Locally Regularised Orthogonal Least Squares Algorithm for the Construction of Sparse Kernel Regression Models

Sheng Chen
Department of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ
United Kingdom
Email: sqc@ecs.soton.ac.uk

Abstract

The paper proposes to combine an orthogonal least squares (OLS) model selection with local regularisation for efficient sparse kernel data modelling. By assigning each orthogonal weight in the regression model with an individual regularisation parameter, the ability for the OLS model selection to produce a very parsimonious model with excellent generalisation performance is greatly enhanced.

I. Introduction

A basic principle in practical data modelling is the parsimonious principle. The OLS algorithm [1],[2] is an efficient learning procedure for constructing sparse regression models. A key feature of the OLS algorithm is its ability to selects significant regressors. The parsimonious principle alone however is not entirely immune to overfitting. If data are highly noisy, small models constructed may still fit into noise. A useful technique for overcoming overfitting is regularisation [3],[4]. By combining the parsimonious principle with a regularisation method, a regularised OLS algorithm has been developed [5]. As this algorithm employs a same regularisation parameter for every weights in the model, it will be referred to as the uniformly regularised OLS (UROLS) algorithm. From the Bayesian viewpoint, a regularisation parameter is equivalent to the ratio of the related hyperparameter to the noise parameter [6].

An effective Bayesian learning method is the evidence procedure which iteratively optimises model parameters and associated hyperparameters [6]. For kernel regression models, this leads to the relevance vector machine (RVM) method [7]. A key feature of the RVM is the introduction of an individual hyperparameter for each weight, which is responsible for the sparsity properties of the RVM method. This paper proposes a regularised OLS algorithm by combining the OLS selection and the idea of associating each model weight with an individual regularisation parameter. The algorithm will be called the locally regularised OLS (LROLS). As regularisation is introduced in the orthogonal weight space, the Hessian matrix needed for updating regularisation parameters is diagonal, giving considerably numerical advantages. The algorithm retains the ability to select significant regressors, and local regularisation further enforces sparsity. The end result is a simple and efficient algorithm for constructing sparse models that generalise well.

II. The general kernel regression model

Consider the kernel regression model of the form:

$$y(k) = \hat{y}(k) + e(k) = \sum_{i=1}^{n_M} \theta_i \phi_i(k) + e(k), \quad 1 \le k \le N, \tag{1}$$

where $y(k)$ is the target, $e(k)$ is the error between $y(k)$ and the model output $\hat{y}(k)$, θ_i are the model weights, $\phi_i(k)$ are the regressors, n_M is the total number of candidate regressors, and N the number of training samples. By defining

$$\mathbf{y} = [y(1) \cdots y(N)]^T, \tag{2}$$

$$\mathbf{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{n_M}], \tag{3}$$

$$\boldsymbol{\phi}_i = [\phi_i(1) \cdots \phi_i(N)]^T, \tag{4}$$

$$\boldsymbol{\theta} = [\theta_1 \cdots \theta_{n_M}]^T, \tag{5}$$

$$\mathbf{e} = [e(1) \cdots e(N)]^T, \tag{6}$$

the regression model (1) can be written in the matrix form

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\theta} + \mathbf{e}. \tag{7}$$

Let an orthogonal decomposition of the matrix $\mathbf{\Phi}$ be

$$\mathbf{\Phi} = \mathbf{W}\mathbf{A} \tag{8}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,n_M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n_M-1,n_M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{9}$$

and

$$\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_{n_M}] \tag{10}$$

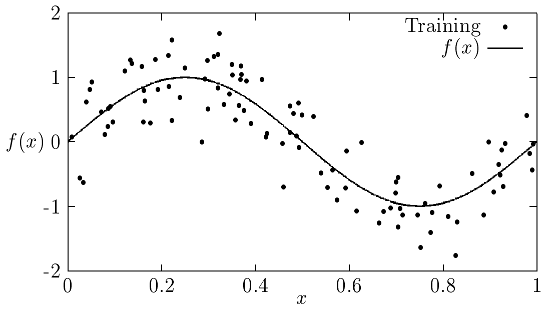


Fig. 1. Noisy training data y (dots) and underlying function $f(x)$ (curve) for the simple scalar function modelling problem.

with columns satisfying $\mathbf{w}_i^T\mathbf{w}_j = 0$, if $i \neq j$. The regression model (7) can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\mathbf{g} + \mathbf{e} \tag{11}$$

where the orthogonal weight vector $\mathbf{g} = [g_1 \cdots g_{n_M}]^T$ satisfy the triangular system

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{g}. \tag{12}$$

Knowing $\mathbf{A}$ and $\mathbf{g}$, $\boldsymbol{\theta}$ can readily be solved from (12).

III. The locally regularised OLS algorithm

According to the Bayesian learning principle [6], the following error criterion can be adopted:

$$J_B(\mathbf{g}, \mathbf{h}, \beta) = \beta\mathbf{e}^T\mathbf{e} + \sum_{i=1}^{n_M} h_i g_i^2 = \beta\mathbf{e}^T\mathbf{e} + \mathbf{g}^T\mathbf{H}\mathbf{g} \tag{13}$$

where β is the noise parameter, $\mathbf{h} = [h_1 \cdots h_{n_M}]^T$ is the hyperparameter vector, and $\mathbf{H} = \text{diag}\{h_1, \cdots, h_{n_M}\}$. Let $\lambda_i = h_i/\beta$. An equivalent error criterion to (13) is:

$$J_R(\mathbf{g}, \boldsymbol{\lambda}) = \mathbf{e}^T\mathbf{e} + \sum_{i=1}^{n_M} \lambda_i g_i^2 = \mathbf{e}^T\mathbf{e} + \mathbf{g}^T\boldsymbol{\Lambda}\mathbf{g} \tag{14}$$

where $\boldsymbol{\lambda} = [\lambda_1 \cdots \lambda_{n_M}]^T$ is the regularisation parameter vector, and $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \cdots, \lambda_{n_M}\}$. It can readily be shown that the criterion (14) can be expressed as

$$\mathbf{e}^T\mathbf{e} + \mathbf{g}^T\boldsymbol{\Lambda}\mathbf{g} = \mathbf{y}^T\mathbf{y} - \sum_{i=1}^{n_M} \left(\mathbf{w}_i^T\mathbf{w}_i + \lambda_i\right) g_i^2. \tag{15}$$

Normalising (15) by $\mathbf{y}^T\mathbf{y}$ yields

$$\left(\mathbf{e}^T\mathbf{e} + \mathbf{g}^T\boldsymbol{\Lambda}\mathbf{g}\right)/\mathbf{y}^T\mathbf{y} = 1 - \sum_{i=1}^{n_M} \left(\mathbf{w}_i^T\mathbf{w}_i + \lambda_i\right) g_i^2/\mathbf{y}^T\mathbf{y}. \tag{16}$$

As in the case of the OLS algorithm [1], the regularised error reduction ratio due to $\mathbf{w}_i$ is defined by

$$[\text{rerr}]_i = \left(\mathbf{w}_i^T\mathbf{w}_i + \lambda_i\right) g_i^2/\mathbf{y}^T\mathbf{y}. \tag{17}$$

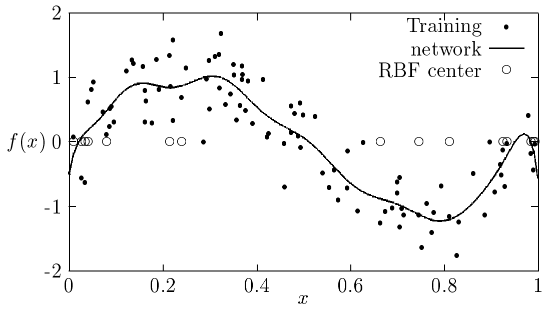


Fig. 2. Model mapping (curve) produced by the OLS algorithm for the simple scalar function modelling problem. Dots indicate noisy training data y and circles the RBF centers.

Based on this ratio, significant regressors is selected in a manner exactly as in the case of the OLS algorithm [1]. The selection is terminated at the n_s-th stage when

$$1 - \sum_{l=1}^{n_s} [\text{rerr}]_l < \xi \tag{18}$$

is satisfied, where $0 < \xi < 1$ is a chosen tolerance. This produces a sparse model containing n_s ($\ll n_M$) significant regressors. Notice that, in the selection procedure, if $\mathbf{w}_i^T\mathbf{w}_i$ is too small, this term will not be selected. Thus, any ill-conditioning can automatically be avoided.

The Bayesian evidence procedure [6] can readily be used to optimise the regularisation parameters. Applying this evidence procedure leads to the updating formulas:

$$\lambda_i^{\text{new}} = \frac{\gamma_i^{\text{old}}}{N - \gamma^{\text{old}}} \frac{\mathbf{e}^T\mathbf{e}}{g_i^2}, \ \ 1 \le i \le n_M, \tag{19}$$

where

$$\gamma_i = \frac{\mathbf{w}_i^T\mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T\mathbf{w}_i} \tag{20}$$

and

$$\gamma = \sum_{i=1}^{n_M} \gamma_i. \tag{21}$$

Usually a few iterations are sufficient to find an optimal $\boldsymbol{\lambda}$.

IV. Two examples

Example 1. Consider modelling the scalar function

$$f(x) = \sin(2\pi x), \ \ 0 \le x \le 1, \tag{22}$$

by a Gaussian radial basis function (RBF) model. The Gaussian kernel function used had a variance of 0.04. One hundred training data were generated from $y = f(x) + \epsilon$, where the input x was uniformly distributed in $(0,\ 1)$ and the noise ϵ was Gaussian with zero mean and variance 0.16. The noisy training points y and the underlying function $f(x)$ are plotted in Fig. 1. As each training data x was considered as a candidate RBF center, there were $n_M = 100$ regressors in the model (1).

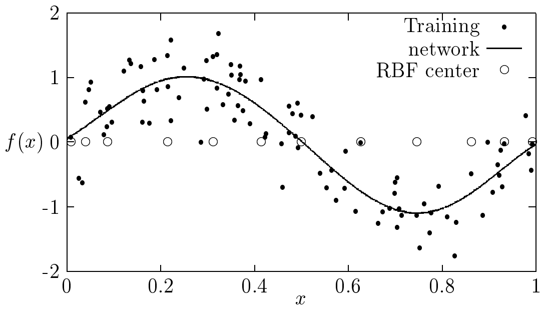


Fig. 3. Model mapping (curve) produced by the UROLS algorithm for the simple scalar function modelling problem. Dots indicate noisy training data y and circles the RBF centers.

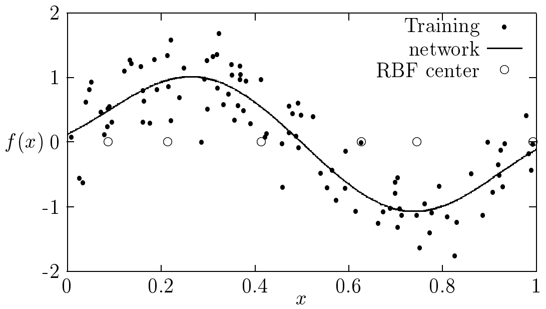


Fig. 4. Model mapping (curve) produced by the LROLS algorithm for the simple scalar function modelling problem. Dots indicate noisy training data y and circles the RBF centers.

The training data were very noisy, and this learning problem was very ill-conditioned.

The selection process of the OLS algorithm is listed in Table I. Notice that the normalised MSE continuously decreased as more terms were added. The procedure stopped at the 16-th stage, when it detected that adding one more term would cause the problem to be singular. This produced a 15-term model. The model weights had very large value, a typical sign of overfitting. The MSE over the training set was smaller than the noise variance, indicating that the model was fitted into the noise. Overfitting can also be seen clearly by the model map given in Fig. 2.

The UROLS selection procedure, after the single λ had converged, is listed in Table II. The selection stopped at the 14-th stage, as there was no more candidate which would not cause an ill-conditioning or singular problem. The modelling accuracy $1-\sum[\text{rerr}]_l$ remained unchanged after the 11-th stage. The weight of the 13-th regressor was effectively zero, indicating a 12-term model. The model map produced by this 12-term model is depicted in Fig. 3, where it is clearly seen that overfitting did not occur.

The LROLS selection procedure, after $\boldsymbol{\lambda}$ had converged, is listed in Table III. The selection stopped at the 14-th stage, as there was no more candidate which would not cause an ill-conditioning or singular problem. The modelling accuracy $1-\sum[\text{rerr}]_l$ however remained unchanged after the 6-th stage. The regularisation parameters related with the 7-th to 13-th terms were all very large, and the associated model weights were effectively zero. This clearly indicated a 6-term model. The model map produced by this 6-term model is depicted in Fig. 4, where it can be seen that the generalisation performance of this 6-term model was similar to that of the 12-term model produced by the UROLS algorithm.

Example 2. This example constructed a model representing the relationship between the fuel rack position (input) and the engine speed (output) for a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed. It is known that at low engine speed, the relationship between the input and output is nonlinear [8]. Detailed system description and experimental setup can be found in [8]. The data set contained 410 samples. The first 210 data points were used in modelling and the last 200 points in model validation. A RBF model of the form:

$$\hat{y}(k) = f_{RBF}(y(k-1), u(k-1), u(k-2)) \qquad (23)$$

TABLE I
OLS SELECTION FOR THE SIMPLE SCALAR FUNCTION MODELLING.

stage l	$1-\sum[\text{err}]_l$	weight θ_l
1	0.6461718264	2.60935e+06
2	0.2840641827	-2.28370e+06
3	0.2416057207	-1.29831e+08
4	0.2260673781	-2.21722e+09
5	0.2189319619	3.63027e+08
6	0.2179112365	1.66438e+09
7	0.2169210404	-3.19282e+09
8	0.2156145110	1.70011e+09
9	0.2135190658	4.06932e+09
10	0.2113153903	-1.94658e+09
11	0.2108713704	-2.72236e+08
12	0.2095033180	-4.28658e+07
13	0.2093349973	5.60372e+06
14	0.2091282455	-1.59224e+06
15	0.2068241235	3.83400e+05
stop due to no term selected at 16 stage		
MSE over noisy training set: 0.147430		

TABLE II
UROLS SELECTION FOR THE SIMPLE SCALAR FUNCTION MODELLING AFTER λ HAS CONVERGED.

stage l	$1-\sum[\text{rerr}]_l$	weight θ_l
1	0.6490143575	1.62388e+00
2	0.2908595802	-2.28935e+00
3	0.2508542689	-8.48791e-01
4	0.2361130705	8.22056e-01
5	0.2322792890	1.03731e+00
6	0.2312755537	-3.73154e-01
7	0.2312749762	3.01529e-02
8	0.2312737869	-1.51268e-02
9	0.2312736479	-5.40054e-03
10	0.2312736475	3.76698e-04
11	0.2312736474	9.55162e-05
12	0.2312736474	-1.27653e-05
13	0.2312736474	-2.25256e-07
stop due to no term selected at 14 stage		
MSE over noisy training set: 0.156678		
regularization parameter λ: 3.09037e-01		

was used to model the data. As each data vector $[y(k-1)\ u(k-1)\ u(k-2)]^T$ was considered as a candidate RBF center, there were $n_M = 210$ regressors in the regression model (1). The variance of the RBF kernel function was chosen to be 1.69.

The OLS algorithm selected a 60-term model, the UROLS constructed a 46-term model, and the LROLS algorithm constructed a 34-term model. The mean square error values over the training and testing sets for these three models are given in Table IV. The constructed RBF model was used to generate the one-step prediction $\hat{y}(k)$ of the system output according to (23). The iterative model output $\hat{y}_d(k)$ was also produced using

$$\hat{y}_d(k) = f_{RBF}(\hat{y}_d(k-1), u(k-1), u(k-2)). \qquad (24)$$

The one-step model prediction and iterative model output for the 34-term model selected by the LROLS algorithm are shown in Fig. 5, in comparison with the system output.

V. Conclusions

A locally regularised OLS algorithm has been presented. The proposed algorithm combines both the advantages of OLS model selection, which has ability to select only those significant regressors to explain training data, and local regularisation, which enforces sparsity of models. As regularisation is introduced in the orthogonal weight space, computational requirements of the iterative model selection procedure are very simple and straightforward. A further advantage of this algorithm is that when to terminate the model selection procedure can be made easily based only on the training data, thus avoiding the costly cross-validation using a separate testing data set.

References

[1] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, Vol.50, No.5, pp.1873–1896, 1989.

[2] S. Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.2, No.2, pp.302–309, 1991.

[3] A.E. Hoerl and R.W. Kennard, "Ridge regression: biased estimation for non-orthogonal problems," *Technometrics*, Vol.12, pp.55–67, 1970.

[4] C.M. Bishop, "Improving the generalisation properties of radial basis function neural networks," *Neural Computation*, Vol.3, No.4, pp.579–588, 1991.

[5] S. Chen, E.S. Chng and K. Alkadhimi, "Regularised orthogonal least squares algorithm for constructing radial basis function networks," *Int. J. Control*, Vol.64, No.5, pp.829–837 1996.

[6] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, Vol.4, No.3, pp.415–447, 1992.

[7] M.E. Tipping, "The relevance vector machine," in Sara A. Solla, Todd K. Leen and Klaus-Robert Müller, eds., *Advances in Neural Information Processing Systems 12*, Cambridge, MA: MIT Press, 2000.

[8] S.A. Billings, S. Chen and R.J. Backhouse, "The identification of linear and non-linear models of a turbocharged automotive diesel engine," *Mechanical Systems and Signal Processing*, Vol.3, No.2, pp.123–142, 1989.

TABLE III

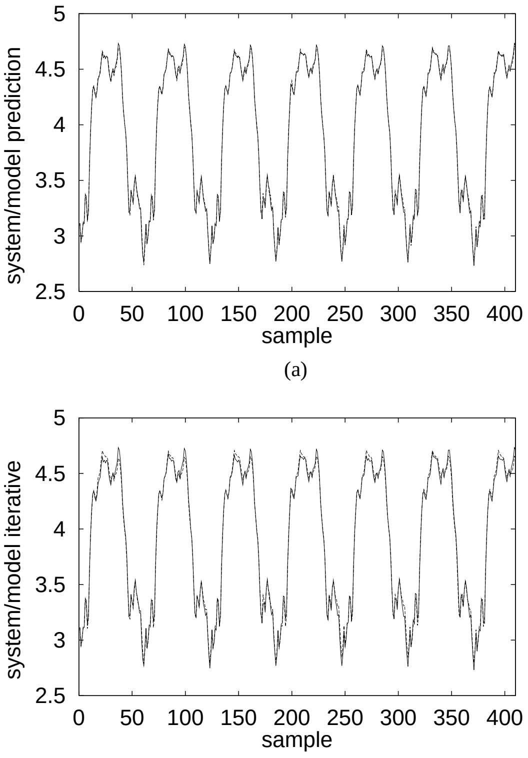
LROLS selection for the simple scalar function modelling after $\boldsymbol{\lambda}$ has converged.

stage l	$1-\sum[\text{rerr}]_l$	weight θ_l	regularizer λ_l
1	0.64850542	1.87494e+00	2.53227e-01
2	0.28873137	-1.70014e+00	1.81540e-01
3	0.25008959	-1.00970e+00	2.01490e-01
4	0.23493277	5.67310e-01	8.64601e-01
5	0.23367247	4.17979e-01	1.36357e+00
6	0.23328275	-1.51352e-01	6.93984e-01
7	0.23328275	-9.49873e-10	5.67623e+07
8	0.23328275	-2.79967e-10	1.11770e+08
9	0.23328275	7.14157e-11	1.03860e+07
10	0.23328275	-2.05313e-12	1.92708e+08
11	0.23328275	-1.32386e-13	7.85977e+08
12	0.23328275	2.29641e-14	4.09979e+08
13	0.23328275	-2.53260e-38	1.15132e+32
stop due to no term selected at 14 stage			
MSE over noisy training set: 0.159167			

TABLE IV

Comparison of modelling accuracy for the engine data example.

model	MSE for training	MSE for testing
60-term (OLS)	0.000336	0.000872
46-term (UROLS)	0.000427	0.000532
34-term (LROLS)	0.000435	0.000487



(a)

(b)

Fig. 5. System output $y(k)$ (solid) superimposed on (a) model one-step prediction $\hat{y}(k)$ (dashed) and (b) model iterative output $\hat{y}_d(k)$ (dashed). The model was selected by the LROLS.