ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 305-310

Local Modeling Optimization for Time Series Prediction

James McNames

Portland State University
Electrical and Computer Engineering
PO Box 751
Portland, Oregon 97207-0741

Abstract. Local models have emerged as one of the leading methods of chaotic time series prediction. However, the accuracy of local models is sensitive to the choice of user-specified parameters, not unlike neural networks and other methods. This paper describes a method of optimizing these parameters so as to minimize the leave-one-out cross-validation error. This approach reduces the burden on the user to pick appropriate values and improves the prediction accuracy.

1. Introduction

Unlike global models, local models postpone the computation required for construction until the input vector is available. The nearest neighbors in the data set are then located, a simple model is constructed using only the neighboring points, and the model is evaluated using the input vector to produce the local model output.

One of the most vexing problems facing users who wish to use a local model is how to choose appropriate values for the model parameters. Since the best parameter values depend on the properties of the data set, there is little to guide users in making this decision.

This paper introduces a method for optimizing the traditionally user-specified parameter values to maximize the model performance. The advantages of this method are that it relieves the user of the burden of specifying critical parameter values, it gives the user control of the computation used for optimization, and it improves the model accuracy as compared to the initial values provided by the user.

1.1. Local versus Global Modeling

Local models have performed very well in comparative studieson time series prediction problems and in most cases have generated more accurate predictions than global methods [5, 4, 3]. However,each of these studies is subject to the problem of *expert bias* in which the researcher may unintentionally bias a

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 305-310

comparative study because they are more skilled at applying the methods that they favor. This problem is largely circumvented by competitions that bring together a large group of researchers to compare their preferred prediction methods on a common set of problems.

Two of these competitions are of special significance. A time series prediction competition was held by the Santa Fe institute in 1991 [9]. Although several types of time series analysis were included in the competition, the prediction of a chaotic time series produced by a laser received the most attention and entries. The winner of the competition used a novel neural network architecture. The second place entry, generated by a local linear model, was nearly as good. A further comparison of these two methods was performed after the competition. On other segments of the time series the local model performed better than the neural network in three out of four trials.

A second competition was held in Leuven, Belgium in 1998 to assess the changes and improvements that had occurred in the field of time series prediction since the Santa Fe competition [8, 7]. Entrants were given a timeseries consisting of 2,000 points and were asked to predict the following 200 points. Both the winning entry and the second place entry were generated by a local model and only local models were able to forecast the first 80 steps accurately.

The scope of these competitions was too narrow to conclusively determine that any specific type of nonlinear modeling is best because they both used only a single prediction sequence from a single time series. However, the strong showing of local models in both competitions strongly supports the claim that local models are among the best techniques for time series prediction.

2. Cross Validation Error (CVE)

Almost all nonlinear models optimize model parameters to minimize some measure of performance. In most cases the measure of performance is an average error, such as mean squared error, taken over the entire data set. This approach often causes the model to be accurate at the points in the data set but to vary substantially at other points, a problem known as *overfitting*.

To solve this problem, users often divide the data set into two parts: a training data set and a test data set. The nonlinear model is then iteratively optimized to minimize the average error on the training data set and the optimization is stopped once the average error on the test set increases. A disadvantage of this approach is that only half of the data is used to directly build the model.

Local models can use a much more accurate technique of estimating the model performance. This technique consists of taking a single point out of the data set, building a nonlinear model using the remaining points in the data set, and using the nonlinear model to estimate the prediction performance for the removed point. The process is repeated for many points in the data set and the average error is calculated. This error is called the *leave-one-out* cross-validation error (CVE).

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 305-310

The computational cost of calculating the average CVE is prohibitive for most global models because it requires the model to be constructed many times. Calculating the average CVE multiple times, as would be necessary to use the CVE in an iterative optimization of model parameters, is even more daunting.

Local models can calculate the CVE almost as efficiently as they can calculate the local model outputs[1]. To estimate the error for an input vector taken from the data set, the $k+2$ nearest neighbors are found and the nearest neighbor, which is identical to the input vector, is discarded. The model error is then evaluated using the vector's $k+1$ neighbors.

The ability to calculate the cross-validation error efficiently is a very important advantage of local models and it plays a vital role in the optimization algorithm described in the next section.

3. Cyclic Coordinate Optimization

Gradient-based optimization algorithms can greatly improve the initial parameter values provided by the user. However, this approach cannot be used to optimize integer-valued parameters, such as the number of neighbors, or parameters for which the gradient cannot be calculated. To optimize these parameters, an algorithm that does not require the gradient must be used. One of the simplest of these algorithms is the cyclic coordinate method. This method optimizes each parameter one at a time, and then repeats until convergence [1, pp. 283–5]. For example, if the parameters to be optimized are stored in a vector $\gamma \in \mathbb{R}^N$, the cyclic coordinate method is as follows.

Cyclic Coordinate Method

1. For $i = 1$ to n,
   1.1 $\gamma_i := \underset{\alpha}{\operatorname{argmin}} \operatorname{CVE}([\gamma_1, \ldots, \gamma_{i-1}, \alpha, \gamma_{i+1}, \ldots, \gamma_n]^T)$.
   1.2 Next i.
2. If not converged, then goto 1.

Since each step in the loop can only decrease the cross-validation error, this method can only improve the model performance; and under very general conditions the algorithm is guaranteed to converge [1, p. 285].

Since the algorithm optimizes each parameter individually, this method is not computationally efficient for models that have a large number of parameters. However, it is an efficient approach for models that have relatively few parameters (less than a dozen), such as local models. Several new parameterizations of local models are described in [6].

3.1. Semi-global Line Search

Each step in the inner loop of the cyclic coordinate method tries to find the value of a single parameter that minimizes the cross-validation error. Since only

[1] An efficient method of calculating the CVE using iterative prediction is described in [2].

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 305-310

one parameter is optimized at each step, this is essentially a one-dimensional minimization problem, also known as the line search problem.

If the parameter to be optimized is an integer, a user-specified range of values can be searched for the best value. For example, the number of neighbors, k, could be optimized by calculating the cross-validation error (CVE) for a range of values, $\{k_{\min}, k_{\min}+1, \ldots, k_{\max}\}$, and retaining the value with the smallest CVE.

If the parameter is a real number, any of a number of line search algorithms could be used to find a local minimum [1]. However, a semi-global line search algorithm is preferable if the CVE contains many shallow local minima, as is the case with most of the local model parameters.

The semi-global line search algorithm used in the results reported here tries increasing and decreasing the parameter value by a range of amplification factors, a set of scalar multipliers. For example, if γ_i is the parameter being minimized and Φ is a set of possible amplification factors, the minimization in step 1.1 would consist of evaluating the CVE with $\alpha = \phi\gamma_i$ for each $\phi \in \Phi$. The parameter γ_i would then be replaced with the value that minimized the CVE.

To ensure that a wide range of parameter values is examined, the amplification factors can be evenly spaced on a logarithmic scale. For example, if the user wished to evaluate the CVE at only eleven points and wanted to try amplification factors ranging from $\frac{1}{10}$ to 10, the amplification factors would be $\{0.100, 0.158, 0.251, 0.398, 0.631, 1.00, 1.58, 2.51, 3.98, 6.31, 10.0\}$.

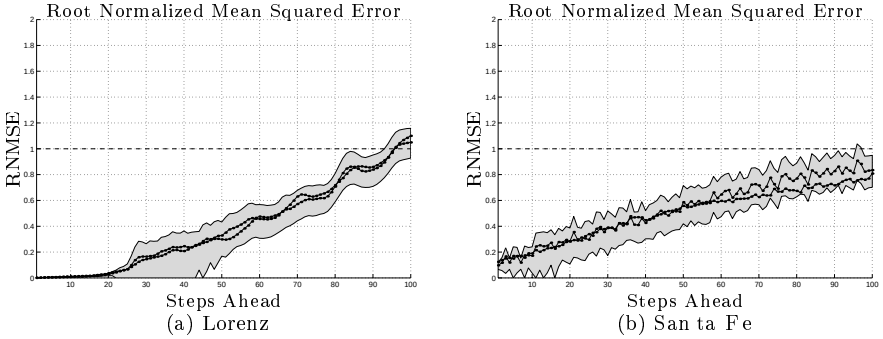
After the cyclic coordinate method converges, the parameter values can be found with greater precision by reducing the range of the amplification factors. For example, after initial convergence, the range could be reduced to $\phi_{\min} = 0.2$ and $\phi_{\max} = 5$.

Cyclic coordinate optimization is better for local models than gradient-based optimization because it can avoid shallow local minima and does not require computation of the gradient. However, the cyclic coordinate method's rate of convergence is much slower than gradient-based optimization methods. If the gradient can be calculated for only some of the model parameters, the rate of convergence can be increased by combining the cyclic coordinate method with a gradient-based algorithm. This type of hybrid approach, called the generalized cyclic coordinate method, converges substantially faster than the cyclic method and is described in detail in [6].

4. Overfitting

An accurate method of estimating the model accuracy is an essential component of model optimization algorithms such as the cyclic coordinate method described in the previous section. Although the leave-one-out cross-validation error (CVE) is intuitively more accurate than a partitioning of the data into a training set and test set, it is not obvious how much overfitting occurs and how biased the CVE estimate of model performance is.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 305-310



(a) Lorenz (b) Santa Fe

Figure 1: Prediction horizons for (a) the Lorenz time series and (b) the Santa Fe competition time series. The horizons show the square root of the mean squared error divided by the sample variance of the time series (normalized) versus the number of steps predicted ahead. The two lines show the mean CVE and the test error. The gray region shows the estimated standard deviation of the mean CVE.

To investigate this question a local linear model[2] was optimized using two chaotic time series benchmarks, the Lorenz time series and the Santa Fe competition time series[3]. In both cases the CVE was estimated using 1,000 equally spaced points taken from the first 3,000 points in the time series and the average test error was calculated from 4,000 evenly spaced points taken after the segment used to build the model.

Figure 1 shows the prediction horizons. In each plot the gray region shows three standard deviations of the average CVE. Since the test error is well within this region, these plots[4] give empirical support that the CVE is not significantly biased by the model optimization and overfitting does not occur.

5. Conclusion

This paper introduced a new method of local model optimization based on a generalization of the cyclic coordinate method. This method is especially well suited to local models because the number of parameters is typically small (less than a dozen) and it does not require the error gradient. This method has the additional benefit of converging to better local minima than gradient-based optimization algorithms because each line search is performed semi-globally.

Although optimization of model parameters is not new, there has been little work to apply these methods to local models. This approach replaces the burden on non-expert users of choosing sensitive model parameters with

[2]The details of the local model parameterization and optimization are given in [6], which is available online at *http://www.ee.pdx.edu/~mcnames*.

[3]Both of these are available online at *http://www.ee.pdx.edu/~mcnames/DataSets*.

[4]These results are typical of those observed for many chaotic time series and local models.

the responsibility of choosing a range of parameter values. This is preferable because the user will typically have a much better idea of the range, such as the number of neighbors, than the best value. This also lets the user make the tradeoff of model accuracy for the amount of computation used to optimize the model.

An accurate estimate of the model performance is an essential component of model optimization. Local models have a distinct advantage over global models in this respect because they can efficiently calculate the leave-one-out cross-validation error (CVE). Empirical evidence was given to demonstrate that this measure of model accuracy is not significantly biased by the model optimization and is not susceptible to overfitting.

References

[1] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., second edition, 1993.

[2] G. Bontempi, M. Birattari, and H. Bersini. Local learning for iterated time-series prediction. In *Machine Learning: Proceedings of the Sixteenth International Conference*, pages 32–38, San Francisco, CA, 1999.

[3] Martin Casdagli, Deirdre Des Jardins, Stephen Eubank, J. Doyne Farmer, John Gibson, and James Theiler. Nonlinear modeling of chaotic time series: Theory and applications. In Jong Hyun Kim and John Stringer, editors, *Applied Chaos*, pages 335–380. John Wiley & Sons, Inc., 1992.

[4] J. D. Farmer and John J. Sidorowich. Exploiting chaos to predict the future and reduce noise. In Yee Chung Lee, editor, *Evolution, Learning and Cognition*, pages 277–330. World Scientific, 1988.

[5] J. Doyne Farmer and John J. Sidorowich. Predicting chaotic time series. *Physical Review Letters*, 59(8):845–848, August 1987.

[6] James McNames. *Innovations in Local Modeling for Time Series Prediction*. PhD thesis, Stanford University, 1999.

[7] J.A.K. Suykens and J. Vandewalle, editors. *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modelings*, Katholieke Universiteit Leuven, Belgium, July 1998.

[8] Johan A. K. Suykens and Joos Vandewalle. *Nonlinear Modeling Advanced Black-Box Techniques*. Kluwer Academic Publishers, 1998.

[9] Andreas S. Weigend and Neil A. Gershenfeld. *Time Series Prediction*. Addison-Wesley Publishing Company, 1994.