ACTIVE SUBSET SELECTION APPROACH TO NONLINEAR MODELING OF ECG DATA

C. Merkwirth

Comp. Biology and Applied Algorithmics Group
Max-Planck-Institut für Informatik
Saarbrücken, Germany

J.D. Wichard and M.J.Ogorzałek

Department of Electrical Engineering
University of Mining and Metallurgy
Kraków, Poland

ABSTRACT

In this article we report results concerning an ensembling approach for regression modeling which could be used for data compression and prediction. We train a sequence of models on small subsets of a large data set in order to achieve small computation time and memory consumption. An active learning approach is used to increase the training subset iteratively to cover the full dynamics of the data set without using all observations for the actual training. The algorithm is part of a software toolbox for ensemble regression modeling that is used to demonstrate the performance of this method on examples of measured ECG time series.

1. INTRODUCTION

Boosting for classification ([1],[2]) has proven to be a powerful method for ensembling even weak classification algorithms into an combined classifier that is able to solve difficult classification tasks. Also several quite different boosting schemes for regression have proposed so far ([3],[4]) which can be seen as variants of ensemble methods. Standard ensemble methods try to decorrelate the outputs of several models by manipulating the distribution of the training set (bootstrapping, cross-validation, reweighting etc.) combined with careful model selection ([5],[6]) and are able to enhance regression performance substantially compared to single regressors.

The main difficulty in order to create an ensemble of regression models that performs significantly better than any of its constituting models is to generate models which have at the same time low generalization error each, but are mutually decorrelated (see [5]). It is usually not sufficient to combine several weak models[1] as it is in the case of classification where one can use classifiers that are slightly better than random guessing, so more sophisticated methods to diversify the generated regression models are needed.

The method presented in this study is not intended to be a boosting scheme for regression problems. We discovered it while trying to train k-nearest-neighbor models on large data sets that have despite of low memory consumption good prediction accuracy. It resembles the active learning approach ([5]) by augmenting the data set with the samples on that the current model performs worst. In that way we enlarge our training set with points that are not in direct neighborhood of the used training examples.

This work was supported by the Research Training Network *COSYC of SENS* No. HPRN-CT-2000-00158 within the 5th Framework Program of the EU. We would like to thank all the involved people, especially the team of Prof. M.J. Ogorzałek, and the people of the Akademia Górniczo–Hutnicza in Kraków, for their support.

[1]Weak is used here in sense of high training error.

Since this method has several advantages compared to a straightforward training, we find it to be promising enough to be applied even when the above mentioned constraints are not of highest importance.

2. DESCRIPTION OF THE METHOD

The task consists of learning a regression function from a subset of the entire data set A. The data is given in the form of input-output pairs $(\vec{x}, y)$, where the input variables $\vec{x} \in \mathbf{R}^d$ are mapped to the scalar output variables $y \in \mathbf{R}$. This approach can be extended to multidimensional output as well. The steps of the algorithm:

1. Choose a random subset S_1 of N samples out of the full data set A. N can be chosen much smaller than the size M of A.
2. Train model m_i on subset S_i.
3. Evaluate model m_i for all samples of the full data set. Calculate the residua (absolute difference between predicted value and actual output value) for all samples of A. Also calculate the mean squared error MSE_i for model m_i with respect to A.
4. Identify the $\triangle N$ samples with the highest residua, avoiding samples that are already part of S_i. Add these samples to subset S_i, thus forming S_{i+1}.
5. Repeat steps 2–4 until either a user specified number of rounds R is reached or the error E_i is sufficiently small.
6. Use some combinatorial algorithm to select a subset E out of the sequence of models $\{m_i\}, i = 1, \ldots, R$ so that the error MSE_E of the output of the ensemble E becomes minimal.[2] The output of the ensemble E is defined as follow:

$$E(\vec{x}) = \frac{1}{\|E\|} \sum_{m_i \in E} m_i(\vec{x})$$

In total, the algorithm uses three parameters that must be specified by the user:

- Size N of the initial subset
- Size $\triangle N$ of the chunk of samples that is added each round
- Number of rounds R

It may happen that a model performs worse than predicting the mean of the outputs on the full data set. In this case, we retrain the model until it performs at least slightly better than predicting the mean.

[2]We couldn't observe a strong sensitivity of the final error towards the kind of combinatorial algorithm used (e.g. GA-type). One could also try to determine the optimal weights to create a weighted ensemble of models.

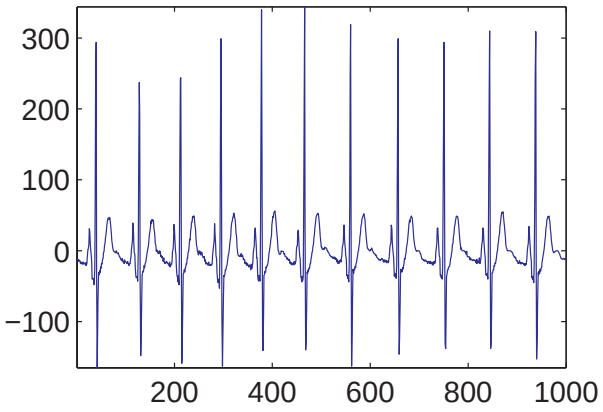

Figure 1: First 1000 samples of the ECG time series used for the numerical experiments. Time-delay-reconstruction with dimension 12 and delay 1 was used to generate the data sets F and G. Both data sets contain 50000 samples each. G is taken from the later part of the ECG time series. Both data sets have no samples in common. Since the MSE of models trained solely on F is lower on G than on F itself, the dynamics of the ECG seems to be stationary.

2.1. Advantages

The described training method has several advantages compared to a straightforward training of models on the full data set:

- Reduced computation time for learning algorithms that have a time complexity higher than $O(N)$.
- Smaller memory consumption for nearest–neighbor and other data driven models that need to store the entire training data set in memory for the lookup procedure. The memory consumption could be further reduced by exploiting the fact that the data sets of consecutive models are nested, thus only needed to be stored once.
- The outputs of independent runs as described above are to some extend decorrelated, which allows to ensemble several ensembles E_j successfully to an ensemble of ensembles EE which exhibits improved prediction accuracy compared to the constituting ensembles (see section 3, table 1).

3. APPLICATION TO REAL WORLD DATA

We used a ECG time series (see figure 1) from the CinC Challenge 2000 data sets: Data for development and evaluation of ECG-based apnea detectors (see [7] for more information about the recording details). The data set a01.dat was cropped to approximately 50000 samples by omitting later measurements. From this time series we generated time-delay-vectors of dimension 12 with delay 1 (see [8]). The modeling task consisted of learning the one-step-ahead prediction for each of the 50000 delay vectors.
Since the final error MSE_E on this full data set F differs from run to run, we repeatedly partitioned F randomly into a training partition A_j of 40000 samples and a test partition A'_j of 10000 samples. A'_j is not used at all during the j-th run of the algorithm as described in section 2, instead it is kept until the end of the training to assess the test error of ensemble E_j.

As parameters for the algorithm we choose $N = 100$, $\triangle N = 10$ and $R = 6$, which results in models m_i that are trained on a maximum of 150 samples out of A_j.
The repeated partitioning and execution of leads to an collection of ensembles $\{E_j\}$, $j = 1, \ldots 10$. We then removed the worst performing ensembles according to their error on the full data set[3] F from that collection and treated the reduced collection as an ensemble of ensembles EE. This method has two advantages:

- If one run diverges or produces an substantially inferior model, it is not used for the final collection of ensembles.
- Due to the stochastic nature of the initial subset and of the training method used, the output of the ensembles generated by the different runs are to some extend decorrelated. This results not only in a substantial reduction of the error $\mathrm{MSE}_{\mathrm{EE}}$ of the collection of ensembles on F, but also in an reduction of the generalization error on unseen data (see [5], [6]).

We used four different underlying model types which range from strictly local to a global models:

1. The model type that was mostly used within the numerical experiments is a variant of the k-nearest-neighbor regressor with adaptive metric. In our case GA-like algorithm adapts the metric coefficients of one of the L_1, L_2 or L_∞ metrics as well as the number k of neighbors. The fitness of an individual is the negative leave-one-out-error on the training data, which can be easily calculated using a fast nearest-neighbor algorithm ([9]). The metric coefficients are adapted by mutation and crossover within a predefined number of generations.
2. As a semi-local model type we decided to use the hybrid PRBFN network based on radial basis function and sigmoidal nodes (see [10]). We found this model type to perform superior on several artificial and real world test problems.
3. As global model type we choose a fully connected neural network with one hidden layer and eight hidden layer neurons. The network is trained using a second-order Levenberg-Marquardt algorithm. The implementation was taken from the NNSYSID2.0 toolbox written by Magnus Nørgaard (see [11]).
4. Another global model type we used was a linear model that was trained using a cross-validation scheme.

3.1. Observations

Several interesting points could be observed:

- During one run of the proposed algorithm, the sequence of errors MSE_i does not simply decrease with increasing i and number of training observations given, instead MSE_i behaves rather discontinuously, partly due to the random nature of some of the training algorithms, partly due to the variations in the distribution of the training data.

[3]The full data F was never available in total during a single training run of the proposed algorithm. However, due to the cross-validation like repartitioning and training scheme, each sample of the full data set is at least member of one of the training sets.

Model type	MSE_E	MSE_{EE}	Gain
k-NN	0.0911	0.0518	43.1%
PRBFN	0.1498	0.0982	34.4%
Neural Net	0.1383	0.0795	42.5%
Linear	0.3851	0.3589	6.8%

Table 1: Results of different model types on the ECG data set F that was used for training of the collection of ensembles. The data set consists of 50000 delay vectors, the task was to predict one-step-ahead. The values in the first two columns of the table are relative Mean Square Errors for the single ensembles MSE_E and for the ensemble of ensembles MSE_{EE}. The last column displays the improvement (Gain) of using an ensemble of ensembles compared to a single ensemble. A value of one for the relative MSE would indicate a model that is as good as predicting the mean of the outputs. The values are averaged over 8 independent runs. All settings are described in section 3.

Model type	MSE_E	MSE_{EE}	Gain
k-NN	0.0712	0.0331	53.5%
PRBFN	0.1157	0.0702	39.3%
Neural Net	0.1001	0.0498	50.3%
Linear	0.3491	0.3248	7.0%

Table 2: Results of different model types on the data set G which was generated from a later part of the ECG time series. G was never used for training the collection of ensembles. All settings are the same as for table 1.

Model type	MSE_E	MSE_{EE}	Gain
k-NN	0.1350	0.0735	45.5%
PRBFN	0.2010	0.1110	44.8%
Neural Net	0.1420	0.0867	39.0%
Linear	0.4168	0.3514	15.7%

Table 3: Results of different model types on data set F. For comparison, we constructed ensembles of models that were trained according to the proposed algorithm, but instead of adding the samples with highest residua, samples were randomly chosen and added to the training subset. All other settings are the same as for table 1.

Model type	MSE on F	MSE on G
k-NN	0.0093	0.0117

Table 4: Results for the k-nearest neighbor model on data set F and G. For comparison, we constructed one ensemble of models that were trained on the full data set F instead of using only a small subset of the full data set.

- We tested the generalization ability of the ensemble on a later part of the same ECG recording. Again approximately 50000 samples were used to create a data set G of 50000 delay vectors. We made sure data set F and G had no common samples. The MSE of both the single ensembles and collection of ensembles on the data set G was throughout lower than on the training data set (see table 2), possibly due to some physiological events occurring during the first 50000 samples of the data set, while the later part shows more regular behavior.

  This indicates that the proposed method isn't simply memorizing the data set F by picking out the most prominent samples, instead it seems to be able to learn and generalize the determinism in this ECG time series which could be used for prediction and compression.

- For comparison, we generated one ensemble of k-nearest-neighbor models that were trained on the full data set F and computed the MSE on training set F and test set G (see table 4). Both MSE are lower compared to the proposed algorithm, which should not surprise since the models were trained on a 33 times larger data set. With this classical training scheme, the error on the test set G is higher than on the training set F, which indicates that slight overfitting occurs.

  The computation time for the training was higher by roughly two magnitudes compared to the training time for the proposed algorithm and the same model type. Unfortunately, it was not possible to extend this comparison to the other model types due to a prohibitively high computation time.

4. DISCUSSION

A main advantage of the proposed algorithm is to construct ensembles of models on very large data sets where the computation time for a standard training can become prohibitively high. The method takes advantage of averaging (ensembling) several small models to an ensemble of models which outperforms each constituting model with respect to training error. Additionally the generated ensembles show no signs of overfitting at all, they even perform better on an unseen test data set. Constructing several models on small subsets of the training data instead of training one model on the full data set is especially useful in conjunction with k-nearest-neighbor regression, which not only yields the lowest error rates (see table 1), but benefits most from the smaller amount of training data actually used due to it's memory based modeling approach.

Since the proposed method is independent of the model type and training algorithm used, it will automatically benefit from switching to a superior model type or training method.

The main drawback for this method is the still lower generalization error that can be achieved by training a model on the full data set. This however is often not feasible for very data sets, so one has to cut down the training set anyway.

Creating ensembles of ensembles with the proposed training scheme yields impressive improvements of the train and test MSE (see tables 1 and 2), but also increases computation time and memory consumption. Nevertheless, the computation time is in both cases still lower than training respective models on the full data set.

5. CONCLUSION

We presented an algorithm that allows the construction of an ensemble of regression models on small subsets of possibly large data sets. The type of the models used for the ensembles is arbitrary, even heterogenous models can be used within the same ensemble. The training scheme constructs ensembles that outperform similar ensembles trained on subsets of same size, but with randomly chosen observations. The scope of the algorithm not restricted to time-series prediction, but covers general regression and function approximation problems.

6. OUTLOOK

The algorithms and methods used for this study will be freely available as a software toolbox for regression modeling and ensembling called *ENTOOL* [12].

7. REFERENCES

[1] Y. Freund, Short introduction to boosting, J. Japan. Soc. for Artif. Intel. 14(5) (1999), 771-780

[2] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer (2001)

[3] R. Avimelech, N. Intrator, Boosting Regression Estimators, Neural Computation, 11 (2) (1999)

[4] H. Drucker, Improving regressors using boosting techniques. In Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann (1997)

[5] A. Krogh, J. Vedelsby, Neural Network Ensembles, Cross Validation and Active Learning, Advances in Neural Information Processing Systems 7, MIT Press (1995)

[6] M. P. Perrone, L. N. Cooper, When networks disagree: Ensemble methods for neural networks, Neural Networks for Speech and Image Processing, Chapman Hall (1993)

[7] Physiobank Database
URL:pcbim2.dsi.unifi.it/physiobank/database/apnea-ecg/index.html

[8] J. Suykens, J. Vandewalle, Nonlinear Modeling - Advanced Black-Box Techniques, Kluwer Academic Publishers (1998)

[9] J. McNames, Innovations in Local Modeling for Time Series Prediction, Ph.D. Thesis, Stanford University (1999)

[10] S. Cohen, N. Intrator, A Hybrid Projection Based and Radial Basis Function Architecture, Lecture Notes in Computer Science, 1857 (2000)

[11] M. Nørgaard, Neural Network Based System Identification Toolbox, Tech. Report. 00-E-891, Department of Automation, Technical University of Denmark (2000)

[12] ENTOOL, A Matlab-toolbox for regression modeling.
URL:www.agnld.uni-potsdam.de/~gamez/Trainingtools.htm