MODELLING AIR POLLUTION BY NEURAL NETWORK AND SUPPORT VECTOR REGRESSION WITH PARAMETER SELECTION

R. BÉCZI[1], I. JUHOS[2] and L. MAKRA[1]

[1]Department of Climatology and Landspace Ecology, University of Szeged,
H-6701 Szeged, P.O.B. 653, Hungary
E-mail: beczir@geo.u-szeged.hu; makra@geo.u-szeged.hu;
[2]Department of Informatics, University of Szeged,
H-6701 Szeged, P.O.B. 652, Hungary, E-mail: juhos@inf.u-szeged.hu;

ABSTRACT – The so-called "inductive learning algorithms" in the field of artificial intelligence can be well applied to the solution of automated and adaptable regression problems and, hence, to the assessment of time series, as well. Forecasts were made by using artificial neural networks, as mostly used method recently, as well as the related support vector regression techniques. These methods are able to perform proper non-linear function fitting, which essential in case of practical non-linear assessment problems. If we combine the methods mentioned above, we can get more precise decisions for the future data. In either case, the efficiency of learning depends on a good choice of the learning algorithms' parameters. For this reason, parameters are selected by simulated annealing. The aim of this paper is to compare the above mentioned prediction techniques in several hours forecast of NO concentrations at a busy cross-road in Szeged (Hungary). For this object, meteorological parameters predicted with given error on their actual values were used.

1. INTRODUCTION

Considering the special meteorological and geographical conditions of Szeged, the dispersion of air pollutants – especially during permanent anticyclone weather conditions in the summer and winter seasons – is extremely slow. A reliable forecast of concentrations of the air pollutants is highly important. Methods of classical statistics as well as methods of neural networks were already used to give short term forecast of various gases and particulate matter. *Gardner and Dorling* (1988) [1] give an excellent account on the applications of neural network methods for forecasting in atmospheric sciences. *Jorquera et al.* (1998) [2] compare a linear model and a fuzzy model as prediction tools of daily maximum ozone concentrations. *Perez et al* (2000) [3] presents an application of neural networks for a few hours prediction of $PM_{2.5}$ in the atmosphere of Santiago city (Chile). *Ziomas et al* (1995) [4] analyses the possibility of forecasting maximum ozone concentrations in Athens city. He applied discriminant analysis to forecast possible increase and decline of NO_2 levels. He considered the following parameters: previous daily maximum ozone concentration; forecasted temperature; wind velocity and direction; an index of the given day's short term emission change; an index of the effect of the precipitation on the given day. In average, 80 % of the forecasts were successful. In this paper, different forecasting methods, the multi layer perceptron and the support vector regression, are compared. Furthermore, they are used to forecast hourly averages of NO concentrations. The parameters that our estimations are based on are NO concentrations from the previous day, wind velocity, temperature and humidity.

2. INDUCTIVE LEARNING OF ATMOSPHERIC PARAMETERS

We apply two different inductive learning techniques to give estimations of future atmospheric parameters such as NO concentrations. Here we briefly recall the Multi Layer Perceptron model and the ν-SVR algorithm that we used.

Inductive learning of a concept means recognizing a hypothesis regarding this concept after presenting the training instances to the learner. The simplest learning case is that, where one part of the training instances is true (positive) and another part is false (negative). A subset of the instances can be regarded as a function, namely as the characteristic function of the subset. The domain of this function consists of the instances, while the values are either true or false (0 or 1) according to the instances belonging to the subset or not. During the training process, the instances are generally represented in the following format:

$$\underset{instance}{x_1, x_2, \ldots, x_n,} \quad \underset{class}{y}$$

where x_i is the i-th attribute of the instance and y is the class of the instance (true or false). A training instance and its class is a training example. In order to find the inductive hypothesis, a function of $y = h(x_1, x_2, \ldots, x_n)$ based on the training instances have to be approximated. The number of the classes can be extended to more than two; thus, the problem can be generalized to the classification into more than two discrete classes or to the learning of functions with not discrete range. Goodness of the hypothesis h can be determined by applying it on the not presented instances (which were not in the training set).

The estimate of the atmospheric parameter corresponds to an inductive learning model. Past data are the instances, and the forecast of the data will be determined by an inductive hypothesis. The accuracy of the learning depends on the number and on the accuracy of the training data (data can come from real process by measurement), while the quality of the learning (the finding of the inductive hypothesis) depends on the chosen learning algorithm.

2.1 Multi Layer Perceptron (MLP)

While the one-layered MLP is capable of approximating continuous functions (Hornik et al., 1989) [5], the two-layered MLP is capable of approximating arbitrary finite sets of real numbers (Chester, 1990) [6]. Thus we choose the latter an regard the number of neurons in each layer as parameters of the learning mechanism. We use the sigmoid activation function (1), and both the input and the output layers have linear units.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

When the l pieces of attributes of the i^{th} learning instance takes the form $(x_{i1}, x_{i2}, \mathrm{K}, x_{il})$ then the output of the s^{th} Perceptron in the 1st layer is given by (2) and the output result is given in (3).

$$o_i^{1s} = \sigma\left(\sum_{t+1}^{l} w_t^{1s} x_{it} + w_{bias}^{1s}\right) \text{ (output of the } s^{th} \text{ Perceptron for the } i^{th} \text{ instance).} \tag{2}$$

$$y_i = \sum_{r=1}^{l_2} w_r \sigma\left(\sum_{s=1}^{l_1} w_s^{2r} o^{1s}{}_i + w_{bias}^{2r}\right) \tag{3}$$

w_t^{1s}, w_m^{2r}, w_r and $w_{bias}^{1s}, w_{bias}^{2r}$: weights in the 1st and 2nd layers and output unit and biases.

l_1, l_2: number of perceptrons in 1st and 2nd layers.

The well-known backpropagation method with momentum is used for adjusting the weights during the training process[1]. Backpropagated MLP learning depends on the following parameters that need to be tuned: number of neurons in the hidden layers, learning rate, momentum and number of training epochs.

2.2 Support Vector Regression (SVR)

There are two commonly used Support Vector Machines for regression the ε-SVR algorithm and its extension the ν-SVR algorithm (see Schölkopf et al, 1998) [7]. We chose the ν-SVR, because it has an advantage in contrast with ε-SVR, being able to automatically adjust the width of the ε-tube around the function being approximated. An SVR maps the $\mathbf{x} = (x_{i1}, x_{i2}, \mathrm{K}, x_{il})$ instances to a usually higher dimension space, called feature space by a $\phi : ¡^{l} \to ¡^{L}, L \geq l$ function. Then it makes a linear fit according to (5) with some precision by optimizing the weights $\mathbf{w} = (w_1, w_2, \mathrm{K}, w_L)$, w_{bias}.

$$y_i \approx \sum_{j=1}^{L} w_j \phi(x_{ij}) + w_{bias} \quad (= \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + w_{bias}) \tag{5}$$

The aim is to find $\mathbf{w}$ and w_{bias} such that $\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + w_{bias}$ approximates y_i best possible with respect to the "distance", the so-called ε-insensitive loss function:

$$\max(|f(\mathbf{x}) - y| - \varepsilon, 0).$$

This means that we imagine an ε-tube around the regression and the points of the feature space that lie within this tube are still considered acceptable. We also wish to keep $|\mathbf{w}|$ small, i.e., the regression flat. The ν-SVR is a modified version of this: Minimise the expression

$$\frac{1}{2}|\mathbf{w}|^2 + C\left(\nu\varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right)$$

in $\mathbf{w}, \varepsilon, \xi_i, \xi_i^*$ and subject to the conditions:

$$y_i - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) \leq \varepsilon + \xi_i^*, \ \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) - y_i \leq \varepsilon + \xi_i \text{ and } \xi_i, \xi_i^* \geq 0.$$

To obtain better performance some parameters in the algorithm ν-SVR may be tuned: regularisation parameters C, ν and other parameters of the function ϕ.

2.3 Discussion of the methods

2.3.1 Over-fitting.

A general drawback in machine learning is over-fitting. When the precision of the approximation of the desired function is increased, the generality, i.e., the applicability of the method to data sets largely different from the training set may be destroyed. Thus, in case of the instances taken outside of the training set the sum of the errors increases. Generally, these phenomena may be observed in the later phases of the training process. Using a larger training set or stopping the training process in due time may provide a solution to this problem; nonetheless, there are no exact definitions of the correct stopping time.

2.3.2 Setting of parameters.

Machine learning techniques, as mentioned above, suffer from problems like overfitting.

[1] Provided by Weka library (Witten et al., 2000) [8].

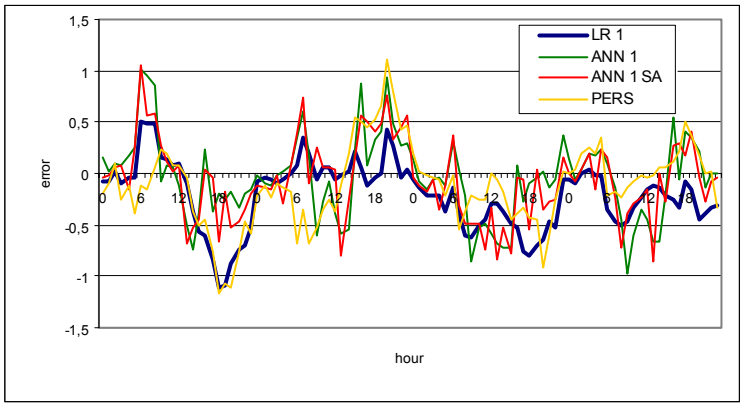
These problems occur also because of not properly chosen learning parameters, e.g., number of neurons in case of neural networks (see Section 2.1). In addition, the prediction tasks can require some other parameters, see, e.g., Section 2.2. The precision of the forecast highly depends on the rightly chosen parameters. Finding the good parameters is hard because of the function which measures their goodness has unknown or bad behaviour from the optimisations point of view, it may not be differentiable or can have many local extrema. Our model selection is based on a validation process. Historical data is divided into training and test sets. The learner uses the training set for making a hypothesis and we validate this on the test set by comparing the hypothesis-provided estimation with the desired real values. Comparison gives a measurement of the goodness. Based on this measurement the model selector can decide acceptance or modification of the parameters and doing again the process in hope of the better solution. Our goodness/fitness measurement function *f* of a **p** parameter vector is the Root Mean Squared Error (RMSE):

$$f(\mathbf{p}) = \sqrt{\sum_{i=1}^{n} \frac{\left(y_i - \hat{y}_i(\mathbf{p})\right)^2}{n}}$$

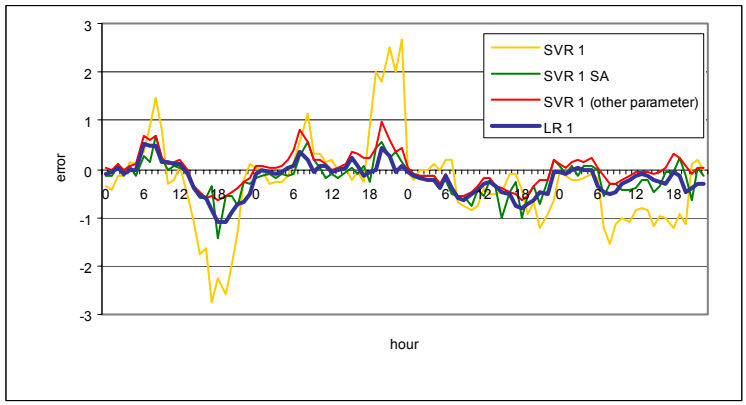
2.3.3 The Simulated Annealing.
The simulated annealing (SA) is a heuristic method of locating the extrema of a function (the terminology is motivated by physical annealing processes). In our case, we are looking for the minimum of the fitness function *f* as described in Section 2.3.2. From the initial point we try to move randomly in the parameter space with varying step size. The algorithm employs a random search that not only accepts changes that decrease objective function *f*, but also some changes that increase it with positive probability. It is an ability to avoid becoming trapped at local optima. Since the algorithm requires no assumption on the shape of *f*, it provides a widely applicable tool. Taking small enough decrease in the temperature at each step, we may get sufficiently close to or reach the optima. Currently we optimize the learner-specific parameters mentioned as tuneable in the Sections 2.1 and 2.2.

3. RESULTS

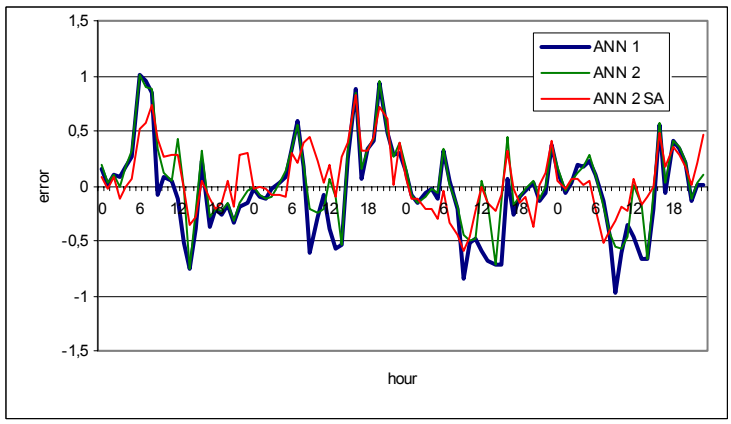
For each 24 hours of a day a different neural network and support vector machine was applied. The aim was to forecast the NO concentrations in a given hour from the previous days already known atmospheric data. We considered three types of estimations. First, we predicted NO concentrations purely from NO concentrations (signed 1 in figures). Second, we took into account certain external factors, as humidity, wind velocity and temperature in a given hour (signed 2 in figures). Finally, we looked whether solely these external factors allow reasonable forecast of NO concentrations (signed 3 in figures). We used as training(learning) set normalized data of September 1. 2000 – March 12. 2001 (every weekday), and the prediction was happened to March 13. 2001 – March 16. 2001 (Tuesday, Wednesday, Thursday, Friday). The figures below depict the error values (RMSE) of the different forecasts for four subsequent four days.
Some abbreviations are ANN (artificial neural network): MLP, LR: linear regression, PERS: persistency, HTW: humidity, temperature and wind velocity. And there are some cases ordered by different learning set (1,2,3).



RMSE(PERS)= 0.408432
RMSE(LR 1)= 0.386363
RMSE(ANN 1)= 0.402882
RMSE(ANN 1 SA)= 0.384444
The different methods give similar results. The SA does not in general improve the results of ANN, on the contrary in certain cases these result are even worse. This is due to the fact that a neural network having better fitness on the test set may give larger errors in the prediction period (this is the over-fitting as described in Section 2.3.1). This can be a general drawback of the SA. The persistency means that the concentration in a given hour of the previous day is considered as the „predicted" value. As can be seen, the neural networks did not achieve much better results than simply the persistency.



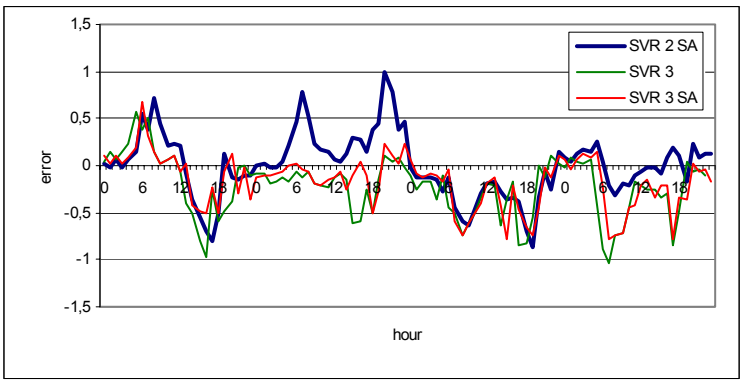
RMSE(SVR 1)= 0.962204
RMSE(SVR 1 SA)= 0.388504
RMSE(SVR 1 with other parameters)= 0.336281 (red)
The first parametrisation of the SVR results in significantly larger errors, but the trend sin the errors is the same. This shows that the proper choice of the parameters influence the quality of the learning to a great extent. The use of SA drastically reduced the mean error, whereas also a disadvantage of the probabilistic search can be seen. It was possible to find manually a slightly better parametrisation than that SA produced. Further when we started from the optimal parameters the probabilistic nature of SA made it possible that the search process *left* the optimum.



RMSE(ANN 2)= 0.35185
RMSE(ANN 2 SA)= 0.29305
The ANN SA 2 reduces the error of the simple ANN 2 in many cases (Tuesdays 5-8), but also sometimes gives worse results (Tuesdays 9-11). In the whole of the prediction period the use of SA resulted in better performance. The ANN 1 and the ANN 2 are quite similar, but the RMSE of the ANN2 is better, which justifies the role of the HTW. The application of SA with ANN 2 improves the results.

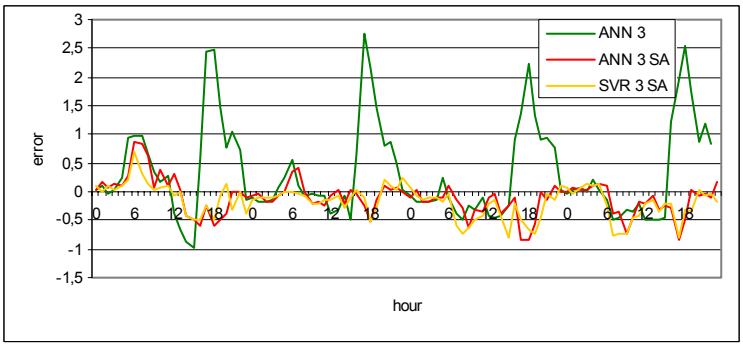
RMSE(SVR 2)= 0.33173 RMSE(SVR 2 SA)= 0.3367

The three additional factors do not improve the SVR 1, indeed the errors are nearly the same. This shows that in the SVR 2 the factors do not play a significant role. Moreover, the SVR 2 was not affected by the SA, the RMSE remained also almost unchanged.



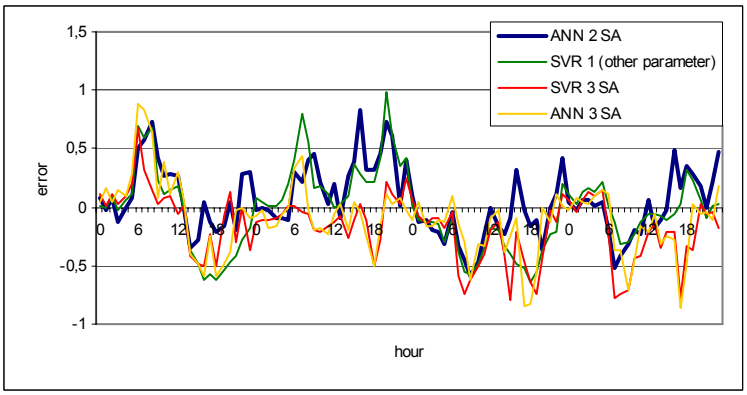
RMSE(SVR 3)= 0.38370
RMSE(SVR 3 SA)= 0.3229

The SVR3 as compared to SVR2 SA reduces the error in general, and most significantly on Wednesday 5-13h and 18-0h, which, in average, comepensate the Friday increase in the error. The use of SA with SVR 2 was highly beneficial, thus among the the SVRs the SVR 3 SA is the most suitable.



RMSE(ANN 3)= 0.869166
RMSE(ANN 3 SA)= 0.32082

The ANN 3 gives extremely bad results, but these are corrected significantly by the SA. It is instructive to observe the similarity between the errors of the ANN 3 SA and the SVR 3.



The four methods with the best RMSE are presented int this figure. In certain cases, nall four give approximately the same errors (Monday, 18h, Thursday 9h). In is not surprising that the two methods based purely on NO (1) and on HTW (3) give different results (Wednesday 12-18h, Friday 17h).

We could not find a method that gives good results in the whole prediction period, but we saw that the combination of the different methods could be very advantageous.

REFERENCES

[1] *Gardner, M.V. and Dorling, S.R.,* 1998: Artificial neural networks (the multilayer perceptron) – a review of applications in atmospheric sciences. *Atmos. Environ., 32,* 2627-2636.

[2] *Jorquera, H., Perez, R., Cipriano, A., Espejo, A., Letelier, M.V. and Acuňa, G.,* 1998: Forecasting ozone daily maximum at Santiago, Chile. *Atmos. Environ.*, *32,* 3415-3424.

[3] *Perez, P., Trier, A., Reyes, J.,* 2000: Prediction of PM 2.5 concentration several hours in advance using neural networks in Santiago, Chile. *Atmos Environ., 34,* 1189-1196.

[4] *Ziomas, I.C., Melas, D., Zerefos, C.S., Bais, A.F. and Paliatsos, A.G.,* 1995: Forecasting peak pollutant levels from meteorological variables. *Atmos. Environ.*, *24,* 3703-3711.

[5] *Hornik, K., Stinchcombe, M. and White, H.,* 1989: Multilayer feedforward networks are universal approximators. *Neural Networks, 2,* 359-366.

[6] *Chester, D.,* 1990: Why Two Hidden Layers are Better Than One. International Joint Conference on Neural Networks, pp. I-265-I-268. Washington, DC.

[7] *Schölkopf, B., Bartlett, P., Smola, A. and Williamson, R.,* 1998: Support Vector Regression with Automatic Accuracy Control, Eighth International Conference on Artificial Neural Networks, pp. 111-116.

[8] http://www.cs.waikato.ac.nz/~ml/