Neurocomputing 122 (2013) 210–220

Contents lists available at ScienceDirect

Neurocomputing

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

Particle swarm optimisation assisted classification using elastic net prefiltering

Xia Hong [a,*], Junbin Gao [b], Sheng Chen [c,d], Chris J. Harris [c]

[a] School of Systems Engineering, University of Reading, Reading RG6 6AY, UK
[b] School of Computing and Mathematics, Charles Sturt University, Australia
[c] Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
[d] Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 11 February 2013
Received in revised form
14 May 2013
Accepted 6 June 2013
Communicated by K. Li
Available online 2 July 2013

Keywords:
Classification
Bayesian evidence
Elastic net
Forward regression
Regularisation
Particle swarm optimisation

ABSTRACT

A novel two-stage construction algorithm for linear-in-the-parameters classifier is proposed, aiming at noisy two-class classification problems. The purpose of the first stage is to produce a prefiltered signal that is used as the desired output for the second stage to construct a sparse linear-in-the-parameters classifier. For the first stage learning of generating the prefiltered signal, a two-level algorithm is introduced to maximise the model's generalisation capability, in which an elastic net model identification algorithm using singular value decomposition is employed at the lower level while the two regularisation parameters are selected by maximising the Bayesian evidence using a particle swarm optimization algorithm. Analysis is provided to demonstrate how "Occam's razor" is embodied in this approach. The second stage of sparse classifier construction is based on an orthogonal forward regression with the D-optimality algorithm. Extensive experimental results demonstrate that the proposed approach is effective and yields competitive results for noisy data sets.

1. Introduction

A basic principle in constructing mathematical models from data is "Occam's razor", as in many data modelling problems, such as regression and pattern classification, the aim is to find the smallest possible models with the capability to approximate system output for unseen new input data. It is known that Occam's razor is naturally embodied by two important modelling approaches of cross validation (CV) [1,2] and evidence maximisation with a Gaussian prior [3]. Models are identified according to some objective criteria which manifest in the Bayesian approach by two levels of inference. At the first level of inference, the model parameters are inferred by maximising the a posterior probability (MAP) of the model parameters, and at the second level of inference, models are ranked by evidence, i.e. the marginal probability for the given hypothesis or model [3].

Modelling techniques based on model construction or selection have been widely studied, e.g. support vector machine (SVM), relevance vector machines (RVM), and orthogonal forward regression (OFR) [4–7]. The orthogonal least square (OLS) algorithm [8] was developed as a practical construction algorithm for linear-in-the-parameters model, which include a large class of non-linear model representations, such as radial basis functions (RBF) networks and SVM. Using the class labels as the desired output for training, a two-class classification problem can be configured into a regression framework that solves a separating hyperplane for two classes. The orthogonal forward selection (OFS) procedure of [8] can then be applied to construct parsimonious two-class classifiers incrementally by maximising the Fisher ratio of class separability measure [9,10] or by minimising the misclassification rate [11].

The l^2-norm regularisation assisted OLS (ROLS) approaches have been proposed based on minimising the leave-one-out criteria for regression, classification and probability density estimation [12]. The l^2-norm regularisation techniques are developed to carry out parameter estimation and model structure selection simultaneously [5,13–16]. It has been shown that l^2 norm parameter regularisation is equivalent to adopting a Gaussian prior for the model parameters from Bayesian viewpoint [3,16]. Therefore, from the powerful Bayesian learning perspective, the l^2 norm 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

* Corresponding author. Tel.: +44 118 378 8222.
E-mail addresses: x.hong@reading.ac.uk. (X. Hong), jbgao@cse.edu.au. (J. Gao), sqc@ecs.soton.ac.uk. (S. Chen), cjh@ecs.soton.ac.uk. (C.J. Harris)

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.06.030

parameters [3,16]. Note that fundamentally l^2-norm regularisation methods can only drive many model parameters to small but non-zero values.

Alternatively, the model sparsity can be achieved by minimising the l^1 norm of the parameters, which is the fundamental approach adopted in the basis pursuit or least absolute shrinkage and selection operator (LASSO) [17,18]. The least angle regression (LAR) procedure [19] is developed for solving the l^1-norm regularisation problem efficiently. The Bayesian interpretation for the l^1-norm regularisation is simply by adopting a Laplacian prior for the parameters. The advantage of the LASSO is that it can achieve much sparser models by forcing many parameters to exactly zero, rather than small, but non-zero, parameter values derived from the minimisation of the l^p norm, where $p>1$. Unfortunately introducing non-differentiable l^1 norm in the cost function leads to the difficulties of model parameter estimation and finding an appropriate l^1 regularizer. Another disadvantage of adopting the l^1-norm optimisation is that a group of correlated model terms cannot be selected together, which is not desirable since intuitively if a particular model term is selected, other correlated model terms should also be included for the sake of model interoperability. The OLS algorithm of [8] has been combined with the l^1-norm regularisation for constructing sparse regression models [20].

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 [21]. The EN keeps the model sparsity of the LASSO [18], while strongly correlated model terms tend to be selected or not selected together. It is shown that the EN problem can be transformed into an equivalent LASSO problem on the augmented data, from which the LAR procedure is applied, which is referred to as the LARS-EN in [21]. Similarly, there is a Bayesian connection to the EN [21,22]. In the work [22], the authors proposed a Bayesian method based on Gibbs sampler to calculate two regularisation parameters simultaneously. Note that there exists a dilemma in the modelling of unknown systems using Bayesian approach. The priors, which are subjective by nature, should be allowed to be flexible, in terms of their functional form, but the problems of evidence maximisation for non-Gaussian priors are generally difficult to compute. It is therefore highly desirable to develop computational methods to tackle the tractability issues, such as the computation of Bayesian evidence based on different priors. Furthermore, since there are two regularisation parameters in the EN, the cross validation has to be performed over a two-dimensional space. Assume that the ten-fold cross validation is used in choosing the two regularisation parameters based on a grid search, as is typically adopted in practice. Then, for each setting of the l^2 norm regularisation parameter over a grid of the l^2 norm regularisation parameter values, the LARS-EN algorithm produces the entire solution path of the EN, which is used to select the l^1 norm regularisation parameter by ten-fold cross validation. Clearly, this may not yield the optimal regularisation 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 to an unacceptably high level.

Against this background, in this paper we propose a novel two-stage construction algorithm for two-class linear-in-the-parameters classifier in order to avoid overfitting to the noise in the training data set as well as to enhance the classifier's generalisation capability. The basic idea is that a sparse classifier is constructed using a prefiltered signal, rather than the original class label vector, as the desired output. The Bayesian EN regularisation is applied to produce the prefiltered signal in the first stage, in which a two-level algorithm is introduced, aiming to maximise the model's generalisation capability by the Bayesian EN approach. Specifically, at the lower level, a new EN model identification algorithm is employed based on the significant eigenvectors of the regression matrix, while the two regularisation parameters are optimised at the upper level using a particle swarm optimisation (PSO) algorithm [23,24] to maximise the Bayesian evidence using the prefiltered signal. It is shown that due to the orthogonality Bayesian evidence can be computed with ease, and the resultant formula also leads to insights on the basic principle of Occam's razor. Furthermore, the PSO aided optimisation is much more efficient than the traditional grid search for optimising the two associated regularisation parameters. The second stage of sparse classifier construction is based on the OFR with D-optimality algorithm [7], which the tested efficiency and simplicity in sparse model construction.

This paper is organised as follows. Section 2 formulates the proposed novel two-stage construction algorithm for selecting sparse two-class classifiers, which are robust to the noise in the training data and have excellent generalisation performance. In Section 3, we obtain the Bayesian evidence formula based on the resultant prefiltered signal, and then present the PSO algorithm to optimise the two EN regularisation parameters. Critical mathe-
matical analysis is provided to interpret the relationship between the Bayesian evidence and the Occam's razor. In Section 4, the experimental results are employed to demonstrate the effectiveness of the proposed approach, leading to a discussion on the merits of this novel two-stage algorithm. Finally, our conclusions are given in Section 5.

2. Two stage classifier construction using elastic net prefiltering

In this section, we first briefly outline the concept of linear-in-the-parameters classifier, and then introduce the proposed two stage procedure for constructing sparse classifiers as depicted in Fig. 1. More specifically, the proposed classifier construction procedure includes the stage one of initial generation of the prefiltered signal, based on singular value decomposition (SVD), using the PSO aided EN regularisation parameters optimization, followed by the stage two of a two-class sparse classifier selection using the OFR with D-optimality algorithm of [7].

2.1. Linear-in-the-parameters classifier

Consider an approximately balanced two-class training data set $D_N=\{\boldsymbol{x}(k),y(k)\}_{k=1}^N$, in which $y(k)\in\{1,-1\}$ denotes the class type for the feature vector $\boldsymbol{x}(k)\in\mathbb{R}^n$. Let a linear-in-the-parameters classifier $f(\boldsymbol{x}):\mathbb{R}^n\rightarrow\{1,-1\}$ be formed using the data set D_N, given

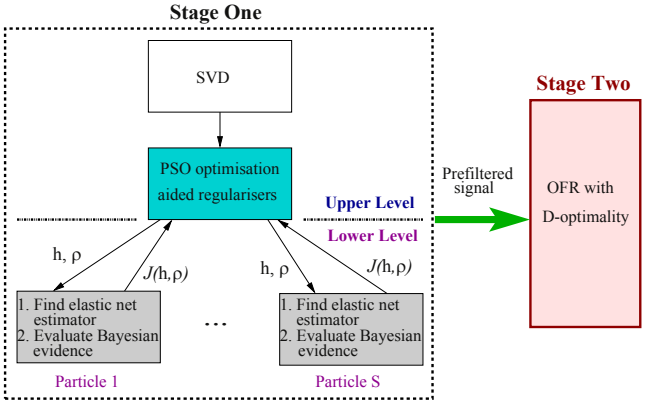


Fig. 1. Schematic diagram of the proposed two stage classifier construction using elastic net prefiltering.

by

$$\hat{y}(k) = \mathrm{sgn}(f(\boldsymbol{x}(k))) \quad \text{with } f(\boldsymbol{x}(k)) = \sum_{i=1}^{L} \theta_i \phi_i(\boldsymbol{x}(k)), \tag{1}$$

where

$$\mathrm{sgn}(s) = \begin{cases} 1 & \text{if } s \geq 0, \\ -1 & \text{if } s < 0, \end{cases} \tag{2}$$

L is the number of regressors or kernels, $\phi_i(\bullet)$ denote the classifier's kernels with a known non-linear basis function, such as radial basis function (RBF), and θ_i are the model parameters, while $\hat{y}(k)$ denotes the predicted class label for $\boldsymbol{x}(k)$. The error between the true class label and the classifier's output signal is given by $e(k) = y(k) - f(\boldsymbol{x}(k))$, which can be written in the matrix form

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{e} \tag{3}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}, \quad \boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \ \boldsymbol{\phi}_2 \ \cdots \ \boldsymbol{\phi}_L],$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_L \end{bmatrix}, \quad \boldsymbol{e} = \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(N) \end{bmatrix}, \tag{4}$$

and the regressor columns are $\boldsymbol{\phi}_i = [\phi_i(\boldsymbol{x}(1)) \ \phi_i(\boldsymbol{x}(2)) \ \cdots \ \phi_i(\boldsymbol{x}(N))]^{\mathrm{T}}$, for $1 \leq i \leq L$.

Geometrically, the hyperplane defined by

$$\sum_{i=1}^{L} \theta_i \phi_i(\boldsymbol{x}) = 0 \tag{5}$$

divides the data into two classes.

2.2. Prefiltering using SVD based elastic net regularisation

The aim of prefiltering is to "filter out" the noise in the training data and, therefore, to define a robust classification boundary over the training data set which can be used as the target or desired output for classifier construction. Consider the SVD of the regression matrix $\boldsymbol{\Phi}$ given by $\boldsymbol{\Phi} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$, where the diagonal matrix $\boldsymbol{S} = \mathrm{diag}\{s_1, s_2, \ldots, s_{n_s}, 0, \ldots, 0\} \in \mathbb{R}^{L \times L}$ with $s_1 \geq s_2 \geq \cdots \geq s_{n_s} > 0$ denoting the resultant n_s non-zero singular values, while $\boldsymbol{U} = [\boldsymbol{u}_1 \boldsymbol{u}_2 \cdots \boldsymbol{u}_L] \in \mathbb{R}^{N \times L}$ and $\boldsymbol{V} = [\boldsymbol{v}_1 \boldsymbol{v}_2 \ldots \boldsymbol{v}_L] \in \mathbb{R}^{L \times L}$ containing the orthogonal columns that satisfy $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = \boldsymbol{I}_L$ and $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{I}_L$, respectively, in which $\boldsymbol{I}_L$ denotes L-dimensional identity matrix. With this SVD of $\boldsymbol{\Phi}$, the regression model (3) can alternatively be expressed as

$$\boldsymbol{y} = \boldsymbol{U}_r \boldsymbol{g} + \boldsymbol{e}, \tag{6}$$

where $\boldsymbol{U}_r = [\boldsymbol{u}_1 \ \boldsymbol{u}_2 \ \ldots \ \boldsymbol{u}_{n_s}] \in \mathbb{R}^{N \times n_s}$ and $\boldsymbol{g} = [g_1 \ g_2 \ \ldots \ g_{n_s}]^{\mathrm{T}}$. Clearly, $\boldsymbol{U}_r^{\mathrm{T}}\boldsymbol{U}_r = \boldsymbol{I}_{n_s}$. Thus, the SVD maps the original L-dimensional space spanned by $\boldsymbol{\Phi}$ onto a lower dimensional space spanned by $\boldsymbol{U}_r$ which represents the true dimension in the sense that $\boldsymbol{\Phi} = \sum_{i=1}^{n_s} s_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}}$.

We note that solving (5) by minimising $\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2$ is an ill-posed problem. Thus, some structural regularisation is needed to emphasize the smoothness of the decision boundary in order to avoid overfitting to the noise. For example, for some appropriately chosen fixed positive λ_1 and λ_2, the naive elastic net (NEN) criterion is defined as [21]

$$L(\lambda_1, \lambda_2, \boldsymbol{\theta}) = \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 + \lambda_2 \|\boldsymbol{\theta}\|^2 + \lambda_1 \|\boldsymbol{\theta}\|_1, \tag{7}$$

where $\|\bullet\|$ denotes the Euclidean norm while $\|\boldsymbol{\theta}\|_1 = \sum_{i=1}^{L} |\theta_i|$ is the l^1 norm of $\boldsymbol{\theta}$. The NEN estimator is the minimiser of

$$\boldsymbol{\theta}^{(NEN)} = \arg\min_{\boldsymbol{\theta}} \{L(\lambda_1, \lambda_2, \boldsymbol{\theta})\}. \tag{8}$$

This can be transformed into an equivalent LASSO problem on the augmented data, and solved by the LARS-EN [21]. The EN has some desirable properties, as it maintains the model sparsity of the LASSO, but is not as aggressive as the LASSO in excluding correlated terms in the model. This is because these terms tend to be either selected or not selected in the model together, as a consequence the l^2 norm regularisation [21]. Note that there is no analytical solution to (8) unless the model terms are orthogonal.

Based on the NEN criterion of (7), in this paper, we propose to apply the following SVD based elastic net criterion:

$$L_e(\lambda_1, \lambda_2, \boldsymbol{g}) = \|\boldsymbol{y} - \boldsymbol{U}_r \boldsymbol{g}\|^2 + \lambda_2 \|\boldsymbol{g}\|^2 + \lambda_1 \|\boldsymbol{g}\|_1. \tag{9}$$

Since the model terms contains in $\boldsymbol{U}_r$ are orthogonal, an analytical solution can be derived by minimising $L_e(\lambda_1, \lambda_2, \boldsymbol{g})$. In fact, the NEN solution for $\boldsymbol{g}$ is obtained by setting the subderivative [25] of L_e with respect to $\boldsymbol{g}$ to zero, namely, $(\partial L_e / \partial \boldsymbol{g}) = \boldsymbol{0}$, which yields

$$\boldsymbol{U}_r^{\mathrm{T}}\boldsymbol{y} - \frac{\lambda_1}{2} \mathrm{sign}(\boldsymbol{g}) = (1 + \lambda_2)\boldsymbol{g}, \tag{10}$$

where $\mathrm{sign}(\boldsymbol{g}) = [\mathrm{sign}(g_1) \ \mathrm{sign}(g_2) \ \ldots \ \mathrm{sign}(g_{n_s})]^{\mathrm{T}}$ with

$$\begin{cases} \mathrm{sign}(s) = 1 & \text{if } s > 0, \\ \mathrm{sign}(s) = -1 & \text{if } s < 0, \\ \mathrm{sign}(s) \in [-1, 1] & \text{if } s = 0. \end{cases} \tag{11}$$

The solution of (10) is readily given by

$$g_i^{(NEN)} = \left(\frac{1}{1 + \lambda_2} \left| g_i^{(LS)} \right| - \frac{\lambda_1 / 2}{1 + \lambda_2} \right)_+ \mathrm{sign}(g_i^{(LS)}), \tag{12}$$

where $g_i^{(LS)} = \boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{y}$, $1 \leq i \leq n_s$, are the usual least squares (LS) estimates of g_i, and

$$z_+ = \begin{cases} z & \text{if } z > 0, \\ 0 & \text{if } z \leq 0. \end{cases} \tag{13}$$

Note that the cost function (9) contains a sparsity inducing l^1 norm so that some parameters $g_i^{(NEN)}$ will be zeros, producing a sparse model containing only $n_m \ll n_s$ significant singular vectors. Let $\boldsymbol{g}^{(NEN)} = [\tilde{g}_1^{(NEN)} \ \tilde{g}_2^{(NEN)} \ \ldots \ \tilde{g}_{n_m}^{(NEN)}]^{\mathrm{T}} \in \mathbb{R}^{n_m}$, consisting of all the non-zeros parameters, and denote the sub-matrix of $\mathbf{U}_r$, which consists of the columns corresponding to the non-zeros parameters, by $\boldsymbol{U}_s = [\tilde{\boldsymbol{u}}_1 \ \tilde{\boldsymbol{u}}_2 \ \ldots \ \tilde{\boldsymbol{u}}_{n_m}] \in \mathbb{R}^{N \times n_m}$. We can construct a prefiltered signal using

$$\boldsymbol{y}_{pre} = [y_{pre}(1) y_{pre}(2), \ldots, y_{pre}(N)]^{\mathrm{T}} = \boldsymbol{U}_s \boldsymbol{g}^{(NEN)}. \tag{14}$$

Notice that the dimension is further reduced in the latent space by eliminating any term with $|g_i^{(LS)}|$ less than the threshold $\lambda_1/2$. We further notice that $|g_i^{(LS)}|$, which is subject to the noise in the estimation data, directly measures the correlation between each singular vector and the noisy system output. This means that if λ_1 is appropriately chosen according to the noise level, we can significantly reduce the error propagation into the prefiltered signal $\boldsymbol{y}_{pre}$ from the noisy training data via model parameters, and thus produces a smoother decision boundary.

Instead of thresholding by λ_1, the effect of λ_2 scales down parameters by multiplication, which offers another degree of freedom in controlling the parameter estimation variance. In the original EN procedure [21], a double shrinkage problem has been observed, and in order to mitigate this problem a rescaling step is applied to the NEN solution obtained by (8). Clearly, in the case of minimising the cost function (7) where the model bases are

correlated, the resultant sparse model terms are dependent on the value of λ_2, and hence the effect of λ_2 lies in the terms selected into the model as well as the associated model parameters. However, as our proposed criterion (9) is defined on an orthogonal space, and this means that the rescaling step, if is applied, is equivalent to setting $\lambda_2 = 0$. In order to have more flexibility in regularisation control, we opt to use the NEN solution directly, which includes $\lambda_1 = 0$ or $\lambda_2 = 0$ as special cases.

Obviously, it is critically important to choose appropriate values for the two regularisation parameters λ_1 and λ_2. The conventional grid search based on cross validation suffers from some serious drawbacks, as discussed previously in the Introduction section. We propose a PSO procedure to efficiently optimise the two regularisation parameters based on a novel Bayesian analysis that is different from any existent approach. This will be detailed in Section 3.

2.3. Sparse classifier construction using OFR with D-optimality

The aim of the sparse classifier construction stage is to identify a linear-in-the-parameters classifier as described in Section 2.1, with excellent generalisation capability and simultaneously a sparse representation containing only a small number of kernels. The advantages of parsimonious models are that they are computationally more efficient when applying to the new data and easier to interpret in physical applications. Although $\boldsymbol{y}_{pre}$ obtained in (14) defines a classification boundary in the latent space via SVD and can be used to generate predicted labels over the training data set, we note that it cannot be directly used as a classifier for unseen data samples. Nor does it lead to a sparse kernel classifier, because each singular vector $\boldsymbol{u}_i$ is a linear combination of all the kernels $\phi_i(\boldsymbol{x}(k))$.

However, the prefiltered signal $y_{pre}(k)$, $1 \le k \le N$, can be used as the desired output to construct a kernel classifier $f(\boldsymbol{x}(k))$ in a regression framework, where the OFR with D-optimality algorithm [7] can readily be applied to automatically select a sparse classifier with excellent generalisation properties. The advantage of using the prefiltered signal, instead of the original training labels, as the desired output is that the noise in the original training data has been filtered out and this substantially reduces the adverse efforts of the noise in sparse classifier construction. The D-optimality is a model structure robustness criterion used in experimental design to tackle ill-conditioning in model structure and to minimise estimation variance [26]. The OFR with D-optimality algorithm is an efficient forward regression method incorporating structure selection and parameter estimation simultaneously [7,16].

Formally, we can write a regression model linking $y_{pre}(k)$ and $f(\boldsymbol{x}(k))$ as

$$y_{pre}(k) = f(\boldsymbol{x}(k)) + \varepsilon(\boldsymbol{x}(k)) = \sum_{i=1}^{L} \theta_i \phi_i(\boldsymbol{x}(k)) + \varepsilon(k), \tag{15}$$

where $\varepsilon(k)$ is defined as the modelling error at $\boldsymbol{x}(k)$ between the proposed two-class kernel classifier and the prefiltered signal. Since the target $y_{pre}(k)$ is free of noise, $E[\varepsilon^2(k))]$ is expected to be just the approximation error and much smaller than $E[y_{pre}^2(k)]$. It can then be assumed that the classification performance of the final optimal sparse model classifier $f(\boldsymbol{x}(k))$ is close to that of the prefiltered signal $y_{pre}(k)$. For example, unless

$$|\varepsilon(\boldsymbol{x}(k))| > |f(\boldsymbol{x}(k))| \quad \text{and} \quad \mathrm{sgn}(\varepsilon(\boldsymbol{x}(k))) \neq \mathrm{sgn}(f(\boldsymbol{x}(k))),$$

which is most unlikely, the predicted class label based on the sparse classifier $f(\boldsymbol{x}(k))$ should be the same as that of $y_{pre}(k)$.

By denoting $\boldsymbol{\epsilon} = [\varepsilon(1)\ \varepsilon(2)\ \ldots\ \varepsilon(N)]^{\mathrm{T}}$, the regression model (15) can be written in the matrix form

$$\boldsymbol{y}_{pre} = \boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\epsilon}. \tag{16}$$

Let the orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}$ be given by

$$\boldsymbol{\Phi} = \boldsymbol{W}\boldsymbol{A}, \tag{17}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,L} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{L-1,L} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{18}$$

and

$$\boldsymbol{W} = [\boldsymbol{w}_1\ \ \boldsymbol{w}_2\ \ \cdots\ \ \boldsymbol{w}_L] \tag{19}$$

with columns satisfying $\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_j = 0$, if $i \neq j$. The regression model (16) can alternatively be expressed as

$$\boldsymbol{y}_{pre} = \boldsymbol{W}\boldsymbol{\gamma} + \boldsymbol{\epsilon}, \tag{20}$$

where the weight vector $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2\ \ldots\ \gamma_L]^{\mathrm{T}}$ is easily estimated by minimising the LS criterion $J(\boldsymbol{\gamma}) = \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon}$. Then the original model parameter vector $\boldsymbol{\theta}$ can be determined using $\boldsymbol{A}\boldsymbol{\theta} = \boldsymbol{\gamma}$ by backward substitution.

The OFR with D-optimality algorithm selects model terms one at a time with the final model consisting of $\overline{n}_s \ll L$ columns of $\boldsymbol{\Phi}$, which is denoted as $\boldsymbol{\Phi}_s$. The D-optimality [7] is defined as $\max \det[\boldsymbol{\Phi}_s^{\mathrm{T}}\boldsymbol{\Phi}_s]$. Since $\det[\boldsymbol{\Phi}_s^{\mathrm{T}}\boldsymbol{\Phi}_s] = \det[\boldsymbol{W}_s^{\mathrm{T}}\boldsymbol{W}_s] = \prod_{l=1}^{\overline{n}_s} \boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{w}_l$, where $\boldsymbol{W}_s$ is the orthogonal matrix corresponding to $\boldsymbol{\Phi}_s$, the combined error reduction ratio (CERR) defined as

$$[\mathrm{cerr}]_l = (\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{w}_l\gamma_l^2 + \beta \log(\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{w}_l))/\boldsymbol{y}_{pre}^{\mathrm{T}}\boldsymbol{y}_{pre} \tag{21}$$

is used to select the lth model term at the lth forward selection stage. Note that this CERR is aimed at maximising the reduction in modelling error (the term $\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{w}_l\gamma_l^2$) and the D-optimality (the term $\log(\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{w}_l)$) simultaneously, where β is a fixed small positive weighting for the D-optimality cost. The first part is related to the training performance, while the second part is related to the generalisation performance [7,16].

Since the D-optimality naturally penalises overparameterisation, the modelling process can automatically terminates so as to achieve a sparse model by setting appropriately β as a predetermined very small number. More specifically, at some stage, which is referred to as the $\overline{n}_s$th stage, the remaining unselected model terms will meet the condition

$$[\mathrm{cerr}]_l \le 0, \quad \overline{n}_s + 1 \le l \le L, \tag{22}$$

and this terminates the model construction process with a sparse model containing $\overline{n}_s \ll L$ significant model terms [7,16]. The OFS with the D-optimality algorithm utilising the modified Gram–Schmidt scheme is given in Appendix A.

3. PSO assisted regularisation parameter selection via Bayesian evidence

This section details the two-level algorithm used in the stage one of Fig. 1. From the discussion in Section 2.2, it can be seen that optimising the two regularisation parameters is crucial in order to produce the optimal prefiltered signal $y_{pre}(k)$ in terms of its generalisation ability. In this section, we describe the Bayesian framework for two level inference, and then the problem of Bayesian evidence maximisation for selecting the two regularisation parameters. Finally, the PSO algorithm is applied as the optimisation tool for the problem.

3.1. Likelihood and priors

At the first level of inference, the model parameters are inferred by the MAP estimate of the parameters [3]. Specifically,

for the regression model described in (6), the optimal $\boldsymbol{g}$ is obtained by maximising the posterior probability of $\boldsymbol{g}$, given by

$$p(\boldsymbol{g}|\boldsymbol{y},\boldsymbol{h},\rho)=\frac{p(\boldsymbol{y},\boldsymbol{g}|\boldsymbol{h},\rho)}{p(\boldsymbol{y}|\boldsymbol{h},\rho)}=\frac{p(\boldsymbol{y}|\boldsymbol{g},\rho)p(\boldsymbol{g}|\boldsymbol{h})}{p(\boldsymbol{y}|\boldsymbol{h},\rho)}, \tag{23}$$

where ρ denotes the inverse of the noise variance in the target, and the likelihood is assumed to be

$$p(\boldsymbol{y}|\boldsymbol{g},\rho)=\left(\frac{\rho}{2\pi}\right)^{N/2}\exp\left(-\frac{\rho}{2}\|\boldsymbol{y}-\boldsymbol{U}_r\boldsymbol{g}\|^2\right). \tag{24}$$

For Bayesian elastic net, the priors over $\boldsymbol{g}$ is assumed to be

$$p(\boldsymbol{g}|\boldsymbol{h})=(C(\boldsymbol{h}))^{n_s}\exp\left(-\frac{h_2}{2}\|\boldsymbol{g}\|^2-\frac{h_1}{2}\|\boldsymbol{g}\|_1\right), \tag{25}$$

with $\boldsymbol{h}=[h_1\ h_2]^{\mathrm{T}}$ denoting the vector of the two hyperparameters. Both h_1 and h_2 are positive parameters. This is a compromise between Gaussian and Laplacian distribution. It can be shown (see Appendix B) that the normalising constant $C(\boldsymbol{h})$ is given by

$$C(\boldsymbol{h})=\frac{\sqrt{h_2}}{\sqrt{2\pi}}\exp\left(-\frac{h_1^2}{8h_2}\right)\frac{1}{\mathrm{erfc}\left(\dfrac{h_1}{2\sqrt{2h_2}}\right)}, \tag{26}$$

where

$$\mathrm{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty}\exp\left(-t^2\right)dt. \tag{27}$$

Maximising $\log p(\boldsymbol{g}|\boldsymbol{y},\boldsymbol{h},\rho)$ with respect to $\boldsymbol{g}$ is equivalent to minimising the following Bayesian cost function:

$$L_B(\boldsymbol{h},\rho,\boldsymbol{g})=\rho\|\boldsymbol{y}-\boldsymbol{U}_r\boldsymbol{g}\|^2+h_2\|\boldsymbol{g}\|^2+h_1\|\boldsymbol{g}\|_1. \tag{28}$$

It can easily be seen that the criterion (28) is equivalent to (9) with the relationships $\lambda_1=h_1/\rho$ and $\lambda_2=h_2/\rho$.

3.2. Bayesian evidence

The second level inference is used to perform model selection, i.e. to determine which model structure or prior is more plausible given the data. In our problem, our aim is to find the optimal prefiltered signal $y_{pre}(k)$ with respect to the two regularisation parameters. To infer from the data what values should λ_1 and λ_2 have, we evaluate the evidence $p(\boldsymbol{y}|\boldsymbol{h},\rho)$ given by

$$\Xi(\boldsymbol{h},\rho)=p(\boldsymbol{y}|\boldsymbol{h},\rho)=\int p(\boldsymbol{y},\boldsymbol{g}|\boldsymbol{h},\rho)\,d\boldsymbol{g}, \tag{29}$$

in which $p(\boldsymbol{y},\boldsymbol{g}|\boldsymbol{h},\rho)$ can be rewritten as

$$\begin{aligned}p(\boldsymbol{y},\boldsymbol{g}|\boldsymbol{h},\rho)&=p(\boldsymbol{y}|\boldsymbol{g},\rho)p(\boldsymbol{g}|\boldsymbol{h})\\&=\left(\frac{\rho}{2\pi}\right)^{N/2}(C(\boldsymbol{h}))^{n_s}\\&\quad\times\exp\left(-\frac{\rho}{2}\|\boldsymbol{y}-\boldsymbol{U}_r\boldsymbol{g}\|^2-\frac{h_2}{2}\|\boldsymbol{g}\|^2-\frac{h_1}{2}\|\boldsymbol{g}\|_1\right).\end{aligned} \tag{30}$$

In general, the integral (29) is difficult to solve, and closed-form solutions are only available for very limited types of probability functions. Due to the orthogonality introduced by the proposed SVD based EN regularisation, this difficulty is alleviated.

To account for sparsity factor, denote the index set of the selected singular vectors in the prefilter as $\mathcal{S}$. Following Appendix C, we express the log evidence $\log(\Xi(\boldsymbol{h},\rho))$ as:

$$\begin{aligned}J(\boldsymbol{h},\rho)=&\underbrace{\frac{N}{2}\left(\log\left(\frac{\rho}{2\pi}\right)-\rho\right)}_{A}\\&+\underbrace{|\mathcal{S}|\log\left(\frac{1}{2}\sqrt{\frac{h_2}{\rho+h_2}}\right)}_{B}\underbrace{-|\mathcal{S}|\log\left(r\left(\frac{h_1}{2\sqrt{2h_2}}\right)\right)}_{C}\\&\underbrace{+\sum_{i\in\mathcal{S}}\log\left(r\left(\frac{\frac{h_1}{2}+\rho g_i^{(LS)}}{2\sqrt{\frac{\rho+h_2}{2}}}\right)+r\left(\frac{\frac{h_1}{2}-\rho g_i^{(LS)}}{2\sqrt{\frac{\rho+h_2}{2}}}\right)\right)}_{D},\end{aligned} \tag{31}$$

where $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$, $r(z)=\exp(z^2)\,\mathrm{erfc}(z)$, and $g_i^{(LS)}$ is the LS estimate of g_i. In the following, we reveal some remarkable properties about the maximum of the log evidence, which give some intuitive insights into the proposed approach. Clearly $r(z)>0$ and $r(0)=1$. In addition, we have:

Lemma 1. *$r(z)$ is a monotonically decreasing function.*

Proof. Proof is given in Appendix D.

It can be seen that the log evidence is composed of a number of additive terms, denoted by terms A, B, C and D in (31), that are either concave or monotonic. Their effects can be analysed as follows:

1. Term A is a concave function with respect to ρ, with the maximum occurring at $\rho=1$. This implies that ρ far away from 1 will be penalised when the log evidence is maximised.
2. The contribution from term B is always negative, reducing the log evidence. For fixed ρ, the larger h_2, the lesser the reduction.
3. The contribution from term C is always positive, increasing the log evidence. For fixed h_2, the larger h_1, the larger the contribution.
4. Term D is composed of the contributions from model terms. It can be verified that any model term with $|g_i^{(LS)}|>h_1/2\rho=\lambda_1/2$ increases the log evidence. This corresponds to the selected singular vector from (12).

While increasing h_1, which is favoured by the above point (3), there will be fewer terms to be remained in the model to gain the positive contributions due to term D. Effectively, the contributory model terms are in conflict with each other and, therefore, there exists a compromised solution. Basically, the best models are the simplest ones but also with sufficient number of model terms, in agreement with "Occam's razor".

3.3. Evidence maximisation using PSO

From the previous discussion, in order to obtain the optimal prefiltered signal $y_{pre}(k)$, the two regularisation parameters $\boldsymbol{\lambda}=[\lambda_1\ \lambda_2]^{\mathrm{T}}$ should be optimised, and this can be achieved by maximising the log evidence (31). Formally, this optimisation problem is stated as follows:

$$(\boldsymbol{h}_{\mathrm{opt}},\rho_{\mathrm{opt}})=\arg\max_{(\boldsymbol{h},\rho)}\{J(\boldsymbol{h},\rho)\}, \tag{32}$$

$$\boldsymbol{\lambda}_{\mathrm{opt}}=\frac{\boldsymbol{h}_{\mathrm{opt}}}{\rho_{\mathrm{opt}}}. \tag{33}$$

The evidence (31) is non-differentiable and, therefore, it is difficult to apply a gradient based optimisation algorithm. On the other hand, the conventional three-dimensional grid search is inefficient to solve the Bayesian evidence maximisation (32). We propose to apply the PSO algorithm [23,24] to efficiently solve the optimisation problem (32).

The PSO [23,24] constitutes a population based stochastic optimisation technique, which is 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. The entire swarm then endeavours to find a global optimal solution collaboratively by utilising swarm intelligence. Specifically, each

particle gradually adjusts its trajectory with the aid of its own cognitive information (its own best location) and the swarm's social information (the best position of the entire swarm) at each optimisation iteration. The PSO method is popular owing to its simplicity in implementation, inherent ability to rapidly converge to a "reasonably good" solution and to "steer clear" of local minima. It has been successfully applied to wide-ranging practical optimisation problems [12,27–32].

Referring to the stage one in Fig. 1, the upper level is the PSO optimiser with a population size of S. It learns the two optimal regularisation parameters based on the log evidence values provided by the lower level of the S particles. At the lower level, each particle calculates the associated the log evidence value using (31) for the given value of $(\boldsymbol{h}, \rho)$ by the upper level. For notational convenience, denote $\boldsymbol{\omega} = [\omega_1\ \omega_2\ \omega_3]^{\mathrm{T}} = [h_1\ h_2\ \rho]^{\mathrm{T}}$. The optimisation (32) is represented by

$$\boldsymbol{\omega}_{\mathrm{opt}} = \arg \max_{\boldsymbol{\omega} \in \prod_{j=1}^{3} [0, \Omega_{j,\max}]} J(\boldsymbol{\omega}), \tag{34}$$

where

$$\prod_{j=1}^{3} [0, \Omega_{j,\max}] \tag{35}$$

defines the search space. A swarm of particles, $\{\boldsymbol{\omega}_i^{(m)}\}_{i=1}^{S}$, that represent potential solutions are "flying" in the search space (35), where m denotes the iteration step. Each particle has a velocity, denoted as $\boldsymbol{\gamma}_i^{(m)}$, to direct its search. In order to avoid excessive roaming of particles beyond the search space [29], a velocity space is imposed, so that

$$\boldsymbol{\gamma}_i^{(m)} \in \prod_{j=1}^{3} [-\Upsilon_{j,\max}, \Upsilon_{j,\max}]. \tag{36}$$

The flowchart of the PSO optimiser is depicted in Fig. 2, where $I_{\max}$ denotes the maximum number of iterations, and $\boldsymbol{pb}_i^{(m)}$ denotes the cognitive information of the ith particle at the mth iteration, while $\boldsymbol{gb}^{(m)}$ is the swarm's social information at the mth iteration. The PSO algorithm for maximising the log evidence is summarised in Appendix E.

The search space (35) is defined by the specific problem to be solved, and the velocity space (36) can be empirically set. Usually, the velocity limit $\Upsilon_{j,\max}$ is related to $\Omega_{j,\max}$ by $\Upsilon_{j,\max} = \Omega_{j,\max}$ or $\Upsilon_{j,\max} = \Omega_{j,\max}/2$. Appropriate swarm size S and maximum number of iterations $I_{\max}$ are empirically chosen, and they can typically be set to relatively small values.

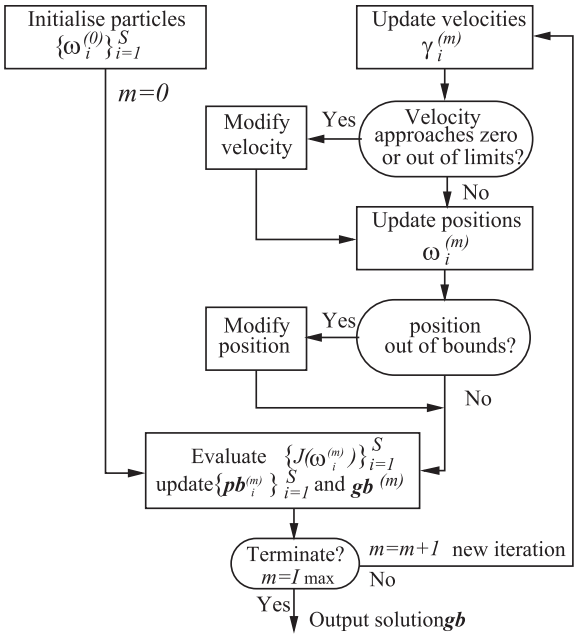


Fig. 2. Flowchart of PSO optimiser for Bayesian evidence maximisation.

3.4. Complexity of proposed two stage classifier construction

We have completed the descriptions of all the components for the proposed two-stage classifier construction algorithm depicted in Fig. 1. Our proposed algorithm overcomes two major obstacles. Firstly, we have derived the analytical Bayesian evidence formula based on the cost function (9) rather than (7), which would otherwise be very difficult to compute for models with non-orthogonal basis functions. Secondly, because the evidence formula is non-differentiable and subject to positiveness constraints of the regularisation parameters, this would make it very difficult for conventional gradient based optimization algorithms. Thus, we resort to the PSO as a highly effective optimisation tool to solve this problem.

We are now ready to analyse the computational complexity of the proposed two stage classifier construction procedure. The computational costs of the proposed two-stage algorithm as depicted in Fig. 1 comprise: (1) the cost of the SVD which is in the order of $O(N^3)$; (2) the cost of the PSO assisted two-level procedure for generating the optimal prefiltered signal, which is in the order of $O(S \times I_{\max} \times N)$; and (3) the cost of the OFS with D-optimality for the final construction of a sparse classifier, which is in the order of $O(L \times N)$. The total computational complexity is less than twice of that of the SVD. This is because $S \times I_{\max}$ and $\overline{n}_s$ are much smaller than N for the large training data sets. L can be set to the same value as N, or smaller when N is very large. Our extensive experience with the PSO algorithm suggests that S and $I_{\max}$ can be chosen to be relatively small values.

4. Experimental results

Eight two-class classification experiments were performed to demonstrate the effectiveness of the proposed algorithm, in comparison to the six existing state-of-the-arts classification algorithms studied in [33]. We also compare with our recent work, referred to as prefiltering with LOO algorithm [34], which is also based on the idea of elastic net based prefiltering, but the leave one out (LOO) misclassification rate was minimized for regularization parameters optimization. Eight noisy data sets were chosen from [35] for our experimentation, and they are: Banana, Breast Cancer, Diabetes, German, Heart, Flare Solar, Titanic, and Waveform. The specifications of these two-class data sets are listed in Table 1. Each data set contains 100 realisations, while each realisation consists of N training patterns and N_{test} test patterns, respectively.

The Gaussian RBF kernel $\phi_i(\boldsymbol{x}) = \exp(-(\|\boldsymbol{x}-\boldsymbol{c}_i\|^2)/2\sigma^2)$ was employed in all the experiments. A common kernel width σ was predetermined to derive individual models for all the 100 realisations of each data set. For each realisation of each data set, the full training data set was used as the RBF centre set $\{\boldsymbol{c}_i\}_{i=1}^{N}$ to form the

Table 1
Summary of the data sets [35].

Data set	Feature space dimension n	Training data size N	Test data size N_{test}	Number of realisations
Banana	2	400	4900	100
Breast Cancer	9	200	77	100
Diabetes	8	468	300	100
German	20	700	300	100
Heart	13	170	100	100
Flare Solar	9	666	400	100
Titanic	3	150	2051	100
Waveform	21	400	4600	100

candidate regressor set, namely, we set $L=N$. For each experimental data set listed in Table 1, 100 models were constructed over the 100 training data sets and the generalisation performance was evaluated using the average misclassification rate of the corresponding models over the 100 test data sets. The test performance and the average model sizes achieved by the proposed two stage classifier construction algorithm are summarised in Tables 2–9, respectively, for the eight experiment data sets, in comparison with the results of the six existing state-of-the-arts classification algorithms quoted from [33] as well as our recent work [34].

The test results listed in Tables 2–9 show that the proposed approach can construct parsimonious classifiers with competitive test classification accuracy for all the data sets experimented. Although our model sizes are not generally smaller than the first five methods, we point out that the first five methods quoted from [33] used various sophisticated non-linear optimisation algorithms to optimise the non-linear Gaussian RBF network with a fixed number of RBF units, where the model size for each experiment data set was predetermined using cross validation based on their RBF-based model (the first method). In other words, the first five methods cannot perform model structure selection automatically by the algorithms. Like our proposed method, the SVM classifier quoted in [33] is also based on the linear-in-the-parameters model structure with the full training data set used as the RBF centre set, and is capable of performing model structure selection automatically. For the SVM classifier, however, no average model size was given in [33]. Our extensive experience with the SVM method suggests that the SVM approach is generally not very sparse, and the average SVM model size was likely to be over 100 for each of these experiment data sets. Thus, our proposed method achieves a much sparser classifier than the SVM method. We further point out that the proposed algorithm is very robust in that a common value σ was used for all 100 realisations of each data set and we also found that the performance remained good over a wide range of the σ values, which indicates that our proposed method is insensitive to the kernel width σ. Hence, for practical use, the proposed method is a very good choice for classifiers with noisy data, especially if robust and superior classification performance is sought, with additional benefits of low computational cost and tuning effort.

Table 2
Average misclassification rate in % and model size over 100 realizations of the Banana test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	10.8 ± 0.6	18
Adaboost with RBF	12.3 ± 0.7	18
AdaBoost-Reg	10.9 ± 0.4	18
LP-Reg-AdaBoost	**10.7** ± 0.4	18
QP-Reg-AdaBoost	10.9 ± 0.5	18
SVM with RBF kernel	11.5 ± 0.7	Not available
Prefiltering with LOO [34]	**10.7** ± 0.5	28.7 ± 1.4
Proposed algorithm	**10.7** ± 0.5	27.6 ± 1.6

Table 3
Average misclassification rate in % and model size over 100 realizations of the Breast Cancer test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	27.6 ± 4.7	5
Adaboost with RBF	30.4 ± 4.7	5
AdaBoost-Reg	26.5 ± 4.5	5
LP-Reg-AdaBoost	26.8 ± 6.1	5
QP-Reg-AdaBoost	*25.9* ± 4.6	5
SVM with RBF kernel	26.0 ± 4.7	Not available
Prefiltering with LOO [34]	**25.0** ± 4.2	26.4 ± 2
Proposed algorithm	*25.1* ± 0.4	24.7 ± 1.9

Table 4
Average misclassification rate in % and model size over 100 realizations of the Diabetes test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	24.3 ± 1.9	15
Adaboost with RBF	26.5 ± 2.3	15
AdaBoost-Reg	23.8 ± 1.8	15
LP-Reg-AdaBoost	24.1 ± 1.9	15
QP-Reg-AdaBoost	25.4 ± 2.2	15
SVM with RBF kernel	*23.5* ± 1.7	Not available
Prefiltering with LOO [34]	**23.3** ± 1.7	7.7 ± 1.5
Proposed algorithm	*23.4* ± 1.7	7 ± 1.2

Table 5
Average misclassification rate in % and model size over 100 realizations of the German test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	24.7 ± 2.4	8
Adaboost with RBF	27.5 ± 2.5	8
AdaBoost-Reg	24.3 ± 2.1	8
LP-Reg-AdaBoost	24.8 ± 2.2	8
QP-Reg-AdaBoost	25.3 ± 2.1	8
SVM with RBF kernel	**23.6** ± 2.1	Not available
Prefiltering with LOO [34]	*24.3* ± 2.2	12.8 ± 1.3
Proposed algorithm	*24.2* ± 2.2	11.8 ± 1.4

Table 6
Average misclassification rate in % and model size over 100 realizations of the Heart test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	17.6 ± 3.3	4
Adaboost with RBF	20.3 ± 3.4	4
AdaBoost-Reg	16.5 ± 3.5	4
LP-Reg-AdaBoost	17.5 ± 3.5	4
QP-Reg-AdaBoost	17.2 ± 3.4	4
SVM with RBF kernel	*16.0* ± 3.3	Not available
Prefiltering with LOO [34]	**15.9** ± 3.0	8.8 ± 1.0
Proposed algorithm	*16.0* ± 3.0	8.8 ± 1.0

Table 7
Average misclassification rate in % and model size over 100 realizations of the Flare Solar test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	34.4 ± 2.0	4
Adaboost with RBF	35.7 ± 1.8	4
AdaBoost-Reg	34.2 ± 2.2	4
LP-Reg-AdaBoost	34.7 ± 2.0	4
QP-Reg-AdaBoost	36.2 ± 1.8	4
SVM with RBF kernel	**32.4** ± 1.8	Not available
Prefiltering with LOO [34]	*33.2* ± 1.7	6.7 ± 0.8
Proposed algorithm	*33.4* ± 1.6	7 ± 0.6

5. Conclusions

We have proposed an efficient two stage construction algorithm for linear-in-the-parameters two-class classifiers when robust and accurate classification is required over noisy data. The first stage of our approach constructs a prefiltered signal that is then used as the desired output for the second stage construction of a sparse linear-in-the-parameters classifier. The prefiltering stage is performed by a

Table 8
Average misclassification rate in % and model size over 100 realizations of the Titanic test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	23.3 ± 1.3	4
Adaboost with RBF	22.6 ± 1.2	4
AdaBoost-Reg	22.6 ± 1.2	4
LP-Reg-AdaBoost	24.0 ± 4.4	4
QP-Reg-AdaBoost	22.7 ± 1.1	4
SVM with RBF kernel	22.4 ± 1.0	Not available
Prefiltering with LOO [34]	$\mathbf{22.3} \pm 1.0$	11.1 ± 1.0
Proposed algorithm	$\mathbf{22.3} \pm 1.0$	11.1 ± 1.1

Table 9
Average misclassification rate in % and model size over 100 realizations of the Waveform test data set. The first six results are quoted from [33].

Method	Misclassification rate	Model size
RBF	10.7 ± 1.1	10
Adaboost with RBF	10.8 ± 0.6	10
AdaBoost-Reg	$\mathbf{9.8} \pm 0.8$	10
LP-Reg-AdaBoost	10.5 ± 1.0	10
QP-Reg-AdaBoost	10.1 ± 0.5	10
SVM with RBF kernel	9.9 ± 0.4	Not available
Prefiltering with LOO [34]	$\mathbf{9.8} \pm 0.4$	34.1 ± 1.9
Proposed algorithm	$\mathbf{9.8} \pm 0.4$	31.9 ± 2

novel two-level algorithm to maximise the model's generalisation capability. Using SVD, a new elastic net model identification algorithm is employed at the lower level, and the two regularisation parameters are found by a particle swarm optimisation algorithm to maximise Bayesian evidence at the upper level. Our original contributions are firstly to define an elastic net cost function based on left singular vectors, which facilitates: (i) the closed-form of elastic net solution based on a small number of singular vectors and (ii) efficient evaluation of Bayesian evidence using PSO. As a result, a fully automated procedure is achieved without resorting to any other validation data set for iterative model evaluation. Secondly, using mathematical analysis we provide insights as how "Occam's razor" is embodied in this approach. The second stage of sparse classifier construction is based on the well tested and highly efficient orthogonal forward regression with D-optimality algorithm. Eight benchmark examples are included to demonstrate the competitiveness of our new approach.

Acknowledgement

The authors gratefully acknowledge that part of this work was supported by the UK EPSRC.

Appendix A. The OFR with D-optimality using the modified Gram–Schmidt orthogonalisation procedure

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

$$\left.\begin{aligned} &\boldsymbol{w}_l = \boldsymbol{\phi}_l^{(l-1)}, \\ &a_{l,j} = \boldsymbol{w}_l^{\mathrm{T}} \boldsymbol{\phi}_j^{(l-1)} / (\boldsymbol{w}_l^{\mathrm{T}} \boldsymbol{w}_l), \ l+1 \le j \le L, \\ &\boldsymbol{\phi}_j^{(l)} = \boldsymbol{\phi}_j^{(l-1)} - a_{l,j} \boldsymbol{w}_l, \ l+1 \le j \le L, \end{aligned}\right\} \quad l = 1, 2, \ldots, L-1. \tag{37}$$

The last stage of the procedure is simply $\boldsymbol{w}_L = \boldsymbol{\phi}_L^{(L-1)}$. The elements of $\boldsymbol{\gamma}$ are computed by transforming $\boldsymbol{y}_{pre}^{(0)} = \boldsymbol{y}_{pre}$ in a similar way

$$\left.\begin{aligned} &\gamma_l = \boldsymbol{w}_l^{\mathrm{T}} \boldsymbol{y}^{(l-1)} / (\boldsymbol{w}_l^{\mathrm{T}} \boldsymbol{w}_l), \\ &\boldsymbol{y}_{pre}^{(l)} = \boldsymbol{y}_{pre}^{(l-1)} - \gamma_l \boldsymbol{w}_l, \end{aligned}\right\} 1 \le l \le L. \tag{38}$$

This orthogonalisation scheme can be used to derive a simple and efficient algorithm for selecting subset models in a forward-regression manner [8]. First define

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

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

Step 1. For $l \le j \le L$, compute

$$\left.\begin{aligned} &\gamma_l^{(j)} = (\boldsymbol{\phi}_j^{(l-1)})^{\mathrm{T}} \boldsymbol{y}^{(l-1)} / ((\boldsymbol{\phi}_j^{(l-1)})^{\mathrm{T}} \boldsymbol{\phi}_j^{(l-1)}), \\ &[\mathrm{cerr}]_l^{(j)} = ((\gamma_l^{(j)})^2 (\boldsymbol{\phi}_j^{(l-1)})^{\mathrm{T}} \boldsymbol{\phi}_j^{(l-1)} + \beta \log ((\boldsymbol{\phi}_j^{(l-1)})^{\mathrm{T}} \boldsymbol{\phi}_j^{(l-1)})) / (\boldsymbol{y}_{pre}^{\mathrm{T}} \boldsymbol{y}_{pre}). \end{aligned}\right\}$$

Step 2. Find

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

Then the j_lth column of $\boldsymbol{\Phi}^{(l-1)}$ is interchanged with the lth column of $\boldsymbol{\Phi}^{(l-1)}$, the j_lth column of $\boldsymbol{A}$ is interchanged with the lth column of $\boldsymbol{A}$ 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 $\boldsymbol{A}$ and to transform $\boldsymbol{\Phi}^{(l-1)}$ into $\boldsymbol{\Phi}^{(l)}$. Calculate γ_l and update $\boldsymbol{y}_{pre}^{(l-1)}$ into $\boldsymbol{y}_{pre}^{(l)}$ in the way shown in (38).

The selection is terminated at the $\overline{n}_s$ stage when the condition (22) is met, and this produces a subset model containing $\overline{n}_s$ significant regressors. The algorithm described here is in its standard form, a fast implementation can be adopted to reduce the computational cost [36].

Appendix B. Derivation of $C(\boldsymbol{h})$

From (25) by separating each component, we have

$$\begin{aligned} &\int_{-\infty}^{\infty} \exp\left(-\frac{h_2}{2} g_i^2 - \frac{h_1}{2} |g_i|\right) dg_i \\ &= \int_{-\infty}^{0} \exp\left(-\frac{h_2}{2} g_i^2 + \frac{h_1}{2} g_i\right) dg_i + \int_{0}^{\infty} \exp\left(-\frac{h_2}{2} g_i^2 - \frac{h_1}{2} g_i\right) dg_i \\ &= 2 \int_{0}^{\infty} \exp\left(-\frac{h_2}{2} g_i^2 - \frac{h_1}{2} g_i\right) dg_i \ \text{(by variable substitution)} \\ &= 2 \exp\left(\frac{h_1^2}{8h_2}\right) \int_{0}^{\infty} \exp\left(-\left(\frac{\sqrt{h_2}}{\sqrt{2}} g_i + \frac{h_1}{2\sqrt{2h_2}}\right)^2\right) dg_i \\ &= \frac{\sqrt{2\pi}}{\sqrt{h_2}} \exp\left(\frac{h_1^2}{8h_2}\right) \mathrm{erfc}\left(\frac{h_1}{2\sqrt{2h_2}}\right). \end{aligned} \tag{40}$$

By using $\int p(\boldsymbol{g}|\boldsymbol{h}, \rho)\, d\boldsymbol{g} = 1$, we have

$$C(\boldsymbol{h}) = \frac{\sqrt{h_2}}{\sqrt{2\pi}} \exp\left(-\frac{h_1^2}{8h_2}\right) \frac{1}{\mathrm{erfc}\left(\frac{h_1}{2\sqrt{2h_2}}\right)}. \tag{41}$$

Appendix C. Evaluating Bayesian evidence

The evidence is obtained by working out the following integral:

$$\begin{aligned}\Xi(\boldsymbol{h},\rho) &= \left(\frac{\rho}{2\pi}\right)^{N/2}(C(\boldsymbol{h}))^{n_s}\exp\left(-\frac{\rho\|\boldsymbol{y}\|^2}{2}\right)\\ &\quad\times\int\exp\left(-\frac{\rho}{2}\left(-2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{U}_r\boldsymbol{g}+\frac{\rho+h_2}{\rho}\|\boldsymbol{g}\|^2+\frac{h_1}{\rho}\|\boldsymbol{g}\|_1\right)\right)d\boldsymbol{g}\\ &= \left(\frac{\rho}{2\pi}\right)^{N/2}(C(\boldsymbol{h}))^{n_s}\exp\left(-\frac{\rho N}{2}\right)\prod_{i=1}^{n_s}I_i,\end{aligned}\tag{42}$$

where $\|\boldsymbol{y}\|^2=N$ and

$$\begin{aligned}I_i &= \int\exp\left(-\frac{\rho}{2}\left(-2g_i^{(LS)}g_i+\frac{\rho+h_2}{\rho}g_i^2+\frac{h_1}{\rho}|g_i|\right)\right)dg_i\\ &= \int_{-\infty}^{0}\exp\left(-\left(\left(-\frac{h_1}{2}-\rho g_i^{(LS)}\right)g_i+\frac{\rho+h_2}{2}g_i^2\right)\right)dg_i\\ &\quad+\int_0^{\infty}\exp\left(-\left(\left(\frac{h_1}{2}-\rho g_i^{(LS)}\right)g_i+\frac{\rho+h_2}{2}g_i^2\right)\right)dg_i.\end{aligned}\tag{43}$$

By variable substitution

$$\begin{aligned}I_i &= \int_0^{\infty}\exp\left(-\left(\left(\frac{h_1}{2}+\rho g_i^{(LS)}\right)g_i+\frac{\rho+h_2}{2}g_i^2\right)\right)dg_i\\ &\quad+\int_0^{\infty}\exp\left(-\left(\left(\frac{h_1}{2}-\rho g_i^{(LS)}\right)g_i+\frac{\rho+h_2}{2}g_i^2\right)\right)dg_i\\ &= \exp\left(\frac{b_{i1}^2}{4a^2}\right)\int_0^{\infty}\exp\left(-\left(ag_i+\frac{b_{i1}}{2a}\right)^2\right)dg_i\\ &\quad+\exp\left(\frac{b_{i2}^2}{4a^2}\right)\int_0^{\infty}\exp\left(-\left(ag_i+\frac{b_{i2}}{2a}\right)^2\right)dg_i\\ &= \frac{\sqrt{\pi}}{2a}\exp\left(\frac{b_{i1}^2}{4a^2}\right)\mathrm{erfc}\left(\frac{b_{i1}}{2a}\right)+\frac{\sqrt{\pi}}{2a}\exp\left(\frac{b_{i2}^2}{4a^2}\right)\mathrm{erfc}\left(\frac{b_{i2}}{2a}\right),\end{aligned}\tag{44}$$

with

$$\begin{cases}a=\sqrt{\dfrac{\rho+h_2}{2}}>0,\\ b_{i1}=\dfrac{h_1}{2}+\rho g_i^{(LS)},\\ b_{i2}=\dfrac{h_1}{2}-\rho g_i^{(LS)}.\end{cases}\tag{45}$$

Finally, we arrive the following formula for the evidence:

$$\begin{aligned}\Xi(\boldsymbol{h},\rho) &= \left(\frac{\rho}{2\pi}\right)^{N/2}(\tilde{C}(\rho,\boldsymbol{h}))^{n_s}\exp\left(-\frac{\rho N}{2}\right)\\ &\quad\times\prod_{i=1}^{n_s}\left(\exp\left(\frac{b_{i1}^2}{4a^2}\right)\mathrm{erfc}\left(\frac{b_{i1}}{2a}\right)+\exp\left(\frac{b_{i2}^2}{4a^2}\right)\mathrm{erfc}\left(\frac{b_{i2}}{2a}\right)\right),\end{aligned}\tag{46}$$

where

$$\tilde{C}(\rho,\boldsymbol{h})=\frac{1}{2}\sqrt{\frac{h_2}{\rho+h_2}}\exp\left(-\frac{h_1^2}{8h_2}\right)\frac{1}{\mathrm{erfc}\left(\dfrac{h_1}{2\sqrt{2h_2}}\right)}.\tag{47}$$

Appendix D. Proof of Lemma 1

Consider

$$r(z)=\exp(z^2)\mathrm{erfc}(z).\tag{48}$$

Using the identity $(d/dz)\mathrm{erfc}(z)=-(2/\sqrt{\pi})\exp(-z^2)$ and the inequality [37]

$$\mathrm{erfc}(z)\le\frac{2}{\sqrt{\pi}}\frac{\exp(-z^2)}{z+\sqrt{z^2+4/\pi}},\tag{49}$$

we have

$$\begin{aligned}\frac{d}{dz}r(z) &= 2z\exp(z^2)\mathrm{erfc}(z)+\exp(z^2)\left(-\frac{2}{\sqrt{\pi}}\exp(-z^2)\right)\\ &= 2z\exp(z^2)\mathrm{erfc}(z)-\frac{2}{\sqrt{\pi}}\\ &\le\frac{4}{\sqrt{\pi}}\frac{z}{z+\sqrt{z^2+4/\pi}}-\frac{2}{\sqrt{\pi}}=\frac{4}{\sqrt{\pi}}\frac{\frac{\sqrt{\pi}z}{2}}{\frac{\sqrt{\pi}z}{2}+\sqrt{\left(\frac{\sqrt{\pi}z}{2}\right)^2+1}}-\frac{2}{\sqrt{\pi}}.\end{aligned}\tag{50}$$

Letting $\tan\vartheta=\sqrt{\pi}z/2$, we have

$$\frac{d}{dz}r(z)\le\frac{4}{\sqrt{\pi}}\frac{\sin\vartheta}{1+\sin\vartheta}-\frac{2}{\sqrt{\pi}}\le 0.\tag{51}$$

This concludes the proof.

Appendix E. PSO optimiser for Bayesian evidence maximisation

Referring to the flowchart of Fig. 2, the PSO algorithm consists of the following steps:

(a) *Swarm initialisation*. Set the iteration index $m=0$ and randomly generate the initial population of the particles, $\{\boldsymbol{\omega}_i^{(m)}\}_{i=1}^{S}$, in the search space (35).

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

For $(i=1; i\le S; i++)$
  If $(J(\boldsymbol{\omega}_i^{(m)})>J(\boldsymbol{pb}_i^{(m)}))\ \boldsymbol{pb}_i^{(m)}=\boldsymbol{\omega}_i^{(m)}$;
End for
$i^*=\arg\max_{1\le i\le S}J(\boldsymbol{pb}_i^{(m)})$;
If $(J(\boldsymbol{\omega}_{i^*}^{(m)})>J(\boldsymbol{gb}^{(m)}))\ \boldsymbol{gb}^{(m)}=\boldsymbol{\omega}_{i^*}^{(m)}$;

(c) *Swarm update*. The velocity of the i particle is updated at each iteration according to

$$\begin{aligned}\boldsymbol{\gamma}_i^{(m+1)} &= \mu_1*\boldsymbol{\gamma}_i^{(m)}+\mu_1*rand()*(\boldsymbol{pb}_i^{(m)}-\boldsymbol{\omega}_i^{(m)})\\ &\quad+\mu_2*rand()*(\boldsymbol{gb}^{(m)}-\boldsymbol{\omega}_i^{(m)}),\end{aligned}\tag{52}$$

where *rand*() denotes the uniformly distributed random number in [0, 1], and μ_1 is known as the inertia weight, while μ_1 and μ_2 are the two acceleration coefficients. The generated velocity $\boldsymbol{\gamma}_i^{(m+1)}$ is then checked to make sure it is within the velocity space defined by (36) using the following operation:

$$\begin{aligned}&\text{If } (\boldsymbol{\gamma}_i^{(m+1)}|_j>\Upsilon_{j,\max})\ \boldsymbol{\gamma}_i^{(m+1)}|_j=\Upsilon_{j,\max},\\ &\text{If } (\boldsymbol{\gamma}_i^{(m+1)}|_j<-\Upsilon_{j,\max})\ \boldsymbol{\gamma}_i^{(m+1)}|_j=-\Upsilon_{j,\max},\end{aligned}\tag{53}$$

where $\boldsymbol{\gamma}|_j$ denotes the jth element of $\boldsymbol{\gamma}$. Moreover, if the velocity generated in (52) approaches zero, it is reinitialised proportional to $\Upsilon_{j,\max}$ with a small factor ν

$$\text{If } (\boldsymbol{\gamma}_i^{(m+1)}|_j==0)\ \boldsymbol{\gamma}_i^{(m+1)}|_j=\pm rand()*\nu*\Upsilon_{j,\max}.\tag{54}$$

The position of the ith particle is then updated according to

$$\boldsymbol{\omega}_i^{(m+1)}=\boldsymbol{\omega}_i^{(m)}+\boldsymbol{\gamma}_i^{(m+1)}.\tag{55}$$

Similarly, if a particle $\omega_i^{(m+1)}$ moves to outside the search space, it should be projected back to the boundary of the search space, or alternatively it can be moved back inside the search space to a random position.

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

Three common choices of the inertia weight are $\mu_I=0$, setting μ_I to a small positive constant, or $\mu_I=rand()$. We use $\mu_I=rand()$ at each iteration. An appropriate value of the small control factor ν in (54) for avoiding zero velocity is empirically found to be $\nu=0.1$ for our application. The two acceleration coefficients μ_1 and μ_2 can empirically be set to some appropriate constant values. However, the time varying acceleration coefficient (TVAC) mechanism [27], in which μ_1 is reduced from 2.5 to 0.5 and μ_2 is increased from 0.5 to 2.5 during the iterative procedure according to

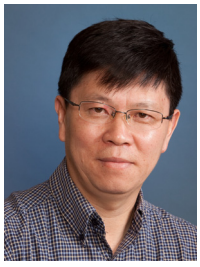
$$\mu_1=(0.5-2.5)*m/I_{\max}+2.5,$$
$$\mu_2=(2.5-0.5)*m/I_{\max}+0.5, \tag{56}$$

usually works well. 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 for better exploiting the search space, hence avoiding local solutions. In the later stages, a small cognitive component and a large social component help particles to converge quickly to a global solution.

References

[1] M. Stone, Cross-validatory choice and assessment of statistical predictions, J. Roy. Stat. Soc. Series B 36 (2) (1974) 117–147.
[2] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control AC-19 (December (6)) (1974) 716–723.
[3] D.J.C. MacKay, Bayesian Methods for Adaptive Models, Ph.D. Thesis, California Institute of Technology, USA, 1991.
[4] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag New York, 1995.
[5] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (June) (2001) 211–244.
[6] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machine, Regularization, Optimization and Beyond, MIT Press, Cambridge, MA, 2002.
[7] X. Hong, C.J. Harris, Nonlinear model structure design and construction using orthogonal least squares and D-optimality design, IEEE Trans. Neural Networks 13 (September (5)) (2002) 1245–1250.
[8] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their applications to non-linear system identification, Int. J. Control 50 (5) (1989) 1873–1896.
[9] K.Z. Mao, RBF neural network center selection based on fisher ratio class separability measure, IEEE Trans. Neural Networks 13 (September (5)) (2002) 1211–1217.
[10] S. Chen, X.X. Wang, X. Hong, C.J. Harris, Kernel classifier construction using orthogonal forward selection and boosting with fisher ratio class separability, IEEE Trans. Neural Networks 17 (November (6)) (2006) 1652–1656.
[11] X. Hong, S. Chen, C.J. Harris, A fast kernel classifier construction algorithm using orthogonal forward selection to minimize leave-one-out misclassification rate, Int. J. Syst. Sci. 39 (2) (2008) 119–125.
[12] S. Chen, X. Hong, C.J. Harris, Particle swarm optimization aided orthogonal forward regression for unified data modelling, IEEE Trans. Evolution. Comput. 14 (August (4)) (2010) 477–499.
[13] M.J.L. Orr, Regularisation in the selection of radial basis function centers, Neural Comput. 7 (May (3)) (1995) 954–975.
[14] S. Chen, E.S. Chng, K. Alkadhimi, Regularised orthogonal least squares algorithm for constructing radial basis function networks, Int. J. Control 64 (5) (1996) 829–837.
[15] 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 Networks 10 (September) (1999) 1239–1243.
[16] 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 (June (6)) (2003) 1029–1036.
[17] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (March (1)) (2001) 129–159.
[18] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc. Series B 58 (1) (1996) 267–288.
[19] B. Efron, I. Johnstone, T. Hastie, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2) (2004) 407–499.
[20] X. Hong, M. Brown, S. Chen, C.J. Harris, Sparse model identification using orthogonal forward regression with basis pursuit and D-optimality, IEE Proc. Control Theory Appl. 151 (4) (2004) 491–498.
[21] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. Roy. Stat. Soc. Series B 67 (2) (2005) 301–320.
[22] Q. Li, N. Lin, The Bayesian elastic net, Bayesian Anal. 5 (1) (2010) 151–170.
[23] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks (Perth, Australia), vol. 4, November 27–December 1, 1995, pp. 1942–1948.
[24] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, San Mateo, CA, 2001.
[25] R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, NJ, 1997.
[26] A.C. Atkinson, A.N. Donev, Optimum Experimental Designs, Clarendon, Oxford, UK, 1992.
[27] 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 (3)) (2004) 240–255.
[28] M.G.H. Omran, Particle swarm optimization methods for pattern recognition and image processing, Ph.D. Thesis, University of Pretoria, Pretoria, South Africa, 2005.
[29] 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 on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, Australia, December 5–8, 2005, pp. 319–324.
[30] 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. Vehicular Technol. 56 (September) (2007) 3006–3013.
[31] S. Chen, X. Hong, B.L. Luk, C.J. Harris, Non-linear system identification using particle swarm optimisation tuned radial basis function models, Int. J. Bio-Inspired Comput. 1 (4) (2009) 246–258.
[32] W. Yao, S. Chen, L. Hanzo, Particle swarm optimisation aided MIMO multi-user transmission designs, J. Comput. Theor. Nanosci. 9 (2) (2012) 266–275, special issue on a new frontier of cognitive informatics and cognitive computing.
[33] G. Rätsch, T. Onoda, K.-R. Müller, Soft margins for AdaBoost, Mach. Learn. 42 (3) (2001) 287–320.
[34] X. Hong, S. Chen, C.J. Harris, Elastic-net prefiltering for two-class classification, IEEE Trans. Cybern. 43 (February (1)) (2013) 286–295.
[35] ⟨http://www.fml.tuebingen.mpg.de/members/raetsch/benchmark⟩.
[36] S. Chen, J. Wigger, Fast orthogonal least squares algorithm for efficient subset selection, IEEE Trans. Signal Process. 43 (July (7)) (1995) 1713–1715.
[37] M. Abramowitz, I.A. Stegun (Eds.), Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables, Dover, Mineola, NY, 1965.

Xia Hong received her university education at National University of Defense Technology, PR 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 Reader 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 100 research papers, and coauthored a research book. She was awarded a Donald Julius Groen Prize by IMechE, in 1999.



Junbin Gao graduated from Huazhong University of Science and Technology (HUST), China, in 1982 with BSc degree in Computational Mathematics and obtained PhD from Dalian University of Technology, China, in 1991. He is a Professor in Computing Science in the School of Computing and Mathematics at Charles Sturt University, Australia, He was a Senior Lecturer, a Lecturer in Computer Science from 2001 to 2005 at University of New England, Australia. From 1982 to 2001 he was an Associate Lecturer, Lecturer, Associate Professor and Professor in Department of Mathematics at HUST. His main research interests include machine learning, data mining, Bayesian learning and inference, and image analysis.

Sheng Chen received his PhD degree in control engineering from the City University, London, UK, in September 1986. He was awarded the Doctor of Sciences (DSc) degree by the University of Southampton, Southampton, UK, in 2005. From October 1986 to August 1999, he held research and academic appointments at the University of Sheffield, the University of Edinburgh and the University of Portsmouth, all in UK. Since September 1999, he has been with the School of Electronics and Computer Science, University of Southampton, UK. Professor Chen's research interests include wireless communications, adaptive signal Processing for communications, machine learning, and evolutionary computation methods. He has published over 400 research papers. Dr. Chen is a Fellow of IET and a Fellow of IEEE. In the database of the world's most highly cited researchers, compiled by Institute for Scientific Information (ISI) of the USA, Dr. Chen is on the list of the highly cited researchers (2004) in the engineering category.

Chris Harris received university education at Leicester (BSc), Oxford (MA) and Southampton (PhD). He previously held appointments at the Universities of Hull, UMIST, Oxford and Cranfield, as well as being employed by the UK Ministry of Defence. His research interests are in the area of intelligent and adaptive systems theory and its application to intelligent autonomous systems, management infrastructures, intelligent control and estimation of dynamic processes, multi-sensor data fusion and systems integration. He has authored or co-authored 12 books and over 400 research papers, and he was the Associate Editor of numerous international journals including Automatica, Engineering Applications of AI, Int. J. General Systems Engineering, International J. of System Science and the Int. J. on Mathematical Control and Information Theory. He was elected to the Royal Academy of Engineering in 1996, was awarded the IEE Senior Achievement medal in 1998 for his work on autonomous systems, and the highest international award in IEE, the IEE Faraday medal in 2001 for his work in Intelligent Control and Neurofuzzy System.