Backtracking Orthogonal Least Squares Algorithm for Model Selection

E.S.Chng[1], B.Mulgrew[1] and S.Chen[2]

Abstract The orthogonal least squares (OLS) algorithm [1] is an efficient implementation of the forward-selection [2] method for subset model selection. The ability to find good subset parameters with only a linearly increasing computational requirement makes this method attractive for practical implementations. This technique, however, has been criticised [2] as it does not guarantee to find an optimum solution.

In this paper, we examine why forward-selection technique can fail to find optimum subset models and present a modification scheme to improve the selection process.

1. Introduction

In signal prediction of time series with nonlinear dynamics, improved performance can normally be achieved by using a nonlinear predictor rather than a linear one. The current popular implementations of nonlinear predictors include the use of radial basis functions [1,3] expansions and Volterra expansions [4]. These implementations create nonlinear predictors that have the *linear in parameters* structure (figure 1). The recent introduction of fuzzy basis functions by Wang and Mendel [5] also exhibits this linear in parameters structure. Such a structure allows the predictor to be represented by a linear regression model, and thus permits the use of the forward-selection method for subset model selection.

Initial models created from these nonlinear expansion techniques are usually very large. Such large models can normally be reduced to a smaller model without incurring much performance loss if we can select the parsimonious model carefully.

This paper is organised as follows: Section 2 reviews the OLS algorithm for subset model selection. Some results are presented in section 3 showing performance of subset models found using the OLS algorithm. Section 4 examines the problem of sub-optimal selection in applying the OLS algorithm and introduces a simple backtracking approach to improve selection process. The results of using the backtracking OLS are presented along with some concluding remarks in section 5.

2. OLS Algorithm

Let us represent these nonlinear predictors that have the linear in parameters structure as a linear regression model:

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{e} \tag{1}$$

where $\mathbf{y}$ is the desired signal vector, $\mathbf{X}$ is the input matrix of size $N \times K$, $\mathbf{h}$ is the parameter vector of the model and $\mathbf{e}$ the error vector of approximating $\mathbf{y}$ by $\mathbf{X}\mathbf{h}$. The column vectors $\mathbf{y}$ and $\mathbf{e}$ contain N elements, i.e., there are N data samples and N values of error.

The original $\mathbf{X}$ matrix have K columns. To create a parsimonious model which has R parameters, we are trying to pick R columns from the input matrix $\mathbf{X}$ to form a subset input matrix $\mathbf{X}s$. The OLS selects columns from the input matrix sequentially. At each selection, all the unused columns are studied to determine how each column will contribute to fit the desired vector $\mathbf{y}$ with the current subset $\mathbf{X}s$. The column which forms the best combination with $\mathbf{X}s$ to model $\mathbf{y}$ will be picked to form the new $\mathbf{X}s$. The above procedure is then repeated until the number of columns in $\mathbf{X}s$ equals to R.

[1]Department of Electrical Engineering, The University of Edinburgh, UK
[2] Department of Electrical and Electronics Engineering, The University of Portsmouth, UK

To understand the OLS algorithm, a brief review of orthogonal projection is presented. Given a pair of vectors $\mathbf{u}$ and $\mathbf{v}$, the projection of vector $\mathbf{u}$ onto $\mathbf{v}$, $Proj_{\mathbf{v}}\mathbf{u}$, can be thought of as the shadow formed by vector $\mathbf{u}$ onto $\mathbf{v}$ when an imaginary light is directed to $\mathbf{u}$ (see figure 2). Orthogonal projection is useful because it possesses the property that the projected vector $Proj_{\mathbf{v}}\mathbf{u}$ is the best approximate of $\mathbf{u}$ using $\mathbf{v}$. The followings are the definition of projection and the equation to find the orthogonal residual.

$$Proj_{\mathbf{v}}\mathbf{u} = \mathbf{v}\left(\frac{\mathbf{u}^T\mathbf{v}}{||\mathbf{v}||^2}\right) \quad (2)$$

$$\mathbf{w}_{\mathbf{v}} = \mathbf{u} - Proj_{\mathbf{v}}\mathbf{u} \quad (3)$$

where $||\mathbf{v}||$ denotes the Euclidian norm of vector $\mathbf{v}$. $\mathbf{w}_{\mathbf{v}}$ is the residual when $\mathbf{v}$ is used to approximate $\mathbf{u}$. This vector is orthogonal to $\mathbf{v}$.

The orthogonal projection of a vector is used in the OLS algorithm to determine the correlation of each column $[\mathbf{c}_0, \mathbf{c}_1, \cdots, \mathbf{c}_{K-1}]$ of the input matrix $\mathbf{X}$ to the output vector $\mathbf{y}$. The column selected will be the one that has the smallest residual in the Euclidian norm sense when the vector $\mathbf{y}$ is projected to it.

OLS Algorithm

1. Initialise
   - no_col_found = 0;
     Col_Found$[1 \ldots R] = 0$;
   - copy the vector $\mathbf{y}$ to $\mathbf{y}_{original}$.
2. Calculate the residual vector
   - For columns $i = 0$ to $(K-1)$ of the input matrix $\mathbf{X}$ that are unused,
     - Project $\mathbf{y}$ onto each column $\mathbf{c}_i$
     - $\mathbf{residual}_i = \mathbf{y} - Proj_{\mathbf{c}_i}\mathbf{y}$.
3. Select a column
   - The column selected $\mathbf{c}_j$ is the column that has the smallest Euclidian norm for the residual vector, $\mathbf{residual}_j$.
   - no_col_found = no_col_found + 1
     Col_Found[no_col_found] $= j$.
     This array Col_Found stores the indexes of the selected columns of $\mathbf{X}$ to form $\mathbf{X}s$.
     Mark the selected column as used.
   - Update $\mathbf{y}$ as $\mathbf{residual}_j$, i.e.,
     $\mathbf{y} = \mathbf{y} - Proj_{\mathbf{c}_j}\mathbf{y}$.
     This new $\mathbf{y}$ can be thought of as the unexplained desired output of the present model.
4. Update the $\mathbf{X}$ matrix
   For columns $i = 0$ to $(K-1)$ of $\mathbf{X}$ that are unselected, update each column $\mathbf{c}_i$ as follows:
   $$\mathbf{c}_i = \mathbf{c}_i - Proj_{\mathbf{c}_j}\mathbf{c}_i.$$
   This is to remove the contribution already *given* by the selected column $\mathbf{c}_j$.
5. Repeat steps 2-4 until the number of columns found to form $\mathbf{X}s$ is equal to R, the desired number of coefficients for the subset model.

Remark: The use of orthogonal projection to select the columns (steps 2,3) is the key to the OLS algorithm. In fact, this algorithm is very similar to the Gram-Schmidt (GS) orthogonalisation scheme [6]. The only difference is that instead of orthogonalising the $\mathbf{X}$ matrix from the first column to the last in a sequential order as in the GS scheme, the columns are now orthogonalised in the order of column's contribution to the approximation of the desired output vector $\mathbf{y}$.

3. Simulations

A Volterra predictor to perform single-step prediction on two different chaotic time series, the Duffing's and the Mackey-Glass time series, is examined. The Volterra

predictor was created using a degree 3 and embedding vector length 6[1] expansion which generated a model with 84 parameters.

The chaotic time series were generated using the following equations :

- Duffing's time series

$$\frac{d^2s(t)}{dt^2} + \alpha\frac{ds(t)}{dt} - s(t) + s^3(t) = \beta cos(t) \tag{4}$$

where $\alpha = 0.25$, $\beta = 0.3$, $s(0) = 0$, $\frac{ds(0)}{dt} = 0$ and step-size = 0.2 sec. Gaussian noise were added to this series to create a signal to noise ratio (SNR) of -50dB.

- Mackey-Glass time series

$$\frac{ds(t)}{dt} = -bs(t) + \frac{\alpha s(t-\tau)}{1 + s^{10}(t-\tau)} \tag{5}$$

where $\tau = -21$, $\alpha = 0.2$, $b = 0.1$, initial conditions $s(t-\tau) = 0.5$ for $0 \leq t \leq \tau$, step-size = 2 sec. Gaussian noise were added to create a SNR of -50dB.

The following measure of prediction quality, the normalised mean square error *(NMSE)*, was used:

$$NMSE = 10log_{10}\left(\frac{\sum_{i=0}^{N-1} e_i^2}{\sum_{i=0}^{N-1} y_i^2}\right) \tag{6}$$

where N is the number of signal samples. The element e_i is the error between the desired signal y_i and its estimation $\hat{y}_i$. The approximation of the vector $\mathbf{y}$, i.e. $\hat{\mathbf{y}}$, was formed using $\hat{\mathbf{y}} = (\mathbf{X}s)(\mathbf{h}s)$. The parameter vector $\mathbf{h}s$ was calculated using the least squares solution [6].

From equation (6), we can see that in the case when we have perfect prediction, $e_i = 0$ for $0 \leq i \leq N-1$, the *NMSE* will be $-\infty$ dB. When there is no prediction, $\hat{y}_i = 0, e_i = y_i$ for $0 \leq i \leq N-1$, the *NMSE* will be 0 dB.

In the simulation, the OLS algorithm was used to find subset models from the full 84 parameter model. Using the subset models found, single step prediction performance for the two time series was calculated. The results in figure 3 show that for the Duffing's test signal, a subset model of 10 parameters was sufficient to achieve a performance level within 1 dB of the full model. For the Mackey-Glass case (figure 4), a subset model of 40 parameters was required before the performance reaches within 1 dB of the full model.

In both cases, it was shown that a performance very close to the full model's performance can be achieved by using a parsimonious model selected by the OLS algorithm.

4. Backtrack Algorithm

Before introducing the backtrack algorithm, an example is presented to show why the OLS algorithm may fail to select an optimal solution. Let the input matrix $\mathbf{X}$ and the desired output vector $\mathbf{y}$ have the following values:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0.1 \\ 0 & 0 & 0.1 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \end{bmatrix}$$

The columns of $\mathbf{X}$ are numbered as $\mathbf{c}_0, \mathbf{c}_1$ and $\mathbf{c}_2$ respectively. When applying the OLS algorithm to the above problem, the order of columns selected to form the subset model $\mathbf{X}s$ was $\mathbf{c}_2, \mathbf{c}_0$ and $\mathbf{c}_1$. It is true that for the case of a one column $\mathbf{X}s$, using $\mathbf{c}_2$ will give the best

[1] Length 6 was chosen to satisfy Takens' theorem [7] for the length of the embedding vector.

approximation of $\mathbf{y}$. However, when a two parameter subset model is desired, the best choice of $\mathbf{X}s$ will be to use columns $\mathbf{c}_0$ and $\mathbf{c}_1$. Using any other combinations to form $\mathbf{X}s$ will yield poorer results. As the present OLS algorithm does not backtrack to hunt for a better $\mathbf{X}s$ subset, this example shows that a sub-optimal subset of the two parameter $\mathbf{X}s$ was formed using $\mathbf{c}_2$ and $\mathbf{c}_0$.

From this example, it is clear that subset selection can be improved if we can change the columns selected in the previous stages, that is, a larger subset model does not necessarily contain all the columns of the smaller subset selected previously. This is the basic idea of the backtracking approach.

To use the backtracking method, the OLS algorithm is first applied to select the initial subset model of R terms. As the subset model size is increased from 1 to R, the *NMSE* will decrease monotonously. The drop of *NMSE* at each parameter's inclusion is calculated. By studying how each parameter contributes to reducing the approximation error, we can determine if the order of parameter selection should be changed. The idea of the backtrack procedure is to introduce parameters that have provide better performance-gain before those that provide less significant gains. The details of the algorithm are given in the following paragraph:

Backtrack OLS Algorithm

1. Use the OLS approach to find the initial $\mathbf{X}s_R$ set. The subscript R of $\mathbf{X}s$ imply that the matrix $\mathbf{X}s$ has R columns.

2. For $i = 1$ to R
   determine the *NMSE* values of using $\mathbf{X}s_i$ to approximate $\mathbf{y}$, where $\mathbf{X}s_i$ contains i columns selected from $\mathbf{X}$ by OLS algorithm. Store these values into the bench-mark array BM_NMSE, i.e., BM_NMSE[i] = *NMSE* attained using the subset model containing i columns. This array serves as a bench-mark for comparing against the performance of subset models selected using the backtracking approach. The array drop_NMSE is defined as:
   drop_NMSE[i] =
   BM_NMSE[i] - BM_NMSE[i-1]

3. For $i = 2$ to R
   - For $j = i + 1$ to R
   if (drop_NMSE[j] > drop_NMSE[i])

   (a) Form a $\mathbf{X}_{tmp}$ matrix using the first $i - 2$ columns of $\mathbf{X}s_R$ and column j of $\mathbf{X}s_R$. This $\mathbf{X}_{tmp}$ matrix will have $i - 1$ columns.
   If $i = 2$, $\mathbf{X}_{tmp}$ matrix will be a single column matrix with the column from column j of $\mathbf{X}s_R$.

   (b) Re-apply the OLS method to find $\mathbf{X}s'$, where $\mathbf{X}s'$ is the selection of R columns from $\mathbf{X}$ based on the initial set specified in $\mathbf{X}_{tmp}$.

   (c) Determine the *NMSE* values using $\mathbf{X}s'$ to model $\mathbf{y}$ and store it into the array tmp_NMSE. Each element in tmp_NMSE will be compared to its corresponding element in BM_NMSE. Any performance improvement in tmp_NMSE implies that the new subset found using the backtrack procedure is better. The indexes of the columns used to form this subset are stored. We then update the BM_NMSE array with the new improved values.

   (d) Skip the rest of j, i.e. set $j = R+1$.

Remark: Step 3a is the key idea to the backtrack algorithm. A parameter that has provided a better contribution to model vector $\mathbf{y}$ albeit in a larger subset model $\mathbf{X}s$ replaces the original selection. Although this exchange is not optimal in forming the $(i-1)$-terms subset model $\mathbf{X}s_{(i-1)}$, to the i-terms subset model $\mathbf{X}s_i$, it is possible for better combinations of $\mathbf{X}s'$ to be formed.

5. Conclusions The backtracking algorithm was applied to find subset models for predicting the two chaotic time series. The performance of the subset models found using the backtracking approach is shown in figure 3 and 4. For both sets of test data, the backtrack OLS algorithm achieves better performance than the standard OLS algorithm. However, the performance advantage is not significant and when viewed in combination with the additional computational complexity, it would be difficult to recommend it as a significant alternative to the standard OLS algorithm.

One possible way of reducing computation time for the backtrack OLS is to first check the subset model $\mathbf{X}s'$ to be examined. We have observed that the backtracking OLS tends to re-examine many similar models during the selection process and thus computational requirement can be reduced by ignoring those subsets already tested.

Acknowledgements We would like to thank Gary Ushaw and David Hughes for providing the data used in the simulations. E.S. Chng would like to thank his parents for their understanding and support for the undertaking of the PhD studies.

References

[1] S.CHEN, C.F.COWAN, and P.M.GRANT, "Orthogonal least squares learning algorithm for radial basis function networks", *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, 1991.

[2] R.R.HOCKING, "The analysis and selection of variables in linear regression", *Biometrics*, vol. 32, pp. 1–49, 1976.

[3] M.J.D.POWELL, "Radial basis functions for multivariable interpolation: a review", *Algorithms for Approximation*, pp. 143–167, J.C.MASON and M.G.COX (Eds), Oxford, 1987.

[4] P.ALPER, "A consideration of the discrete Volterra series", *IEEE Trans. AC*, vol. AC-10, pp. 322–327, 1965.

[5] L.WANG and J.M.MENDEL, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning", *IEEE Trans. Neural Networks*, vol. 3, pp. 807–814, 1992.

[6] G.H.GOLUB and C.F.VAN LOAN, *Matrix Computations (2nd Edition)*, The John Hopkins University Press, Baltimore, MD, 1989.

[7] F.TAKENS, "Detecting strange attractors in turbulence", *Lecture Notes in Mathematics*, pp. 366–381, Springer-Verlag, New York, 1980.

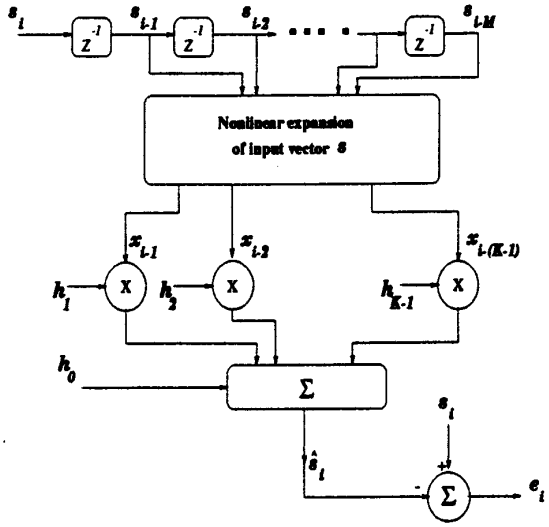

Figure 1: Nonlinear Predictor of Order K

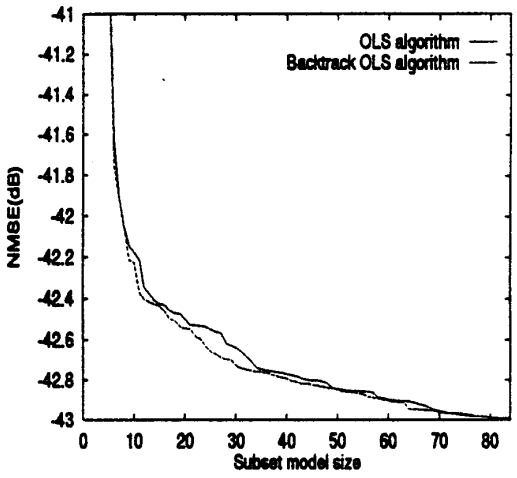

Figure 3: OLS and Backtrack OLS on Duffing's series

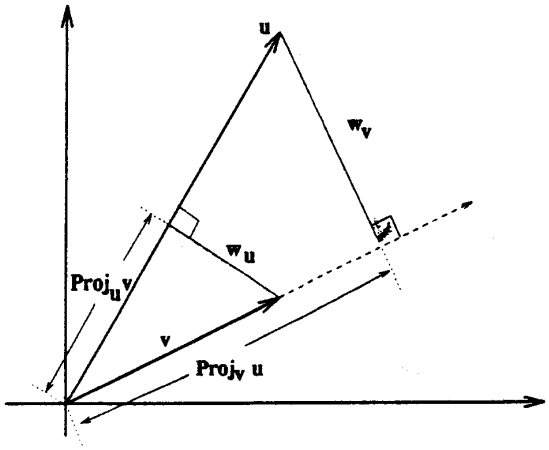

Figure 2: Orthogonal projections

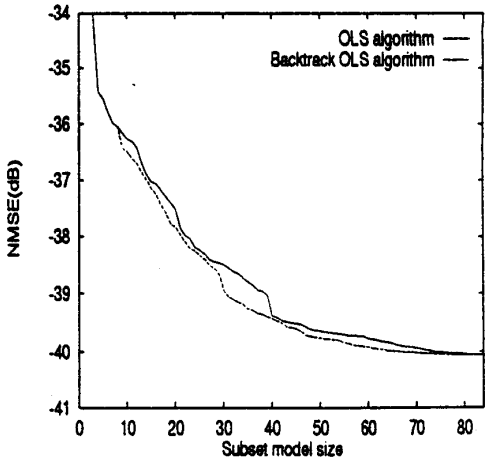

Figure 4: OLS and Backtrack OLS on Mackey-Glass series