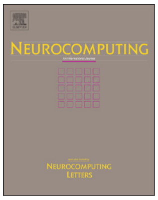
Neurocomputing 148 (2015) 551–560

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Elastic net orthogonal forward regression

Xia Hong [a,*], Sheng Chen [b,c]

[a] School of Systems Engineering, University of Reading, UK
[b] School of Electronics and Computer Science, University of Southampton SO17 1BJ, UK
[c] Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 24 March 2014
Received in revised form
4 June 2014
Accepted 4 July 2014
Communicated by K. Li
Available online 12 July 2014

Keywords:
Elastic net
Forward regression
Linear-in-the-parameters model
Regularisation
Leave one out errors
Cross validation

ABSTRACT

An efficient two-level model identification method aiming at maximising a model's generalisation capability is proposed for a large class of linear-in-the-parameters models from the observational data. A new elastic net orthogonal forward regression (ENOFR) algorithm is employed at the lower level to carry out simultaneous model selection and elastic net parameter estimation. The two regularisation parameters in the elastic net are optimised using a particle swarm optimisation (PSO) algorithm at the upper level by minimising the leave one out (LOO) mean square error (LOOMSE). There are two elements of original contributions. Firstly an elastic net cost function is defined and applied based on orthogonal decomposition, which facilitates the automatic model structure selection process with no need of using a predetermined error tolerance to terminate the forward selection process. Secondly it is shown that the LOOMSE based on the resultant ENOFR models can be analytically computed without actually splitting the data set, and the associate computation cost is small due to the ENOFR procedure. Consequently a fully automated procedure is achieved without resort to any other validation data set for iterative model evaluation. Illustrative examples are included to demonstrate the effectiveness of the new approaches.

1. Introduction

A large class of nonlinear models including some types of neural networks can be classified as linear models which include statistically linear or linear-in-the-parameters models [1,2]. These models have provable learning and convergence conditions and are well suited to be used for adaptive learning. They are amenable to parallel implementations, and have clear applications in many engineering applications [3–5]. A basic principle in practical nonlinear data modelling is the parsimonious principle that ensures the smallest possible model for the explanation of the observational data. For linear models, the forward orthogonal least squares (OLS) algorithm efficiently constructs parsimonious models [6,7], and has been a popular tool in associative neural networks such as fuzzy/neurofuzzy systems [8,9] and wavelet neural networks [10,11]. The algorithm has also been utilised in a wide range of engineering applications, e.g. aircraft gas turbine modelling [12], fuzzy control of multi-input multi-output (MIMO) nonlinear systems [13], power system control [14] and fault detection [15].

The main purpose of model construction is to produce good generalisation (capability to approximate system output for new input data that are not used in estimation), through two important aspects in system identification, i.e. choosing parsimonious model structure and deriving robust model parameter estimates for a smooth prediction surface (e.g. parameter control via regularisation). Fundamental to the evaluation of model generalisation capability is the concept of cross-validation (CV) [16], which can be used either in parameter estimation (e.g. tuning regularisation parameter [17,18], forming new parameter estimates [19]), or to derive model selection criteria based on information theoretic principles [20], which regularises model structure in order to produce parsimonious models, since a parsimonious model is favoured by these criteria. The regularisation assisted OLS (ROLS) approaches have been proposed based on minimising the leave one out criteria for regression, classification and probability density estimation [21]. In particular each radial basis function (RBF) unit has a tunable centre vector as well as an adjustable diagonal covariance matrix [21]. Specifically, at each forward regression stage of the model construction procedure one RBF unit's centre vector and diagonal covariance matrix are optimised using a particle swarm (PSO) algorithm. The PSO [22,23] constitutes a population based stochastic optimisation technique, which was inspired by the social behaviour of bird flocks or fish schools. The algorithm commences with random initialisation of a swarm of individuals, referred to as particles, within the specific problem's search space. It then endeavours to find a globally optimum solution by gradually adjusting the trajectory of each particle

[*] Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2014.07.008
0925-2312/© 2014 Elsevier B.V. All rights reserved.

towards its own best location and towards the best position of the entire swarm at each optimisation step. The PSO method is popular owing to its simplicity in implementation, ability to rapidly converge to a “reasonably good” solution and to “steer clear” of local minima. It has been successfully applied to a wide range of optimisation problems [24–28].

Regularisation methods are developed to carry out parameter estimation and model structure selection simultaneously [29,30]. It has been shown [31,32] that the parameter regularisation is equivalent to a maximised *a posterior* probability (MAP) estimate of parameters from Bayesian viewpoint by adopting a Gaussian prior for parameters. The regularisation [17,18] uses a penalty function on l^2 norms of the parameters. From the powerful Bayesian learning view point, a regularisation parameter is equivalent to the ratio of the related hyperparameter to the noise parameter, lending to an iterative evidence procedure for solving the optimal regularisation parameters [29,32]

Alternatively the model sparsity can be achieved by minimising the l^1 norm of the parameters. The l^1 norm minimisation is fundamental to the basis pursuit or least absolute shrinkage and selection operator (LASSO) [33,34]. The least angle regression (LAR) procedure [35] is developed for solving the problem efficiently. The Bayesian interpretation for LASSO is simply by adopting an Laplacian prior for parameters. The advantage of LASSO is that it can achieve much sparser models by forcing more parameters to zero, than models derived from the minimisation of the l^p norm, as most l^p norms will produce small, but nonzero, values. Unfortunately introducing nondifferentiable l^1 norm in the cost function brings difficulties of model parameter estimation and finding an appropriate l^1 regulariser.

Another disadvantage of using l^1 optimisation is that a group of correlated terms cannot be selected together, which is not desirable for the sake of interpretability of the model in some applications. On the other hand, the use of l^2 will improve model generalisation, but cannot be used for model selection by itself. Combining a locally regularised orthogonal least squares (LROLS) model selection [36] with D-optimality experimental design enhances model robustness [31].

Recently a promising concept of the elastic net (EN) has been proposed by minimising the l^1 and l^2 norms of the parameters together [30]. The EN keeps the model sparsity of LASSO, while strongly correlated terms tend to be in or out of the model together. It is shown that the elastic net problem can be transformed into an equivalent LASSO problem on an augmented data, based on which the LAR procedure is applicable, referred to as LARS-EN [30]. Note that because there are two regularisation parameters in the elastic net, the cross validation has to be performed over a two-dimensional space. The tenfold cross validation was used in the choosing two regularisation parameters by searching over a grid of l^2 norm regularisation parameter values. Then for each setting of the l^2 norm regularisation parameter, the algorithm LARS-EN produces the entire solution path of the elastic net, which is used to select l^1 norm regularisation parameter by tenfold CV. Clearly this may not yield the optimal parameters if the grid search is set at a coarse level, but increasing the grid search at a very fine level would inevitably increase the computational cost. It would be desirable that the two regularisation parameters can be optimised simultaneously based on cross validation as well as in an efficient manner.

In this paper we propose an efficient model identification method aiming at maximising a model's generalisation capability. The paper contains two elements of novel contribution. Firstly an elastic net cost function is defined and applied based on orthogonal decomposition, which facilitates the automatic model structure selection process with no need of using a predetermined error tolerance to terminate the forward selection process. Secondly an original derivation of analytical evaluation of LOOMSE is presented based on the resultant ENOFR models without actually splitting the data set. Consequently a fully automated procedure is achieved without resort to any other validation data set for iterative model evaluation. The algorithm has a two level structure. At the upper level, the two regularisation parameters in the elastic net are optimised using PSO by minimising the LOOMSE. At the lower level are the simultaneous model selection and elastic net parameter estimation. Illustrative examples are included to demonstrate the effectiveness of the new approaches.

2. Preliminaries

Consider the general nonlinear system represented by the nonlinear model [37]:

$$y(k) = f(\mathbf{x}(k)) + e(k), \tag{1}$$

where $\mathbf{x}(k) \in \Re^m$ denotes the system input vector and $y(k)$ is the system output variable, respectively. $e(k)$ is the system white noise and $f(\bullet)$ is the unknown system mapping. The system model (1) is to be identified from an observation data set $D_N = \{\mathbf{x}(k), y(k)\}_{k=1}^N$ using some suitable functional which can approximate $f(\bullet)$ with arbitrary accuracy. One class of such functionals is the kernel regression model of the form:

$$y(k) = \hat{y}(k) + e(k) = \sum_{i=1}^{n_M} \theta_i \phi_i(\mathbf{x}(k)) + e(k), \tag{2}$$

where $\hat{y}(k)$ denotes the model output, θ_i are the model weights, $\phi_i(\mathbf{x}(k))$ are the regressors, and n_M is the total number of candidate regressors or model terms.

By letting $\boldsymbol{\phi}_i = [\phi_i(\mathbf{x}(1)) \cdots \phi_i(\mathbf{x}(N))]^T$, for $1 \le i \le n_M$, and defining

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \quad \boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{n_M}],$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{n_M} \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix}, \tag{3}$$

the regression model (2) can be written in the matrix form

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{e}. \tag{4}$$

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

$$\boldsymbol{\Phi} = \mathbf{WA}, \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,n_M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n_M-1,n_M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{6}$$

and

$$\mathbf{W} = [\mathbf{w}_1 \ldots \mathbf{w}_{n_M}] \tag{7}$$

with columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \ne j$. The regression model (4) can alternatively be expressed as

$$\mathbf{y} = \mathbf{Wg} + \mathbf{e}, \tag{8}$$

where the orthogonal weight vector $\mathbf{g} = [g_1 \cdots g_{n_M}]^T$ satisfy the triangular system $\mathbf{A}\boldsymbol{\theta} = \mathbf{g}$, which can be used to determine model parameters $\boldsymbol{\theta}$, given $\mathbf{A}$ and $\mathbf{g}$.

3. Automatic kernel regression model construction algorithm using ENOFR assisted by PSO

3.1. Elastic net orthogonal forward regression

For any fixed positive λ_1 and λ_2, the naive elastic net (NEN) criterion is defined as [30]

$$L(\lambda_1,\lambda_2,\boldsymbol{\theta}) = \|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\theta}\|^2+\lambda_2\|\boldsymbol{\theta}\|^2+\lambda_1\|\boldsymbol{\theta}\|_1 \tag{9}$$

where $\|\bullet\|$ denotes Euclidean norm, and $\|\boldsymbol{\theta}\|_1=\sum_{i=1}^{n_M}|\theta_i|$. The naive elastic net estimator is the minimiser of

$$\hat{\boldsymbol{\theta}}_{NEN}=\arg\min_{\boldsymbol{\theta}}\{L(\lambda_1,\lambda_2,\boldsymbol{\theta})\} \tag{10}$$

This can be transformed into an equivalent LASSO problem on an augmented data, based on which the LAR procedure is applicable, referred to as LARS-EN [30]. The EN has some desirable properties, as it maintains the model sparsity of LASSO, but not as aggressive as LASSO in excluding correlated terms in the model. This is because these terms tend to be in or out of the model together as a result of the l^2 norm regularisation [30]. Note that there is no analytical solution to (10) unless the model terms are orthogonal.

The key to the proposed concept of ENOFR is to consider the following orthogonal elastic net (NEN) criterion based on (8)

$$L_e(\lambda_1,\lambda_2,\mathbf{g}) = \|\mathbf{y}-\mathbf{W}\mathbf{g}\|^2+\lambda_2\|\mathbf{g}\|^2+\lambda_1\|\mathbf{g}\|_1 \tag{11}$$

The naive elastic net solution for $\mathbf{g}$ is obtained by setting the subderivatives $\partial L_e/\partial\mathbf{g}=\mathbf{0}$, that is,

$$\mathbf{W}^T\mathbf{y}-\frac{\lambda_1}{2}\,\text{sign}(\mathbf{g})=(\mathbf{W}^T\mathbf{W}+\lambda_2\mathbf{I})\mathbf{g}. \tag{12}$$

where $\mathbf{I}$ is an identity matrix of appropriate dimension and $\text{sign}(\mathbf{g})=[\text{sign}(g_1),\ldots,\text{sign}(g_{n_M})]^T$, where

$$\text{sign}(s)\begin{cases} =1 & \text{if } s>0\\ =-1 & \text{if } s<0\\ \in[-1,\ 1] & \text{if } s=0\end{cases} \tag{13}$$

Multiplying $2\mathbf{g}^T$ to both sides of (12) yields

$$2\mathbf{g}^T\mathbf{W}^T\mathbf{y}-\lambda_1\|\mathbf{g}\|_1=2\mathbf{g}^T(\mathbf{W}^T\mathbf{W}+\lambda_2\mathbf{I})\mathbf{g}. \tag{14}$$

Substitute (14) into (11) to yield

$$\begin{aligned}L_e(\lambda_1,\lambda_2,\mathbf{g}) &= \mathbf{y}^T\mathbf{y}-2\mathbf{g}^T\mathbf{W}^T\mathbf{y}+\mathbf{g}^T\mathbf{W}^T\mathbf{W}\mathbf{g}+\lambda_2\|\mathbf{g}\|^2+\lambda_1\|\mathbf{g}\|_1\\ &= \mathbf{y}^T\mathbf{y}-\mathbf{g}^T\mathbf{W}^T\mathbf{W}\mathbf{g}-\lambda_2\|\mathbf{g}\|^2\end{aligned} \tag{15}$$

Normalising by $\mathbf{y}^T\mathbf{y}$,

$$L_e(\lambda_1,\lambda_2,\mathbf{g})/(\mathbf{y}^T\mathbf{y})=1-\sum_{i=1}^{n_M}(\mathbf{w}_i^T\mathbf{w}_i+\lambda_2)(g_i^{(NEN)})^2/(\mathbf{y}^T\mathbf{y}). \tag{16}$$

where the superscript $^{(\text{NEN})}$ denotes the naive elastic net solution. The elastic net error reduction ratio is defined by

$$[\text{eNerr}]_i=(\mathbf{w}_i^T\mathbf{w}_i+\lambda_2)(g_i^{(NEN)})^2/(\mathbf{y}^T\mathbf{y}),\quad i=1,\ldots,n_M \tag{17}$$

where $g_i^{(NEN)}$, $i=1,\ldots n_M$ are the solution of (12), given by

$$g_i^{(NEN)}=\left(\frac{\mathbf{w}_i^T\mathbf{w}_i}{\mathbf{w}_i^T\mathbf{w}_i+\lambda_2}\left|g_i^{(LS)}\right|-\frac{\lambda_1/2}{\mathbf{w}_i^T\mathbf{w}_i+\lambda_2}\right)_+\ \text{sign}(g_i^{(LS)}) \tag{18}$$

with $g_i^{(LS)}=\mathbf{w}_i^T\mathbf{y}/\mathbf{w}_i^T\mathbf{w}_i$ and

$$z_+=\begin{cases} z & \text{if } z>0\\ 0 & \text{if } z\le 0\end{cases} \tag{19}$$

Based on this ratio, significant regressors can be selected in a forward regression procedure. From (17) and (18) it is obvious that, the terms that are selected into the model using the proposed algorithm, and the associated parameter values, are affected by the values of λ_1 and λ_2. Using a simple example we further analyse this effect. A model is to be constructed by three candidate regressors with the same magnitude, $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$, $\boldsymbol{\phi}_3$, as shown in Fig. 1, in which we compare three cases; (i) $\lambda_1=0$, $\lambda_2=0$; (ii) $\lambda_1\neq 0$, $\lambda_2=0$; and (iii) $\lambda_1\neq 0$, $\lambda_2\neq 0$. The following observations can be noted about the first two forward regression steps;

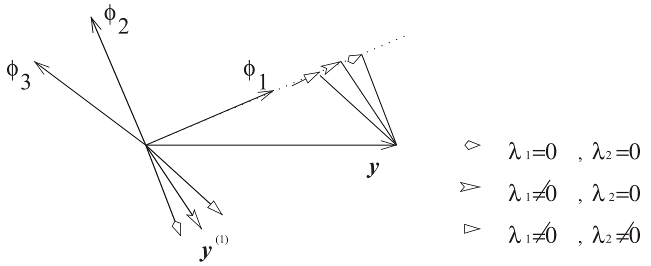


Fig. 1. An illustration of elastic net orthogonal forward regression.

- For the first forward regression step, if $\lambda_1=0$, $\lambda_2=0$, then the resultant model based on $[\text{eNerr}]_1$ for model term selection is equivalent to selecting the model term which can produce the largest projection by $\mathbf{y}$, or the most correlated term to $\mathbf{y}$.
- In all cases, $\boldsymbol{\phi}_1$ is selected from three candidate regressors at the first step, because it produces the highest value of $[\text{eNerr}]_1$.
- At the first forward regression stage, the effects of any nonzero λ_1 and λ_2 are that the explained output variance by the first selected regressor is reduced in comparison with a model using the least square parameter estimate $g_1^{(LS)}$, because it can be seen from (18) that the magnitude of $g_1^{(NEN)}$ is reduced by scaling due to λ_2, followed by thresholding due to λ_1.
- After the first regression step, let the remainder of the output vector be denoted by $\mathbf{y}^{(1)}$. The second forward regression step is to select more significant regressor between $\boldsymbol{\phi}_2$ and $\boldsymbol{\phi}_3$ based on $\mathbf{y}^{(1)}$.
- For the second forward regression step, if $\lambda_1=0$, $\lambda_2=0$, then $\boldsymbol{\phi}_2$ will be selected, because the resultant model by using $[\text{eNerr}]_2$ for model term selection would be equivalent to selecting the model term which can produce the largest projection by $\mathbf{y}^{(1)}$.
- It can be seen from Fig. 1 that values of λ_1 and λ_2 affect the direction of $\mathbf{y}^{(1)}$. As a result, $\boldsymbol{\phi}_2$ may no longer produce the largest projection by $\mathbf{y}^{(1)}$, and it is possible that $\boldsymbol{\phi}_3$ is selected as the significant regressor in the second forward regression step, not $\boldsymbol{\phi}_2$.

The automatic model term selection property of naive elastic net is also explained as follows. Note that for $\lambda_1=0$, $[\text{eNerr}]_i$ becomes the model term selective criterion, the regularised error reduction ratio $[\text{rerr}]_i$, as defined in [38]. In order to produce a sparse model containing n_s ($\ll n_M$) significant regressors, a chosen tolerance ξ ($0<\xi<1$) needs to be preset, and the selection process is terminated at the n_sth stage when

$$1-\sum_{l=1}^{n_s}[\text{rerr}]_l<\xi \tag{20}$$

is satisfied [38]. However using elastic net orthogonal forward regression ($\lambda_1>0$), there is no need of setting ξ. This is because the cost function contains sparsity inducing l^1 norm so that some parameters will be zeros and $[\text{eNerr}]_i$ can return exact zero values during the selection process. The model selection is terminated at the (n_s+1)–th stage when $[eNerr]_{n_s+1}=0$, producing a sparse model containing n_s ($\ll n_M$) significant regressors automatically. The naive elastic net orthogonal forward regression (ENOFR) algorithm based on the modified Gram–Schmidt scheme is given in Appendix A, for a given $\boldsymbol{\lambda}=[\lambda_1,\lambda_2]^T$.

Finally the elastic net (EN) parameter estimate is defined by

$$g_i^{(EN)}=\left(\left|g_i^{(LS)}\right|-\frac{\lambda_1/2}{\mathbf{w}_i^T\mathbf{w}_i}\right)_+\ \text{sign}(g_i^{(LS)}) \tag{21}$$

which produces the elastic net parameter estimates for the n_s term model selected using the algorithm of Appendix A. This step inflates $g_i^{(NEN)}$ by the original shrinkage amount $(\mathbf{w}_i^T\mathbf{w}_i+\lambda_2)/\ \mathbf{w}_i^T\mathbf{w}_i$ and aims to overcome the double shrinkage problem of naive elastic net estimator [30]. This means that the effect of l^2 norm regularisation to parameter estimation is undone by this step, which is helpful to reduce bias in the naive elastic net estimator which could be too large.

We point out that as this rescaling step happens after the model terms selection so the existence of λ_2 has an impact on model structure compared with the case of $\lambda_2=0$, e.g. using the previous example, a two term model could be composed by $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ for $\lambda_2=0$, but by $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_3$ for $\lambda_2 \neq 0$. The effect of l^2 norm regularisation in selecting groups (correlated terms) was analysed [30]. For our proposed algorithm, the analysis to the first two regression steps can be extended to any regression steps. As a result of combined effect of λ_1 and λ_2, the explained output variance by selected regressors at earlier regression steps are reduced in comparison with a model using the least square parameter $g_i^{(LS)}$. Effectively this would allow the model output to be further explained by other regressors, that are correlated to previously selected regressors, to enter the model at later stages. Therefore the proposed algorithm has a similar effect to the original elastic net, of keeping correlated terms in the model, which is advantageous in that less variable models could be produced to provide physical insights on the causal relationships of the systems from large data sets [30].

3.2. Choosing regularisation parameters by optimising the LOOMSE using PSO

Cross validation criteria are metrics that measure a model's generalisation capability. To optimise the model generalisation capability, the model selection criteria are often based on cross-validation [16,39]. Due to its simplicity, a popular version of cross-validation is the so-called leave one out (LOO) cross validation. It is also known that LOO is inconsistent [40]. That is, the probability of selecting the model with the best predictive ability does not converge to one as the total number of data samples approaching infinity. Some theoretical and empirical comparisons for model selection using different cross validation schemes are discussed [41,42].

Consider the general model selection problem from a set of K predictors due to models produced using different setting of regularisation parameters of $\boldsymbol{\lambda}$ indexed by $j=1,2,\ldots,K$. Denote these predictors as $\hat{y}_j(k)$ if they are identified using all N data points. The idea of LOO is that, for any predictor, each data point in the estimation data set D_N is sequentially set aside in turn, a model is estimated using the remaining $(N-1)$ data, and the prediction error is calculated based on the data point that was removed. That is, for $k=1,\ldots,N$, the jth model $(\forall j)$ is estimated by removing the kth data point from the estimation set. The output of the model based on $(N-1)$ data points (with the kth data point removed) is denoted by $\hat{y}_j^{(-k)}(k)$, and the LOO prediction error is calculated as

$$e_j^{(-k)}(k)=y(k)-\hat{y}_j^{(-k)}(k) \tag{22}$$

Finally the leave one out mean square error (LOOMSE) is obtained by computing the average of all these prediction errors as $J(\boldsymbol{\lambda})=E[[e^{(-k)}(k)]^2]$. The regularisation parameter vector associated with the minimal LOOMSE is chosen, i.e.

$$\boldsymbol{\lambda}_{opt}=\arg\left\{\min_{\boldsymbol{\lambda}}\left\{J(\boldsymbol{\lambda})=\frac{1}{N}\sum_{k=1}^{N}[e_j^{(-k)}(k)]^2,\quad \forall j\right\}\right\} \tag{23}$$

and the resultant model is selected.

The above illustrates the concept of the leave one out cross-validation procedure, which seems to be computationally expensive. However, if $f(\bullet)$ is modelled using linear models via least square method, there is an elegant way to generate LOOMSE [43], without actually sequentially splitting the estimation data set by using the Sherman–Morrison–Woodbury theorem [43]. In the following we show that LOOMSE based on the proposed ENOFR estimator can also be evaluated efficiently without actually sequentially splitting the estimation data set.

From (12) and (21), the elastic net parameter estimator based on a specified $\boldsymbol{\lambda}$ using N data points can be represented by

$$\mathbf{g}^{(EN)}=\mathbf{H}^{-1}\left(\mathbf{W}^T\mathbf{y}-\frac{\lambda_1}{2}\,\mathrm{sign}(\mathbf{g}^{(EN)})\right) \tag{24}$$

where $\mathbf{H}=\mathbf{W}^T\mathbf{W}$. The model residual is

$$\begin{aligned}e(k)&=y(k)-(\mathbf{g}^{(EN)})^T\mathbf{w}(k)\\&=y(k)-\left(\mathbf{y}^T\mathbf{W}-\frac{\lambda_1}{2}[\mathrm{sign}(\mathbf{g}^{(EN)})]^T\right)\mathbf{H}^{-1}\mathbf{w}(k)\end{aligned} \tag{25}$$

If the data sample indexed at k is removed from estimation data set, the leave one out elastic net parameter estimator obtained by using only $(N-1)$ data points is given by

$$\begin{aligned}\mathbf{g}^{(EN,-k)}&=[\mathbf{H}^{(-k)}]^{-1}\\&\times\left([\mathbf{W}^{(-k)}]^T\mathbf{y}^{(-k)}-\frac{\lambda_1}{2}\mathrm{sign}(\mathbf{g}^{(EN,-k)})\right)\end{aligned} \tag{26}$$

in which $\mathbf{H}^{(-k)}=[\mathbf{W}^{(-k)}]^T\mathbf{W}^{(-k)}$, $\mathbf{W}^{(-k)}$ and $\mathbf{y}^{(-k)}$ denote the resultant regression matrix and output vector respectively. The leave one out error evaluated at k is given by

$$\begin{aligned}e^{(-k)}(k)&=y(k)-[\mathbf{g}^{(EN,-k)}]^T\mathbf{w}(k)\\&=y(k)-\left([\mathbf{y}^{(-k)}]^T\mathbf{W}^{(-k)}-\frac{\lambda_1}{2}[\mathrm{sign}(\mathbf{g}^{(EN,-k)})]^T\right)\\&\times[\mathbf{H}^{(-k)}]^{-1}\mathbf{w}(k)\end{aligned} \tag{27}$$

It can be shown that

$$\mathbf{H}^{(-k)}=\mathbf{H}-\mathbf{w}(k)\mathbf{w}^T(k) \tag{28}$$

$$[\mathbf{y}^{(-k)}]^T\mathbf{W}^{(-k)}=\mathbf{y}^T\mathbf{W}-y(k)\mathbf{w}^T(k) \tag{29}$$

Applying the matrix inversion lemma to (28), yields

$$\begin{aligned}[\mathbf{H}^{(-k)}]^{-1}&=[\mathbf{H}-\mathbf{w}(k)\mathbf{w}^T(k)]^{-1}\\&=\mathbf{H}^{-1}+\frac{\mathbf{H}^{-1}\mathbf{w}(k)\mathbf{w}^T(k)\mathbf{H}^{-1}}{1-\mathbf{w}^T(k)\mathbf{H}^{-1}\mathbf{w}(k)}\end{aligned} \tag{30}$$

and

$$[\mathbf{H}^{(-k)}]^{-1}\mathbf{w}(k)=\frac{\mathbf{H}^{-1}\mathbf{w}(k)}{1-\mathbf{w}^T(k)\mathbf{H}^{-1}\mathbf{w}(k)} \tag{31}$$

Substituting (29) and (31) into (27), yields

$$\begin{aligned}e^{(-k)}(k)&=y(k)-\left(\mathbf{y}^T\mathbf{W}-y(k)\mathbf{w}^T(k)-\frac{\lambda_1}{2}[\mathrm{sign}(\mathbf{g}^{(EN,-k)})]^T\right)\\&\times\frac{\mathbf{H}^{-1}\mathbf{w}(k)}{1-\mathbf{w}^T(k)\mathbf{H}^{-1}\mathbf{w}(k)}\\&=\frac{y(k)-(\mathbf{y}^T\mathbf{W}-\frac{\lambda_1}{2}[\mathrm{sign}(\mathbf{g}^{(EN,-k)})]^T)\mathbf{H}^{-1}\mathbf{w}(k)}{1-\mathbf{w}^T(k)\mathbf{H}^{-1}\mathbf{w}(k)}\end{aligned} \tag{32}$$

The leave one out mean square error (LOOMSE) can be calculated as

$$J(\boldsymbol{\lambda})=\frac{1}{N}\sum_{k=1}^{N}[e^{(-k)}(k)]^2 \tag{33}$$

$$
\begin{aligned}
J(\lambda) &\approx \frac{1}{N}\sum_{k=1}^{N}\left[\frac{e(k)}{1-\mathbf{w}^T(k)\mathbf{H}^{-1}\mathbf{w}(k)}\right]^2 \\
J(\lambda) &= \frac{1}{N}\sum_{k=1}^{N}\left[\frac{e(k)}{1-\sum_{i=1}^{n_s}[w_i(k)]^2/(\mathbf{w}_i^T\mathbf{w}_i)}\right]^2 \qquad (34)
\end{aligned}
$$

by making use of (25) and assuming that $\text{sign}(\mathbf{g}^{(EN,-k)}) = \text{sign}(\mathbf{g}^{(EN)})$ holds for most k. This assumption is mild because only one data sample is removed at a time, based on significant regressors selected in a forward regression manner.

It is simple to evaluate $J(\lambda)$ as a result of the following reasons.

- Firstly the proposed elastic net cost function is based on parameter regularisation within an orthogonal space, making it possible to derive a closed form expression for the parameters of the elastic net.
- Secondly we provide the above original derivation to show that the LOOMSE based on models using elastic net estimator can be analytically approximately evaluated without actually splitting the data by making use of the matrix inversion lemma and a mild assumption.
- Thirdly as a byproduct of the orthogonalisation procedure $\mathbf{H}$ is diagonal, so that the evaluation of $e^{(-k)}(k)$ does not involve any matrix inversion and has a very small computational cost (see (32)).

We apply the PSO algorithm to solve (23), as shown in Appendix B. The complete algorithm can be illustrated with reference to the schematic diagram of Fig. 2. The algorithm has a two layer structure. The upper level is the PSO with population size of S (Appendix B). It learns the two optimal regularisation parameters based on the LOOMSE values provided by the lower level of S particles. At the lower level, each particle performs the ENOFN algorithm over the iterations, with each iteration consisting of two stages; (i) select a subset model based on the naive elastic net parameter estimator using the MGS algorithm in Appendix A; and (ii) determine the elastic net model parameters for the selected model terms using (21) and then calculate the associated LOOMSE using (32) and (33).

The computation cost of the PSO is dominated by that of the cost function evaluation. So the total computational complexity of the proposed two-level learning scheme is determined by the total number of function evaluations of PSO ($S \times I_{max}$), multiplying the average computation cost of each particle, i.e, that of the elastic forward regression. The latter is in the order of $O(N)$, which is further scaled by the product of candidate and final model size $n_s \times n_M$. Note that n_M can be set much lower than N if the latter is too large in order to save computation cost. The computational cost of the proposed algorithm is much smaller than conventional cross validation approaches of grid search over a two-dimensional space. For example if the ten-fold cross validation is used for a very coarse grid search of 3 by 3 on $\boldsymbol{\lambda}$, its computation cost is roughly the same as the proposed algorithm with $S=9$ and $I_{max}=9$ which is found to be appropriate from our experience. However the grid search of 3 by 3 on $\boldsymbol{\lambda}$ is likely to be too coarse to produce reasonably solutions.

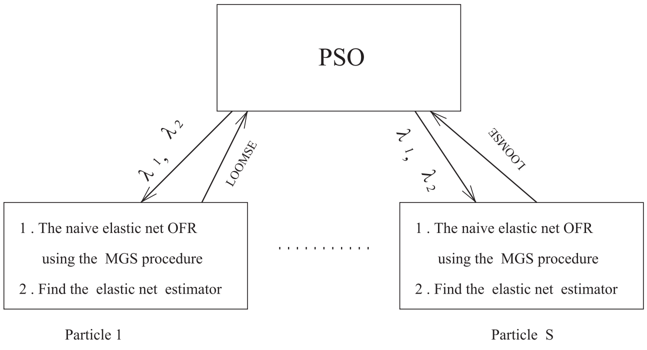

PSO
λ 1, λ 2
LOOMSE
λ 1, λ 2
LOOMSE
1 . The naive elastic net OFR
using the MGS procedure
2 . Find the elastic net estimator
. . . .
1 . The naive elastic net OFR
using the MGS procedure
2 . Find the elastic net estimator
Particle 1
Particle S

Fig. 2. A schematic diagram of the proposed ENOFR using PSO.

4. Modeling examples

In this section we demonstrate the effectiveness of the proposed algorithm using simulations. One example on multivariate linear regression and one example nonlinear static function approximation are presented, followed by two examples on data from real nonlinear dynamical systems.

4.1. Multivariate linear regression

Prostate cancer example was taken from a study of prostate cancer [30,44]. The inputs are eight clinical measures: log(cancer volume) (lcavol), log(prostate weight) (lweight), age, the logarithm of the amount of benign prostatic hyperplasia (lbph), seminal vesicle invasion (svi), log(capsular penetration) (lcp), Gleason score (gleason) and percentage Gleason score 4 or 5 (pgg45). The response is the logarithm of prostate-specific antigen (lpsa). The prostate cancer data were divided into two parts: a training set with 67 observations and a test set with 30 observations. We use the linear model with scaled inputs so that each has zero mean and unit variance, and construct our model using the proposed algorithm. The search space of PSO was set $[10, 20]$ for λ_1, and $[100, 1000]$ for λ_2. $S=20$, $I_{max}=20$ were predetermined. The proposed algorithm automatically selects a final model with 4 terms, produced by regularisation parameters $\lambda_1 = 11.7509$, $\lambda_2 = 138.7415$ found by the PSO based on the LOOMSE criterion without using another validation data set. Table 1 shows the test mean square error against the results of different methods in [30]. Note that in [30] tenfold cross validation of the training set was used in the grid search of the regularisation parameters, enabling the standard deviation in the brackets to be obtained. In our algorithm the training data set was not actually split up. The result of our model is better than all other methods except for the original elastic net method.

4.2. Nonlinear static function approximation

Consider using a RBF network to approximate an unknown scalar function

$$
f(x) = \frac{\sin(x)}{x} \qquad (35)
$$

A data set of two hundred points was generated from $y = f(x) + \xi$, where the input x was uniformly distributed in $[-10,10]$ and the noise ξ was Gaussian with zero mean and standard deviation 0.2.

Table 1
Prostate cancer data: comparing different methods. The results of the first five methods were quoted from [30].

Method	Test mean square error	Variables selected
Ordinary least squares [30]	0.586 (0.184)	All
Ridge regression [30]	0.566 (0.188)	All
Lasso [30]	0.499 (0.161)	(1, 2, 4, 5, 8)
Naive elastic net [30]	0.566 (0.188)	All
Elastic net [30]	0.381 (0.105)	(1, 2, 5, 6, 8)
The proposed ENOFR	0.4563	(1, 2, 5, 4)

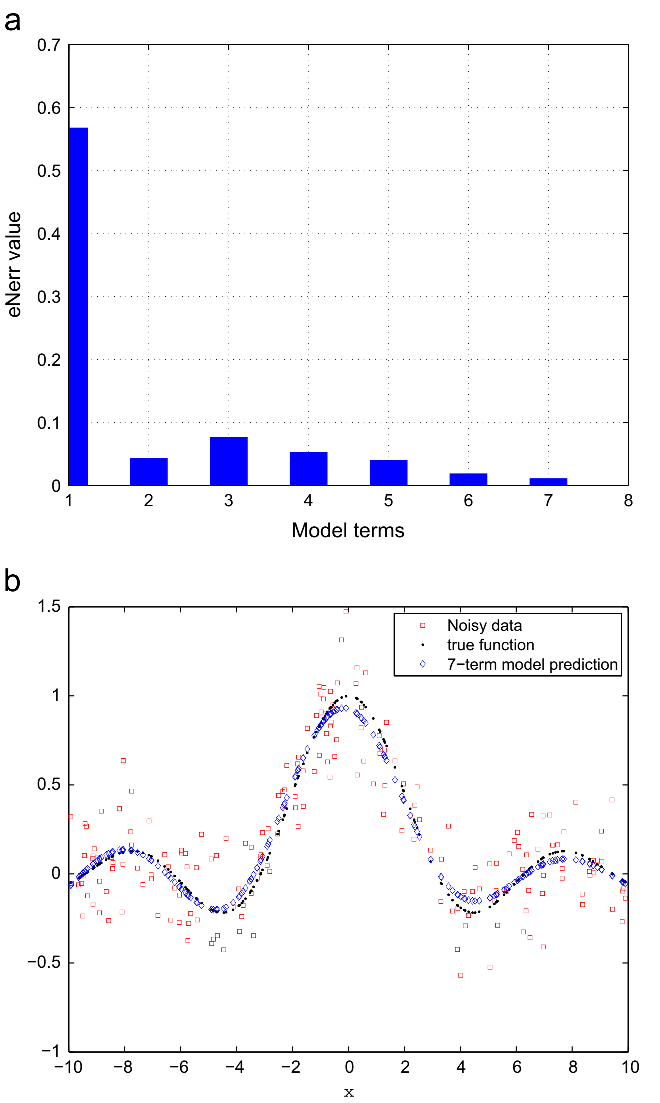


Fig. 3. The modeling results of the simple scalar function problem by the selected model ($\lambda_1 = 0.0465$, $\lambda_2 = 0.145$); (a) 7 nonzero $eNerr_j$ values during the elastic net orthogonal forward regression steps; and (b) model predictions of the 7-term model.

The data were very noisy. The Gaussian function

$$\phi_i(x) = \exp\left(-\frac{(x - c_i)^2}{2\tau^2} \right) \tag{36}$$

was used as the basis function to construct a RBF model, with a kernel width $\tau^2 = 10$. All the two hundred data points were used as the candidate RBF centre set for c_i. The search space of PSO was set $[10^{-7}, 0.1]$ for λ_1, and $[10^{-7}, 1]$ for λ_2. S=5, $I_{max} = 5$ were predetermined. The proposed algorithm automatically selects a final model with only 7 terms, produced by regularisation parameters $\lambda_1 = 0.0465$, $\lambda_2 = 0.145$. These were automatically determined by the PSO based on the LOOMSE criterion without using another validation data set. Fig. 3(a) depicts $[eNerr]_j$ values against the forward regression process, which automatically terminated at the 8th step when $[eNerr]_8 = 0$. Fig. 3(b) depicts the model prediction of the resultant 7-term model in comparison to the noisy data used for training and the unknown true function. The resultant 7-term model produces a mean square error of 0.0015 with respect to the true function, illustrating the excellent model generalisation capability of the model in this particular problem.

For comparison we construct models using ENOFR algorithm introduced in the paper, except that for selecting $\boldsymbol{\lambda}$ tenfold cross validation was used, rather than LOOMSE with PSO. By setting a grid of $\lambda_1 = [10^{-7}, 10^{-5}, 10^{-4}, 10^{-3}, 0.1]$ and $\lambda_2 = [10^{-7}, 10^{-5}, 10^{-3}, 0.1, 1]$, 25 settings of $\boldsymbol{\lambda}$ are evaluated using tenfold cross validation. We used the same kernel width $\tau^2 = 10$, and for each fold all resultant 180 training data points were used as the candidate RBF centre set. The estimated computational cost is roughly nine times of using LOOMSE with PSO in terms of how many times the MGS algorithm is applied. We also assume that, due to the reduction of 10% in training data set size for tenfold cross validation, there is also 10% computational cost reduction. The best $\boldsymbol{\lambda}$ is found to be $\lambda_1 = 0.1$, $\lambda_2 = 0.001$. For each fold, a 7-term model was produced. With respect to the true function, the resultant mean square error for all data points over ten models is 0.0023 ± 0.0003 (mean ± standard deviation), illustrating that selecting $\boldsymbol{\lambda}$ using tenfold cross validation does not offer superior performance to the proposed algorithm for this particular problem.

4.3. Nonlinear dynamical system modeling

Example 1. The relationship between the fuel rack position (input $u(k)$) and the engine speed (output $y(k)$) is modelled for a Leyland TL11 turbocharged, direct injection diesel engine which is operated at a low engine speed. Detailed system description and experimental setup can be found in [45]. The data set, depicted in Fig. 4(a) and (b), contains 410 samples. The first 210 data samples were used in training and the last 200 data samples for model validation. The previous study has shown that the data set can be modeled adequately using the system input vector $\mathbf{x}(k) = [y(k-1), u(k-1), u(k-2)]^T$. The best Gaussian kernel model provided by the locally regularised orthogonal least squares (LROLS) algorithm with the LOO test score, consisting of 22 terms [46] and with the mean square error (MSE) values over the training and validation data sets of 0.000453 and 0.000490, respectively.

We use the Gaussian radial basis function (RBF) $\phi_i(\mathbf{x}(k)) = \exp\{-\|\mathbf{x}(k) - \mathbf{c}_i\|^2 / 2\tau^2\}$ to construct our model using the proposed algorithm, where $\tau^2 = 1.69$ was set empirically and is the same as that used in [46]. $\mathbf{c}_i$ were formed using all the training data samples. The search space of PSO was set $[10^{-6}, 0.01]$ for λ_1, and $[10^{-6}, 100]$ for λ_2. S=5, $I_{max} = 5$ were predetermined. The proposed algorithm automatically selects a final model with 26 terms where the regularisation parameters were found to be $\lambda_1 = 2.147 \times 10^{-5}$, $\lambda_2 = 10^{-6}$ by the PSO based on the LOOMSE criterion without using another validation data set. Fig. 4 (c) depicts $\log_{10}([eNerr]_j)$ values against the forward regression process, which automatically terminated at the 27th step as $[eNerr]_{27} = 0$. For this model the mean square error (MSE) values over the training and validation data sets are 0.000447 and 0.000470, respectively. Clearly the modelling results are comparable to that of [46], as it has a slightly better predictive performance than [46], but slightly larger model size.

For comparison we construct models using ENOFR, in which tenfold cross validation was used for selecting $\boldsymbol{\lambda}$, rather than using LOOMSE with PSO. By setting a grid of $\lambda_1 = [10^{-7}, 10^{-5}, 10^{-4}, 10^{-3}, 0.01]$ and $\lambda_2 = [10^{-7}, 10^{-5}, 10^{-3}, 1, 100]$, we evaluated 25 settings of $\boldsymbol{\lambda}$ using tenfold cross validation, in which 20 data points from first 210 data samples are sequentially preset as test data points for each fold producing ten different data partitions. We used the same kernel width $\tau^2 = 1.69$, and for each fold all resultant 190 training data points were used as the candidate RBF centre set. The resultant best model is from $\lambda_1 = 10^{-4}$, $\lambda_2 = 10^{-7}$. Over ten models

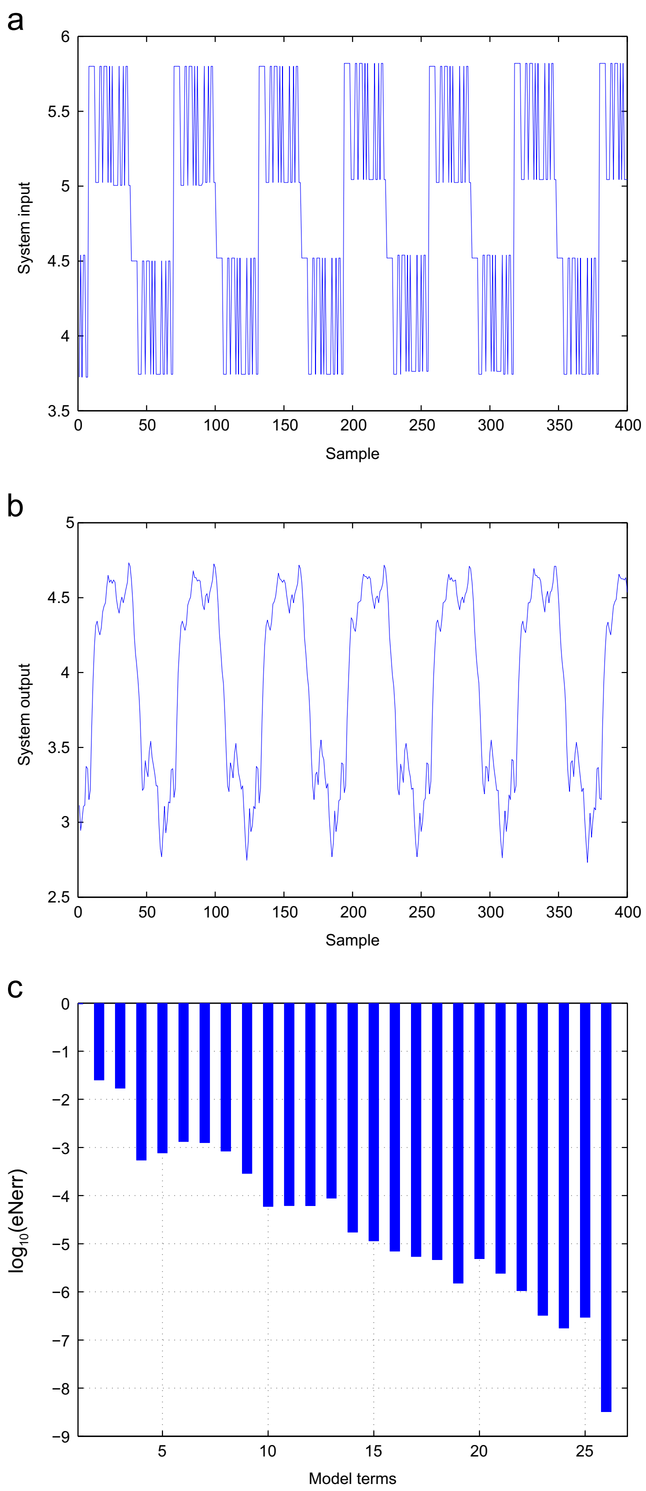


Fig. 4. Engine data set. (a) System input $u(t)$; (b) system output $y(t)$; and (c) the logarithm of 26 nonzero $eNerr_j$ values during the elastic net orthogonal forward regression steps for $\lambda_1 = 2.147 \times 10^{-5}$ and $\lambda_2 = 10^{-6}$.

produced from 10 different partitions, we recorded the mean square error over estimation data set as $0.000463 \pm 3.5833 \times 10^{-5}$ (mean ± standard deviation) and validation data set as $0.000500 \pm 1.4049 \times 10^{-5}$ (mean ± standard deviation), and the model size as 23.9 ± 1.1 (mean ± standard deviation). Clearly performances are also comparable but not superior to the proposed algorithm, because the estimated computational cost is roughly nine times of using LOOMSE with PSO.

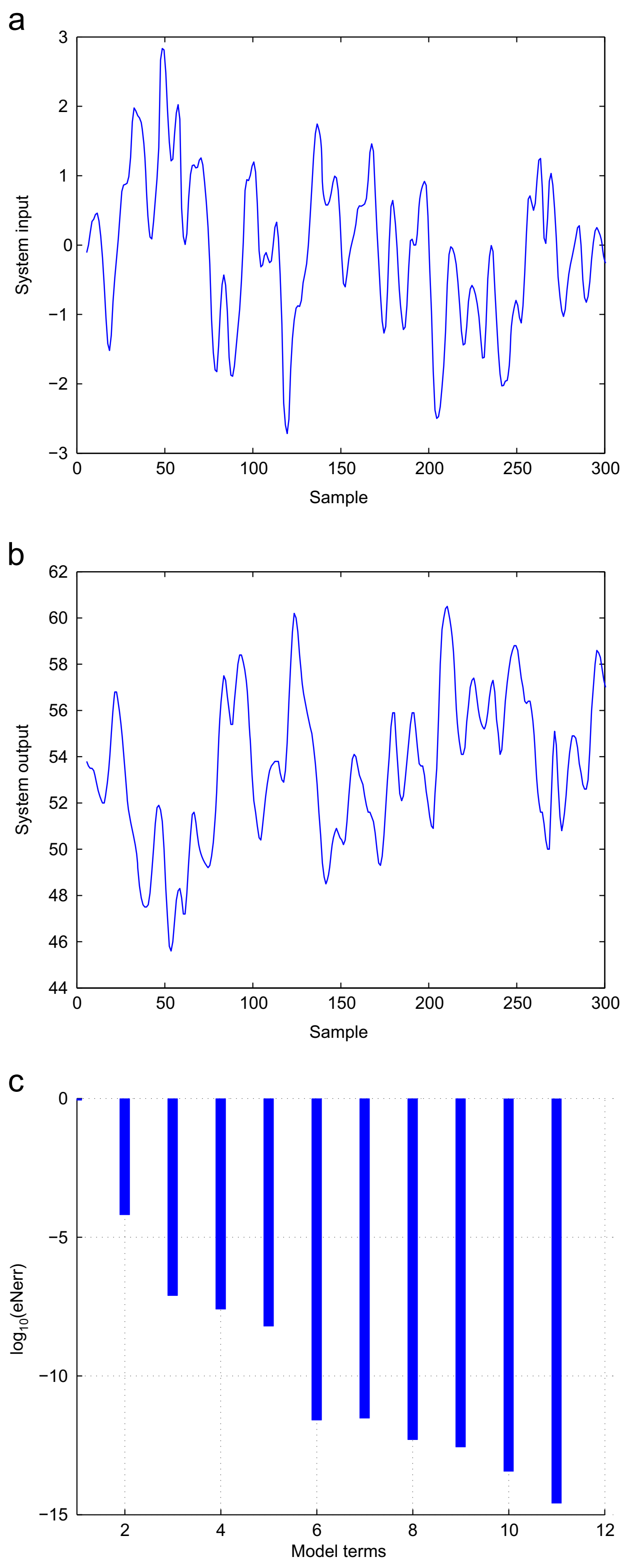


Fig. 5. Gas Furnace Data Set. (a) System input $u(t)$; (b) system output $y(t)$; and (c) the logarithm of 11 nonzero $eNerr_j$ values during the elastic net orthogonal forward regression steps for $\lambda_1 = 0.0003$ and $\lambda_2 = 23.9928$.

Example 2. The gas furnace data set (the time series J in [47]) contained 296 pairs of input–output points as depicted in Fig. 5 (a) and (b), where the input was the coded input gas feed rate and the output represented the CO_2 concentration from the gas furnace. The Gaussian radial basis function (RBF) $\phi_i(\mathbf{x}(k)) = \exp\{-(\|\mathbf{x}(k) - \mathbf{c}_i\|^2)/2\tau^2\}$ was used, with the system input vector

$\mathbf{x}(k)=[y(k-1), y(k-2), y(k-3), u(k-1), u(k-2), u(k-3)]^T$ and $\tau^2=1000$. From Fig. 5, it can be observed that the second half of the data set was different from the first half. Therefore, we used the even-number pairs $\{\mathbf{x}(k), y(k)\}$ for training and the odd-number pairs of $\{\mathbf{x}(k), y(k)\}$ for testing. $\mathbf{c}_i$ were formed using all the training data samples. The resultant model provided by the ROLS algorithm with the LOO test score. The search space of PSO was set $[10^{-6}, 0.01]$ for λ_1, and $[10^{-6}, 100]$ for λ_2. $S=5$, $I_{max}=5$ were predetermined. The proposed algorithm automatically selects a final model with 11 terms where the regularisation parameters were found to be $\lambda_1=0.0003$, $\lambda_2=23.9928$ by the PSO based on the LOOMSE criterion without using another validation data set. Fig. 5 (c) depicts $\log_{10}([eNerr]_j)$ values against the forward regression process, which automatically terminated at the 12th step as $[eNerr]_{12}=0$. For this model the mean square error (MSE) values over the training and validation data sets are 0.0493 and 0.0790, respectively. For comparison a previous study [21] has experimented on the regularised assisted OLS(ROLS) based on the LOO mean square error (referred to as ROLS-LOO algorithm) [46] and using the same common variance. The resultant model provided by the ROLS-LOO algorithm consists of 12 terms and has the mean square error (MSE) values over the training and validation data sets of 0.0474 and 0.0805 respectively. Clearly the modelling results of the proposed approaches are competitive.

Tenfold cross validation rather than LOOMSE was used to select λ based on ENOFR without PSO. Two grids of $\lambda_1=[10^{-7}, 10^{-5}, 10^{-4}, 10^{-3}, 0.01]$ and $\lambda_2=[10^{-7}, 10^{-5}, 10^{-3}, 1, 100]$ were predetermined to obtain 25 settings λ. Tenfold data partitions are based on sequentially taking 14 data points from the original training data samples as test data points, producing ten different data partitions. We used the same kernel width $\tau^2=1000$, and for each fold all resultant 132 training data points were used as the candidate RBF centre set. We obtained $\lambda_1=10^{-4}$, $\lambda_2=10^{-7}$. Over ten models obtained based on different training data set partitions, we recorded the mean square error over estimation data set as 0.0477 ± 0.0027 (mean ± standard deviation) and validation data set as 0.0811 ± 0.0025 (mean ± standard deviation), and the model size as 12.5 ± 0.5 (mean ± standard deviation). Note that the modeling performance is comparable, but the computational cost is approximately nine times of using LOOMSE with PSO.

5. Conclusions

Aiming at maximising a model's generalisation capability, this paper has proposed an efficient two-level model identification method for the linear-in-the-parameters models. At the lower level is the proposed ENOFR algorithm that is able to perform simultaneous model selection and elastic net parameter estimation for a given pair of regularisation parameters. At the upper level these regularisation parameters are optimised using a particle swarm optimisation (PSO) algorithm by minimising the leave one out (LOO) mean square error (LOOMSE). The original contributions are firstly to define an elastic net cost function based on orthogonal decomposition, which facilitates the automatic model structure selection process with no need of using a predetermined error tolerance to terminate the forward selection process. Secondly we derived the LOOMSE formula based on the resultant ENOFR models and show that its computational cost is small due to the proposed ENOFR procedure. As a result a fully automated procedure is achieved without resort to any other validation data set for iterative model evaluation. Illustrative examples are included to demonstrate the effectiveness of the new approaches.

Appendix A. The naive elastic net orthogonal forward regression using the modified Gram–Schmidt (MGS) orthogonalisation procedure

The modified Gram–Schmidt orthogonalisation procedure calculates the $\mathbf{A}$ matrix row by row and orthogonalises $\mathbf{\Phi}$ as follows: at the lth stage make the columns $\boldsymbol{\phi}_j$, $l+1 \le j \le n_M$, orthogonal to the lth column and repeat the operation for $1 \le l \le n_M-1$. Specifically, denoting $\boldsymbol{\phi}_j^{(0)}=\boldsymbol{\phi}_j$, $1 \le j \le n_M$, then

$$\left.\begin{array}{l} \mathbf{w}_l=\boldsymbol{\phi}_l^{(l-1)}, \\ a_{l,j}=\mathbf{w}_l^T\boldsymbol{\phi}_j^{(l-1)}/(\mathbf{w}_l^T\mathbf{w}_l), l+1 \le j \le n_M, \\ \boldsymbol{\phi}_j^{(l)}=\boldsymbol{\phi}_j^{(l-1)}-a_{l,j}\mathbf{w}_l, l+1 \le j \le n_M, \end{array}\right\} \quad l=1,2,\dots,n_M-1. \tag{37}$$

The last stage of the procedure is simply $\mathbf{w}_{n_M}=\boldsymbol{\phi}_{n_M}^{(n_M-1)}$. The elements of the naive elastic net estimator for $\mathbf{g}$ are computed by transforming $\mathbf{y}^{(0)}=\mathbf{y}$ in a similar way:

$$\left.\begin{array}{l} g_l^{(LS)}=\mathbf{w}_l^T\mathbf{y}^{(l-1)}/(\mathbf{w}_l^T\mathbf{w}_l), \\ g_l^{(NEN)}=\left(\dfrac{\mathbf{w}_l^T\mathbf{w}_l}{\mathbf{w}_l^T\mathbf{w}_l+\lambda_2}|g_l^{(LS)}|-\dfrac{\lambda_1/2}{\mathbf{w}_l^T\mathbf{w}_l+\lambda_2}\right)_+ \mathrm{sign}(g_l^{(LS)}), \\ \mathbf{y}^{(l)}=\mathbf{y}^{(l-1)}-g_l^{(EN)}\mathbf{w}_l, \end{array}\right\} \quad 1 \le l \le n_M. \tag{38}$$

This orthogonalisation scheme can be used to derive a simple and efficient algorithm for selecting subset models in a forward-regression manner. First define

$$\mathbf{\Phi}^{(l-1)}=[\mathbf{w}_1 \dots \mathbf{w}_{l-1}\boldsymbol{\phi}_l^{(l-1)} \dots \boldsymbol{\phi}_{n_M}^{(l-1)}]. \tag{39}$$

If some of the columns $\boldsymbol{\phi}_l^{(l-1)}, \dots, \boldsymbol{\phi}_{n_M}^{(l-1)}$ in $\mathbf{\Phi}^{(l-1)}$ have been interchanged, this will still be referred to as $\mathbf{\Phi}^{(l-1)}$ for notational convenience. The lth stage of the selection procedure is given as follows.

Step 1: For $l \le j \le n_M$, compute

$$\left.\begin{array}{l} g_l^{(LS,j)}=(\boldsymbol{\phi}_j^{(l-1)})^T\mathbf{y}^{(l-1)} \\ /((\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{\phi}_j^{(l-1)}), \\ g_l^{(NEN,j)}=\left\{\dfrac{(\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{\phi}_j^{(l-1)}}{(\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{\phi}_j^{(l-1)}+\lambda_2}\left|g_l^{(LS,j)}\right| \right. \\ \left. -\dfrac{\lambda_1/2}{(\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{\phi}_j^{(l-1)}+\lambda_2}\right\}_+ \mathrm{sign}(g_l^{(LS,j)}) \\ [\mathrm{eNerr}]_l^{(j)}=((g_l^{(NEN,j)})^2 \times ((\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{\phi}_j^{(l-1)}+\lambda_2))/(\mathbf{y}^T\mathbf{y}). \end{array}\right\}$$

Step 2: Find

$$[\mathrm{eNerr}]_l=[\mathrm{eNerr}]_l^{(j_l)}=\max\{[\mathrm{eNerr}]_l^{(j)}, l \le j \le n_M\}.$$

Then the j_lth column of $\mathbf{\Phi}^{(l-1)}$ is interchanged with the lth column of $\mathbf{\Phi}^{(l-1)}$, the j_lth column of $\mathbf{A}$ is interchanged with the lth column of $\mathbf{A}$ up to the $(l-1)$th row. This effectively selects the j_lth candidate as the lth regressor in the subset model.

Step 3: Perform the orthogonalisation as indicated in (37) to derive the lth row of $\mathbf{A}$ and to transform $\mathbf{\Phi}^{(l-1)}$ into $\mathbf{\Phi}^{(l)}$. Calculate $g_l^{(NEN)}$ and update $\mathbf{y}^{(l-1)}$ into $\mathbf{y}^{(l)}$ in the way shown in (38).

The selection is terminated at the (n_s+1) stage when $[\mathrm{eNerr}]_{n_s+1}=0$ is satisfied and this produces a subset model containing n_s significant regressors. The algorithm described here is in its standard form. A fast implementation can be adopted, as shown in [48], to reduce complexity.

Appendix B. Particle swarm optimisation for choosing regularisation parameters

In the following we propose to apply the PSO algorithm [22,23], and aim to solve

$$\boldsymbol{\lambda}_{\rm opt} = \arg \min_{\boldsymbol{\lambda} \in \prod_{j=1}^{2} \Lambda_j} J(\boldsymbol{\lambda}), \tag{40}$$

where

$$\prod_{j=1}^{2} \Lambda_j = \prod_{j=1}^{2} [0, \Lambda_{j,\max}] \tag{41}$$

defines the search space. Depending on the problem, $\Lambda_{j,\max}$'s are set empirically. For our problem, it is not difficult to coarsely identify some values above which the resultant solutions are definitely not acceptable in terms of model predictive performance.

A swarm of particles, $\{\boldsymbol{\lambda}_i^{(m)}\}_{i=1}^{S}$, that represent potential solutions are "flying" in the search space $\prod_{j=1}^{2} \Lambda_j$, where S is the swarm size and index m denotes the iteration step. The algorithm is summarised as follows.

(a) *Swarm initialisation*: Set the iteration index $m=0$ and randomly generate $\{\boldsymbol{\lambda}_i^{(m)}\}_{i=1}^{S}$ in the search space $\prod_{j=1}^{2} \Lambda_j$.

(b) *Swarm evaluation*: The cost of each particle $\boldsymbol{\lambda}_i^{(m)}$ is obtained as $J(\boldsymbol{\lambda}_i^{(m)})$. Each particle $\boldsymbol{\lambda}_i^{(m)}$ remembers its best position visited so far, denoted as $\mathbf{pb}_i^{(m)}$, which provides the cognitive information. Every particle also knows the best position visited so far among the entire swarm, denoted as $\mathbf{gb}^{(m)}$, which provides the social information. The cognitive information $\{\mathbf{pb}_i^{(m)}\}_{i=1}^{S}$ and the social information $\mathbf{gb}^{(m)}$ are updated at each iteration:

```
For $(i=1;\ i \le S;\ i++)$
    If $(J(\boldsymbol{\lambda}_i^{(m)}) < J(\mathbf{pb}_i^{(m)}))$ $\mathbf{pb}_i^{(m)} = \boldsymbol{\lambda}_i^{(m)}$;
End for;
$i^* = \arg\min_{1 \le i \le S} J(\mathbf{pb}_i^{(m)})$;
If $(J(\mathbf{pb}_{i^*}^{(m)}) < J(\mathbf{gb}^{(m)}))$ $\mathbf{gb}^{(m)} = \mathbf{pb}_{i^*}^{(m)}$;
```

(c) *Swarm update*: Each particle $\boldsymbol{\lambda}_i^{(m)}$ has a velocity, denoted as $\boldsymbol{\gamma}_i^{(m)}$, to direct its "flying". The velocity and position of the ith particle are updated in each iteration according to

$$\boldsymbol{\gamma}_i^{(m+1)} = \mu_0 * \boldsymbol{\gamma}_i^{(m)} + rand() * \mu_1 * (\mathbf{pb}_i^{(m)} - \boldsymbol{\lambda}_i^{(m)}) + rand() * \mu_2 * (\mathbf{gb}^{(m)} - \boldsymbol{\lambda}_i^{(m)}), \tag{42}$$

$$\boldsymbol{\lambda}_i^{(m+1)} = \boldsymbol{\lambda}_i^{(m)} + \boldsymbol{\gamma}_i^{(m+1)}, \tag{43}$$

where μ_0 is the inertia weight, μ_1 and μ_2 are the two acceleration coefficients. $rand()$ denotes the uniform random number between 0 and 1. In order to avoid excessive roaming of particles beyond the search space [27], a velocity space

$$\prod_{j=1}^{2} \Upsilon_j = \prod_{j=1}^{2} [-\Upsilon_{j,\max}, \Upsilon_{j,\max}] \tag{44}$$

is imposed on $\boldsymbol{\gamma}_i^{(m+1)}$ so that

$$\text{If } (\boldsymbol{\gamma}_i^{(m+1)}|_j > \Upsilon_{j,\max})\ \boldsymbol{\gamma}_i^{(m+1)}|_j = \Upsilon_{j,\max};$$
$$\text{If } (\boldsymbol{\gamma}_i^{(m+1)}|_j < -\Upsilon_{j,\max})\ \boldsymbol{\gamma}_i^{(m+1)}|_j = -\Upsilon_{j,\max};$$

where $\boldsymbol{\gamma}|_j$ denotes the jth element of $\boldsymbol{\gamma}$. Moreover, if the velocity as given in Eq. (42) approaches zero, it is reinitialised proportional to $\Upsilon_{j,\max}$ with a small factor ν

$$\text{If}(\boldsymbol{\gamma}_i^{(m+1)}|_j == 0)\boldsymbol{\gamma}_i^{(m+1)}|_j = \pm\, rand() * \nu * \Upsilon_{j,\max}; \tag{45}$$

(d) *Termination condition check*: If the maximum number of iterations, $I_{\max}$, is reached, terminate the algorithm with the solution $\mathbf{gb}^{(I_{\max})}$; otherwise, set $m = m+1$ and go to Step (b).

Ratnaweera and co-authors [25] reported that using a time varying acceleration coefficient (TVAC) enhances the performance of PSO. We adopt this mechanism, in which μ_1 is reduced from 2.5 to 0.5 and μ_2 varies from 0.5 to 2.5 during the iterative procedure:

$$\mu_1 = (0.5 - 2.5) * m / I_{\max} + 2.5,$$
$$\mu_2 = (2.5 - 0.5) * m / I_{\max} + 0.5. \tag{46}$$

The reason for good performance of this TVAC mechanism can be explained as follows. At the initial stages, a large cognitive component and a small social component help particles to wander around or better exploit the search space, avoiding local minima. In the later stages, a small cognitive component and a large social component help particles to converge quickly to a global minimum. We use $\mu_0 = rand()$ at each iteration.

The search space as given in Eq. (41) is defined by the specific problem to be solved, and the velocity limit $\Upsilon_{j,\max}$ is empirically set. An appropriate value of the small control factor ν in Eq. (45) for avoiding zero velocity is empirically found to be $\nu = 0.1$ for our application.

References

[1] C.J. Harris, X. Hong, Q. Gan, Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach, Springer-Verlag, Heidelberg, Germany, 2002.

[2] M. Brown, C.J. Harris, Neurofuzzy Adaptive Modelling and Control, Prentice Hall, Hemel Hempstead, 1994.

[3] A.E. Ruano, Intelligent Control Systems using Computational Intelligence Techniques, IEE Publishing, Stevenage, UK, 2005.

[4] R. Murray-Smith, T.A. Johansen, Multiple Model Approaches to Modelling and Control, Taylor and Francis, UK, 1997.

[5] S.G. Fabri, V. Kadirkamanathan, Functional Adaptive Control: An Intelligent Systems Approach, Springer, London, 2001.

[6] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their applications to non-linear system identification, Int. J. Control 50 (1989) 1873–1896.

[7] M.J. Korenberg, Identifying nonlinear difference equation and functional expansion representations: the fast orthogonal algorithm, Ann. Biomed. Eng. 16 (1988) 123–142.

[8] L. Wang, J.M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, IEEE Trans. Neural Netw. 5 (1992) 807–814.

[9] X. Hong, C.J. Harris, Neurofuzzy design and model construction of nonlinear dynamical processes from data, IEE Proc. Control Theory Appl. 148 (6) (2001) 530–538.

[10] Q. Zhang, Using wavelets network in nonparametric estimation, IEEE Trans. Neural Netw. 8 (2) (1993) 1997.

[11] S.A. Billings, H.L. Wei, The wavelet-narmax representation: a hybrid model structure combining polynomial models with multiresolution wavelet decompositions, Int. J. Syst. Sci. 36 (3) (2005) 137–152.

[12] N. Chiras, C. Evans, D. Rees, Nonlinear gas turbine modelling using narmax structures, IEEE Trans. Instrum. Meas. 50 (4) (2001) 893–898.

[13] Y. Gao, M.J. Er, Online adaptive fuzzy neural identification and control of a class of MIMO nonliear systems, IEEE Trans. Fuzzy Syst. 11 (4) (2003) 462–477.

[14] K.M. Tsang, W.L. Chan, Adaptive control of power factor correction converter using nonlinear system identification, IEE Proc. Electr. Power Appl. 152 (3) (2005) 627–633.

[15] G.C. Luh, W.C. Cheng, Identification of immune models for fault detection, Proc. Instn. Mech. Eng. Part I: J. Syst. Control Eng. 218 (2004) 353–367.

[16] M. Stone, Cross validatory choice and assessment of statistical predictions, J. R. Stat. Soc. Ser. B 36 (1974) 117–147.

[17] S. Chen, Y. Wu, B.L. Luk, Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks, IEEE Trans. Neural Netw. 10 (1999) 1239–1243.

[18] M.J.L. Orr, Regularisation in the selection of radial basis function centers, Neural Comput. 7 (3) (1995) 954–975.

[19] X. Hong, S.A. Billings, Parameter estimation based on stacked regression and evolutionary algorithms, IEE Proc. - Control Theory Appl. 146 (5) (1998) 406–414.

[20] L. Ljung, T. Glad, Modelling of Dynamic Systems, Prentice Hall, Englewood Cliffs, NJ, 1994.

[21] S. Chen, X. Hong, C.J. Harris, Particle swarm optimization aided orthogonal forward regression for unified data modelling, IEEE Trans. Evol. Comput. 14 (4) (2010) 477–499.

[22] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of 1995 IEEE International Conference on Neural Networks, Perth, Australia, November 27–December 1, vol. 4, 1995, pp. 1942–1948.
[23] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, San Franscisco, USA, 2001.
[24] D.W. van der Merwe, A.P. Engelbrecht, Data clustering using particle swarm optimization, in: Proceedings of CEC 2003, Cabberra, Australia, December 8–12, 2003, pp. 215–220.
[25] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (June) (2004) 240–255.
[26] M.G.H. Omran, Particle swarm optimization methods for pattern recognition and image processing (Ph.D. thesis), University of Pretoria, Pretoria, South Africa, 2005.
[27] S.M. Guru, S.K. Halgamuge, S. Fernando, Particle swarm optimisers for cluster formation in wireless sensor networks, in: Proceedings of the 2005 International Conference Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, Australia, December 5–8, 2005, pp. 319–324.
[28] K.K. Soo, Y.M. Siu, W.S. Chan, L. Yang, R.S. Chen, Particle-swarm-optimization-based multiuser detector for CDMA communications, IEEE Trans. Veh. Technol. 56 (September) (2007) 3006–3013.
[29] S. Chen, X. Hong, C.J. Harris, Sparse kernel regression modelling using combined locally regularised orthogonal least squares and D-optimality experimental design, IEEE Trans. Autom. Control 48 (6) (2003) 1029–1036.
[30] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stast. Soc. B 67 (2) (2005) 301–320.
[31] S. Chen, Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models, in: Proceedings of 6th International Conference on Signal Processing, Beijing, China, 2002, pp. 1229–1232.
[32] D.J.C. MacKay, Bayesian methods for adaptive models (Ph.D. thesis), California Institute of Technology, USA, 1991.
[33] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1) (1998) 33–61.
[34] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B 58 (1) (1996) 267–288.
[35] B. Efron, I. Johnstone, T. Hastie, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2004) 407–451.
[36] S. Chen, Local regularization assisted orthogonal least squares regression, Neurocomputing 69 (4–6) (2006) 559–585.
[37] S. Chen, S.A. Billings, Representation of nonlinear systems: the NARMAX model, Int. J. Control 49 (3) (1989) 1013–1032.
[38] S. Chen, E.S. Chng, K. Alkadhimi, Regularized orthogonal least squares algorithm for constructing radial basis function networks, Int. J. Control 64 (5) (1996) 829–837.
[39] L. Ljung, System Identification: Theory for the User, Prentice Hall, New Jersey, 1987.
[40] J. Shao, Linear model selection by cross validation, J. Am. Stat. Assoc. 88 (422) (1993) 486–494.
[41] I. Rivals, L. Personnaz, On cross validation for model selection, Neural Comput. 11 (4) (1999) 863–870.
[42] A.M. Molinaro, R. Simon, R.M. Pfeiffer, Prediction error estimation: a comparison of resampling methods, Bioinformatics 21 (15) (2005) 3301–3307.
[43] R.H. Myers, Classical and Modern Regression with Applications, 2nd edn., PWS-KENT, Boston, 1990.
[44] T. Stamey, J. Kabalin, J. McNeal, I. Johnsdtone, F. Freiha, E. Redwine, N. Yang, Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate ii: radical prostatectomy treated patients, J. Urol. 16 (1989) 1076–1083.
[45] S.A. Billings, S. Chen, R.J. Backhouse, The identification of linear and nonlinear models of a turbocharged automative diesel engine, Mech. Syst. Signal Process. 3 (2) (1989) 123–142.
[46] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modelling using orthogonal forward regression with PRESS statistic and regularization, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 34 (2) (2004) 898–911.
[47] G.E.P. Box, G.M. Jenkins, Time Series Analysis, Forecasting and Control, Holden-Day Inc, San Franscisco, USA, 1976.
[48] S. Chen, J. Wigger, Fast orthogonal least squares algorithm for efficient subset selection, IEEE Trans. Signal Process. 43 (7) (1995) 1713–1715.

Xia Hong received her university education at National University of Defense Technology, P. R. China (BSc, 1984, MSc, 1987), and University of Sheffield, UK (PhD, 1998), all in automatic control. She worked as a research assistant in Beijing Institute of Systems Engineering, Beijing, China from 1987 to 1993. She worked as a research fellow in the Department of Electronics and Computer Science at University of Southampton from 1997 to 2001. She is currently a Professor at School of Systems Engineering, University of Reading. She is actively engaged in research into nonlinear systems identification, data modelling, estimation and intelligent control, neural networks, pattern recognition, learning theory and their applications. She has published over 140 research papers, and coauthored a research book. She was awarded a Donald Julius Groen Prize by IMechE in 1999.

Sheng Chen received his BEng degree from the East China Petroleum Institute, China, in January 1982, and his PhD degree from the City University, London, in September 1986, both in control engineering. In 2005, he was awarded the higher doctorate degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK.

From 1986 to 1999, He held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with Electronics and Computer Science, the University of Southampton, UK, where he currently holds the post of Professor in Intelligent Systems and Signal Processing. Dr Chen's research interests include adaptive signal processing, wireless communications, modelling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods and optimisation. He has published over 500 research papers.

Dr. Chen is a Fellow of IEEE and a Fellow of IET. He is a Distinguished Adjunct Professor at the King Abdulaziz University, Jeddah, Saudi Arabia. He is an ISI highly cited researcher in the engineering category (March 2004).