Intelligent Information Systems VII
Proceedings of the Workshop held in
Malbork, Poland, June 15-19, 1998

Predictions in presence of outliers: Which tool to choose?

Anna Bartkowiak and Adam Szustalewicz
Institute of Computer Science, University of Wrocław,
Przesmyckiego 20, Wrocław 51-151
e-mail aba@ii.uni.wroc.pl and asz@ii.uni.wroc.pl

Institute of Computer Science, University of Wrocław, Przesmyckiego 20, Wrocław 51-151
e-mail: aba@ii.uni.wroc.pl or asz@ii.uni.wroc.pl

Abstract. We consider two sets of data known to be benchmarks for regression in presence of outliers. These are: the *stackloss* data and the *Hawkins-Bradu-Kass* data. The statistical methodology stresses the importance of detecting and accounting for the outlying and/or influential data vectors. The neural network methodology hopes that the problem will be tackled automatically by itself. We investigate in more detail the two approaches. We obtain the same goodness of approximation provided that we know about the outliers.

Keywords: Regression, Prediction, Outliers, Radial basis neural network

1 Introduction

Predictions are made usually on the basis of an equation constructed using a regression algorithm (statistical methodology) or more complex convolution formula using "weights", "biases" and "activation functions" (neural network methodology).

It is widely believed that the approach by neural networks is more universal and depends less on specific assumptions on the form of the model underlying the investigated data.

We are doubtful about this belief.

In the following we consider two particular examples known as benchmark data for the problem of regressional predictions. The data sets are known to contain observations which are influential for the constructed regression equation.

Important problems to be considered are:

1. Which criterion to choose for minimizing the error of the approximation? It is known, that the least square criterion is a very bad one.
2. How to establish the model used for predictions, how to choose the architecture of the network?
3. Suppose, the model (its size) is already fixed, how to estimate the parameters appearing in the model?

In last years it has been observed, that many statistical ideas infiltrate the neural network methodology [7, 5, 3].

We will show the results (goodness of the predictions measured by the residual sum of squares) obtained using both the statistical and the neural (radial basis) network approach.

The general conclusion is, that we *may* obtain similar results, when we know about the outliers. Without this knowledge we may have some problems.

It is advisable that the analysis will accompanied with a graphical inspection of the data. In case of multivariate data the grand tour [4] proves to be especially useful.

2 Analysis of the stackloss data

The data contain $n = 21$ data vectors, each comprising values of 3 predictors and 1 predicted variable called stackloss. The data are described a.o. by Atkinson [2, 1]. Four data vector: #1, #3, #4 and #21 are known to be influential for the regression. The multivariate structure of the data may be seen in plots retained when running grand tour. Such plots may be found in [4].

Statistical approach

When performing ordinary computations of the regression equation and looking at the obtained residuals, nothing very special about the 4 mentioned data vectors is found. Their special role may be discovered only by applying some special methods (see, e.g. [2, 1, 6, 4]).

The ordinary LSE method with 3 predictors yields a residual sum of squares $RSS = 178$ with $df = 17$ degrees of freedom. Introducing into the regression equation two extra terms for the presence of #1, #3 and #4 (first term) and #21 (second term), the residual sum of squares drops to $RSS = 22.20$ with $df = 15$ degrees of freedom.

This means an enormous improvement of the fit.

Neural network approach

We have used for the analysis the radial basis network [9, 5] implemented in MatLab [8]. The procedure solverb needs a.o. the following inputs: spread of the radial base functions (*spread*), maximal number of neurons (*max_neur*), and the minimum level of error to be attained (*err_goal*).

Putting as *err_goal* the value $RSS = 22$ obtained from the ordinary LSE method, and taking *max_neur=15* we have run the procedure for *spread* from the range 1:0.5:14. We obtained in all runs, that the number of neurons needed to achieve the presumed accuracy

was *nr=14*. The value of *spread* had no very great impact on the results, although the values RSS were the best for values of *spread* lying somewhere in the range of 1–5.

The procedure works by choosing in the space of the predictors some prototypes of the data vectors. The procedure introduces sequentially one neuron ($\equiv$ one radial basis function centered at one chosen data vector) in each iteration (epoch). These were always the 4 outlying points chosen as basis of the radial functions, and next – after all the 4 selected – some others were chosen too. The decay of the RSS value as a function of the number of the chosen basic neurons is given below (run assuming *spread=5.0*:

RSS	732.0	552.4	250.0	246.9	129.9	125.6	111.7	40.4
# of neurons	1	2	3	4	5	6	7	8
RSS	39.4	39.4	35.2	28.7	27.7	12.3		
# of neurons	9	10	11	12	13	14		

The number of parameters in the model is enormous: each radial neuron is characterized by $p = 3$ parameters of its position. Remind, there are $s1 = 14$ neurons in the model. Additionally, there are $s1$ *weights* and *one bias* necessary to make the final prediction with the accuracy comparable to those by the ordinary LSE method. In the considered case this needs 57 parameters of the neural network model (we have together 84 data values), while using the LSE method we need only 6 parameters.

3 Analysis of the Hawkins–Bradu–Kass data

The data contain $n = 75$ data vectors with $p = 3$ predictors and 1 predicted variables. 10 from the 75 vectors are high leverage points very influential for the regression equation. Additionally there are 4 other high leverage points however *not* influential for the regression. When using the grand tour it may be clearly seen that the data set contains 3 distinct groups of data points [4].

Statistical approach

The ordinary LSE method yields residual sum of squares $RSS = 359.5$ with $df = 71$ degrees of freedom. After introducing 1 artificial variable indicating for the presence of the 10 high leverage points the residual sum of squares has dropped to $RSS = 18.93$ with $df = 70$ degrees of freedom.

Neural network approach

The analysis was carried out similarly as with the stackloss data. The *err_goal* was set equal to the RSS obtained when using the LSE method with the additional term for the first 10 data vectors, i.e. *err_goal=19.0*. To achieve this goal we need 10 neurons (radial basis functions). 8 from them are centered at data points belonging to the "bad" high leverage points. The decay of the residual sum of squares as a function of the number of neurons in the network is shown below(run assuming *spread = 5.0*):

RSS	34.5	31.3	22.6	22.1	21.7	21.3	20.6	20.2	19.5	18.8
# of neurons	1	2	3	4	5	6	7	8	9	10

One may see, that already one neuron gives a remarkably smaller residual sum of squares (= 34.5) as that obtained by the ordinary LSE method with 3 predictors (RSS = 359.5). The decay of the RSS when adding additional neurons is very slow.

4 Conclusions

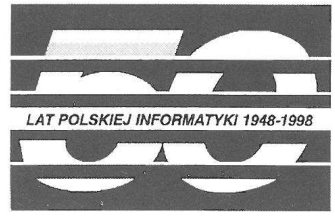
Considering two benchmark data sets we stated that we may obtain the same goodness of approximation (residual sum of squares) both by the statistical and the neural network approach, provided that we know about the outliers.

The statistical approach is much simpler. On the other hand, the radial basis neural network approach is very appealing. None the less: the NN approach is much more uncertain and leaves us always with doubts whether we have attained the proper optimum and there is always the danger of overfitting.

Some complementary graphical displays of the interdependence structure of the investigated data (e.g. running a grand tour) are always advisable.

References

1. A.C. Atkinson (1987): *Plots, Transformations and Regression. An Introduction to Graphical Methods of Diagnostic Regression Analysis.* Clarendon Press. Oxford U.K.
2. A.C. Atkinson (1994): Fast very robust methods for the detection of multiple outliers. *JASA*, 89, 1329–1339.
3. A. Bartkowiak and A. Szustalewicz (1997): Discriminant analysis using neural network algorithms from MatLab – experience with two data sets. *Third Conf. "Neural Networks and Their Applications"*, Kule, Oct. 14–18, 215–220.
4. A. Bartkowiak and A. Szustalewicz (1997): Detecting outliers by a grand tour. *Machine Graphics & Vision* 6, 487–505.
5. C.M. Bishop (1995): *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford U.K.
6. A.S. Hadi and Jeffrey S. Simonoff (1993): Procedures for the identification of multiple outliers in linear models. *JASA*, 88, 1264–1272.
7. T. Masters (1993): *Practical Neural Network Recipes in C++.* Academic Press. Also Polish translation WNT Warszawa 1996.
8. H. Demuth, M. Beale (1997): *Matlab Neural Network Toolbox, User's Guide*, Version 2. The MathWorks Inc., Natick, MA, USA.
9. S. Osowski (1996): *Sieci neuronowe w ujęciu algorytmicznym.* (Neural network, an algorithmic approach, in Polish). 2nd Ed. WNT Warszawa.



Intelligent Information Systems

Proceedings of the Workshop held in Malbork, Poland 15-19 June, 1998

Instytut Podstaw Informatyki
Polskiej Akademii Nauk