

$$\text{maximize } \left\{ -\frac{1}{2}\sum_{i,j=1}^{m}\left(\alpha_i-\alpha_i^*\right)\left(\alpha_j-\alpha_j^*\right)\left\langle x_i,x_j\right\rangle-\varepsilon\sum_{i=1}^{m}\left(\alpha_i+\alpha_i^*\right)+\sum_{i=1}^{m}y_i\left(\alpha_i+\alpha_i^*\right)\right\}$$

subject to the constraints:

$$\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)=0 \text{ and } \alpha_i,\ \alpha_i^*\in[0,C]$$

The function *f* can be rewritten as:

$$f(x)=\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)\left\langle x_i,x\right\rangle+b \tag{1}$$

Relation (1) represents the so - called support vector expansion of the function *f*. The examples corresponding to non-vanishing coefficients are called Support Vectors.
The constant $C > 0$ and ε are parameters of the method. An improved SVR technique, named ν – SVR considers ε itself as a variable in the optimization process introducing a new parameter $\nu\in(0,1)$ (see [1]). The new parameter is more convenient than ε.

In order to solve the non-linear problem we make a projection ϕ of the input data X in a higher dimensional feature Hilbert space F. Using the "kernel trick" we can rewrite (1) for a non - linear function, without knowing explicitly the form of the mapping ϕ :

$$f(x)=\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)K(x_i,x)+b \tag{2}$$

The kernel function *K* represents the inner product in the feature Hilbert space *F*. Several kernels can be used: linear, polynomial, RBF, sygmoidal (see [12]). These kernel functions are defined in Table 7. Other functions satisfying the Mercer's conditions can be used as kernel functions ([13]).

2.4 Settings and performance evaluation

For performance evaluation of the model obtained using GRNN and SVR we divided the data into two parts, one for training and one for validation. The last one contained 10 values for the annual series and 24 values for the monthly ones. Both algorithms were applied using the default settings of DTREG software [6], the number of predictor variables being 1 for the annual data and 2 for the monthly data. We implemented the ν – SVR, with RBF kernel.
The models performance have been analysed using as indicators the Mean Squared Error (MSE) and the Mean Absolute Prediction Error (MAPE) defined by:

$$MSE=\frac{\sum_{i=1}^{n}(x_i-x_i^*)^2}{n} \tag{3}$$

$$MAPE=\frac{\sum_{i=1}^{n}\left|\frac{x_i-x_i^*}{x_i}\right|}{n} \tag{4}$$

where x_i is the *i* - th registered value in the data series and x_i^* is the *i* - th value predicted by the model. We can observe that the MSE is a scale-dependent accuracy measure while the MAPE is scale independent. For models resulted by GRNN application, the correlation between the actual and the predicted value was also reported.

3 Results and discussions

The MSE for the models on the training and validation dataset are presented in Tables 2 and 3. For the first dataset, GRNN performed better than SVM. We remark the big differences of MSEs corresponding to GRNN and SVR on the training sets for TAT, SAT, JAT and CMP. For the validation datasets, MSE are comparable in the case of CAT, JAT. The SVR algorithm performs better on validation data in the case of the series TAT, JAT and SMP.
The MAPE for the models on the training and validation dataset are presented in Tables 4 and 5.
For the validation datasets, MAPE are generally comparable for GRNN and SVR. GRNN performs better in the case of CAT, CMP, CMT and SMT. In the case of training data GRNN significantly outperforms SVR.

Series	GRNN	SVR
CAT	0.066	0.39
TAT	0.000148	0.31
SAT	0.0006	0.27
JAT	0.034	0.39
CMP	6.45	793.85
SMP	327	373.02
CMT	4.08	5.79
SMT	3.95	5.6

Table 2.

MSE for the models on the *training* datasets

Series	GRNN	SVR
CAT	0.434	0.45
TAT	0.99	0.49
SAT	0.503	0.93
JAT	0.518	0.46
CMP	887.17	899.93
SMP	105.09	97
CMT	1.59	3.27
SMT	9.27	11.75

Table 3.

MSE for the models on the *validation* datasets

The correlations between the actual and predicted values in the experiments performed with GRNN are presented in Table 6. GRNN performed better on the training datasets and in five of eight cases on the validation datasets. The exceptions are for CMT and SMT, for which the registered performances are very high

Series	GRNN	SVM
CAT	0.482	4.14
TAT	0.021	3.76
SAT	0.041	4.39
JAT	0.322	4.16
CMP	5.55	267.22
SMP	244.33	209.62
CMT	82.45	107.99
SMT	49.14	60.34

Table 4.

MAPE for the models on the *training* datasets

Series	GRNN	SVM
CAT	4.8	4.88
TAT	6.87	5.12
SAT	7.58	5.15
JAT	5.52	5.21
CMP	580.34	594.75
SMP	247.09	197.5
CMT	24.36	35.78
SMT	244.03	250.94

Table 5.

MAPE for the models on the *validation* datasets

Series	Training		Validation	
	GRNN	SVR	GRNN	SVR
CAT	0.913101	0.207317	0.312878	0.31407
TAT	0.999859	0.623083	-0.394385	0.250404
SAT	0.893635	0.605498	0.250289	-0.247702
JAT	0.958707	0.183927	0.332102	0.349899
CMP	0.996158	0.237336	0.078309	0.049084
SMP	0.394746	0.256092	0.206757	0.250852
CMT	0.96722	0.953139	0.986575	0.971232
SMT	0.969537	0.956431	0.968031	0.961067

Table 6. Correlations between actual and predicted values

We observe that the performances are highly dependent of data. The size and the structure of the training set strongly influences the modeling and forecasting.
Taking into account that SVR underperforms on the validation data set in most of the cases we conclude that SVR overfit the training examples. Possible causes of overfitting phenomenon could be the choice of parameters C and ν or the choice of the kernel. The choice of parameters C and ν in SVR was performed doing a grid search in a 10-fold cross-validation procedure in order to avoid the overfitting [7]. This leads us to the conclusion that the use of a single kernel is not capable to generate an accurate model and more complex kernel must be used.

A possible solution for obtaining best results using SVR for time series prediction is proposed in the next section.

4 Optimal SVR multiple kernels for time series prediction

In [10] we introduced a general frame for building optimal kernels for Support Vector Classification. We implemented many particular methods derived from this frame for data sets Leukemia and Vowel from the standard libsvm package [4] and the results were promising.
In the following we propose a new method for obtaining optimal multiple SVR kernels for time series prediction. The idea is to create a SVR model based on a multiple kernel in order to obtain better prediction results. Multiple kernels are built using the simple standard kernels presented in Table 7 and the set of operations {+, *, *exp*} which preserve Mercer's conditions (see [9] for many details on Mercer's Theorem).

Kernels	
Polynomial: $K_{pol}^{r,d}(x_1, x_2) = (x_1 \cdot x_2 + r)^d; r, d \in \mathbb{Z}_+$	(5)
RBF: $K_{RBF}^{\gamma}(x_1, x_2) = \exp\left(\frac{-1}{2\gamma^2}\lvert x_1 - x_2\rvert^2\right); \gamma \in \mathbb{R}_+$	(6)
Sigmoidal: $K_{sig}^{\gamma}(x_1, x_2) = \tanh(\gamma \cdot x_1 \cdot x_2 + 1); \gamma \in \mathbb{R}_+$	(7)

Table 7. Standard single kernels

Following the models implemented in [10] we proposed a multiple kernel composed by 4 single kernels, but our approach is not restricted to this number of single kernels. The tree formal representation of the multiple kernel K= (K_1 op_2 K_2) op_1 (K_3 op_3 K_4), is given in Fig. 4.

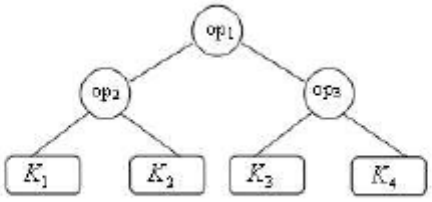

Fig. 4. – Formal representation of multiple kernel [10]

The choice of a multiple kernel supposes the choice of the type of the single kernels and the choice of the corresponding parameters.

To construct the multiple kernels we use an evolutionary method structured on two levels: the macro and the micro levels. In the macro level the multiple kernel is built using a genetic algorithm. The multiple kernels are coded into chromosomes. The chromosomes quality is computed in the micro level using a SVR algorithm. The main difference from the classification case is that we have to additional constants to be appropriately chosen: C and ε (or ν depending on the SVR approach). We propose two solutions for the choice of these parameters in the end of this section.

As structural representation of the multiple kernel we use a linear representation given in Fig.5. For coding the multiple kernel we use 78 genes: 6 genes for operations (2 genes for each operation op_i, i = 1, 2, 3); 6 genes for kernel types (2 genes for each type t_i, i = 1, 2, 3); if the single kernel K_i is polynomial we use 4 genes for the degree parameter d_i and 12 genes for r_i; if the single kernel K_i is not polynomial we use 16 genes to represent the real value v_i..

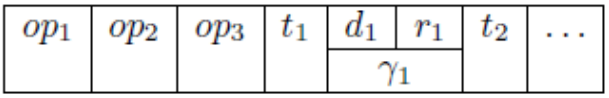

Fig. 5. Linear representation of multiple kernel [10]

In order to evaluate the quality of the chromosomes we use a SVR algorithm acting on a particular set of data. In order to make the evaluation we divide the dataset into two subsets: the training subset and the test subset. The training subset is used for problem modeling and the test subset is used for evaluation. The training subset is also randomly divided in two subsets: the learning and the validation subsets. The learning subset is used for training the SVR algorithm and the validation subset is used for computing the Mean Absolute Prediction Error (MAPE) defined in (4), which represents the fitness function for evaluation of chromosomes.

The performance of our predictive model based on the multiple kernel given by the genetic algorithm is evaluated using cross validation method.

There is two possibility for the choice of parameters C and ε (or ν) from the SVR algorithm. The first one is to adapt a grid choice of parameters which imply to run our method for each pair in the grid and then to chose the best triplet (C, ε, optimal multiple kernel). This method is huge time consuming in the training step. Due to the independence of the computation in each point of the grid it could give good results in the case of parallelization. The second method includes the

two parameters in the chromosome representation such that a choice of a multiple kernel is made together with a choice of the pair (C, ε).

Implementation and practical tests are necessary in order to validate our theoretical proposed method.

5 Conclusions and further directions of work

In this article we studied the performance of modeling meteorological time series using two different approaches – GRNN and SVR. It has been seen that the results are strong dependent on datasets. In terms of Mean Squared Error, GRNN performed better in all cases for the training sets and in five from eight cases for validation sets. Since we needed an invariant measure of performances, we also used Mean Absolute Prediction Error. In this case, GRNN performed better in seven from eight cases for the training sets and in four cases for validation sets. For two series (CMT and SMT) the correlation between predicted and actual values is very good both for training and validation datasets. For the other six series the correlation is under 0.5 and in the case of TAT validation data set for GRNN and SAT validation data set for SVR we obtained a negative correlation. The results suggest us the necessity of a more complex model, if possible a flexible one, able to fit well to different series of meteorological data. The proposed theoretical solution is a new SVR approach using an optimal multiple kernel. The optimal multiple kernel is obtained using an evolutionary algorithm structured on two levels. Further works will be oriented to the implementation and testing of this method on different meteorological datasets in order to validate it.

References

[1] D. Basak, S. Pal, D. C. Patranabis, Support Vector Regression, Neural Information Processing – Letters and Reviews, Vol. 11, No. 10, pp. 203 – 224, 2007.

[2] A. Bărbulescu, L. Barbeş, Mathematical models for inorganic pollutants in Constanța area, Romania. Revista de chimie vol. 64(7), pp. 747 - 753, 2013.

[3] E. Băutu, A. Bărbulescu. Forecasting meteorological time series using soft computing methods. Appl. Math. Inf. Sci., 7, No. 4, pp. 1297 – 1306, 2013.

[4] C-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[5] H. Drucker, C. J.C. Burges, L. Kaufman. A. Smola, V. Vapnik, Support Vector Regression Machines, Neural Information Processing Systems, C.J. Burges, L. Kauffman, A. Smola, V. Vapnik (Eds.), MIT Press, pp. 155 – 161, 1997.

[6] DTREG, Software For Predictive Modeling and Forecasting, http://www.dtreg.com/

[7] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003.

[8] H. N. Nguyen, S. Y. Ohn, W. J. Choi, Combined kernel function for support vector machine and learning method based on evolutionary algorithm, Neural Information Processing, 11th International Conference, ICONIP 2004, N. R. Pal et al. (Eds.), LNCS, vol. 3316 , Springer, pp. 1273 – 1278, 2004.

[9] Ha Quang Minh, Partha Niyogi, Yuan Yao, Mercers Theorem, Feature Maps, and Smoothing, http://people.cs.uchicago.edu/ niyogi/papersps/MinNiyYao06.pdf

[10] D. Simian, F. Stoica, A general frame for building optimal multiple SVM kernels, Lecture Notes in Computer Science, Vol. 7116, Large Scale Scientific Computation, I. Lirkov et al. (Eds.), pp. 256 – 263, 2012.

[11] D. Simian, F. Stoica, C. Simian, Optimization of Complex SVM kernels using a hybrid algorithm based on wasp behaviour, Lecture Notes in Computer Science, vol. 5910, I. Lirkov, S. Margenov, and J. Wasniewski (Eds.), pp. 361 – 368, 2010.

[12] A. J. Smola, B.Schölkopf, A tutorial on support vector regression, Statistics and Computing, 14, Kluwer Academic Publishers, pp. 199 – 222, 2004.

[13]D. F. Specht, "Enhancements to Probabilistic Neural Networks", International Joint Conference on Neural Networks, vol. I, pp. 761 – 768, June 1992.
[14]F. Sprecht. A General Regression Neural Network. IEEE Transactions on Neural Networks, vol. 2, no. 6, pp. 568 – 576, Nov. 1991.
[15]V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, 1995, http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets

BĂRBULESCU Alina
Doctoral School of Civil Engineering
Technical University of Civil Engineering
Bucharest
ROMANIA
alinadumitriu@yahoo.com

SIMIAN Dana
Lucian Blaga University
Sibiu
ROMANIA
dana.simian@ulbsibiu.ro