Applied Soft Computing 23 (2014) 9–18

Contents lists available at ScienceDirect

Applied Soft Computing

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

A radial basis function network classifier to maximise leave-one-out mutual information☆

Xia Hong[a], Sheng Chen[b,c,*], Abdulrohman Qatawneh[c], Khaled Daqrouq[c], Muntasir Sheikh[c], Ali Morfeq[c]

[a] School of Systems Engineering, University of Reading, Reading RG6 6AY, UK
[b] Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
[c] Electrical & Computer Engineering Department, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 27 June 2012
Received in revised form 29 January 2014
Accepted 4 June 2014
Available online 12 June 2014

Keywords:
Cross validation
Mutual information
Orthogonal forward selection
Radial basis function classifier

ABSTRACT

We develop an orthogonal forward selection (OFS) approach to construct radial basis function (RBF) network classifiers for two-class problems. Our approach integrates several concepts in probabilistic modelling, including cross validation, mutual information and Bayesian hyperparameter fitting. At each stage of the OFS procedure, one model term is selected by maximising the leave-one-out mutual information (LOOMI) between the classifier's predicted class labels and the true class labels. We derive the formula of LOOMI within the OFS framework so that the LOOMI can be evaluated efficiently for model term selection. Furthermore, a Bayesian procedure of hyperparameter fitting is also integrated into the each stage of the OFS to infer the l^2-norm based local regularisation parameter from the data. Since each forward stage is effectively fitting of a one-variable model, this task is very fast. The classifier construction procedure is automatically terminated without the need of using additional stopping criterion to yield very sparse RBF classifiers with excellent classification generalisation performance, which is particular useful for the noisy data sets with highly overlapping class distribution. A number of benchmark examples are employed to demonstrate the effectiveness of our proposed approach.

1. Introduction

Model evaluation in terms of good generalisation performance is essential in the development and analysis of data-based learning algorithms for the construction of object classifiers. A fundamental concept in the evaluation of model generalisation capability is that of cross validation [1]. For example, in regression application, leave-one-out (LOO) cross validation is often used to estimate generalisation error by choosing amongst different model architectures [1]. In general, cross validation is required in most algorithms for model generalisation evaluation, and this often contributes significantly to computational cost for many model paradigms. Luckily for the linear-in-the-parameters models, the LOO cross validation can be exercised without actually splitting the training data set and estimating the associated models, by making use of the Sherman–Morrison–Woodbury theorem [2].

Moreover, for the linear-in-the-parameters models, the orthogonal least squares (OLS) based forward selection algorithm can efficiently construct parsimonious models [3,4], and has been a popular learning tool for associative neural networks, such as radial basis function (RBF) networks [5], fuzzy and neuro-fuzzy systems [6,7] as well as wavelets neural networks [8,9]. The OLS algorithm for RBF network learning [5] has also been utilised in a wide range of engineering applications, including aircraft gas turbine modelling [10], fuzzy control of multi-input multi-output nonlinear systems [11], power system control [12], fault detection [13], electric arc furnace load modelling [14], macromodelling of nonlinear digital I/O drivers [15], real-time power dispatch [16], fine tracking of NASA's 70-m-deep space network antennas [17], identification of urinary tract infection [18], stent reendothelialization [19], taxonomy and remote sensing of leaf mass per area [20], and many more.

For regression applications, regularisation methods based on a penalty function on l^2-norms of the model parameters are developed to carry out parameter estimation and model structure selection simultaneously [21–27]. From the powerful Bayesian

☆ X. Hong acknowledges the support of the UK EPSRC. This paper was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, under grant no. (1-4-1432/HiCi). The authors, therefore, acknowledge with thanks the DSR technical and financial support.

* Corresponding author.
E-mail addresses: x.hong@reading.ac.uk (X. Hong), sqc@ecs.soton.ac.uk (S. Chen), qatawneh@kau.edu.sa (A. Qatawneh), haleddaq@yahoo.com (K. Daqrouq), mshaikh@kau.edu.sa (M. Sheikh), morfeq@kau.edu.sa (A. Morfeq).

http://dx.doi.org/10.1016/j.asoc.2014.06.003
1568-4946/

learning viewpoint, it can be shown that for linear-in-the-parameters models this parameter regularisation is equivalent to a maximised *a posteriori* probability (MAP) estimate of the parameters by adopting a Gaussian prior for the model parameters [22,24–28]. Furthermore, 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 [24–28]. Note that, with the OLS algorithm, the evidence procedure for updating regularisation parameters becomes particularly efficient [22,25–27].

In information theory, the mutual information (MI) between two random variables is a quantity that measures the mutual dependence of the two variables [29,30]. The MI measure, as a fundamental measure in communications, has also been extensively used in regression applications, such as nonlinear system modelling [31,32], and pattern recognition applications, such as the feature selection [33], the registration of medical images [34] and gene classifications [35]. Note that in the existing literature MI criteria are normally used for training regression models or classifiers. Naturally if the MI is used as model structure selection metrics for classifier design, there is still the need to address model generalisation issue.

Against this background, in this work we propose to construct two-class RBF classifiers using the orthogonal forward selection (OFS) scheme, which selects one model term at each stage of the construction procedure by maximising the leave-one-out mutual information (LOOMI) between the classifier's predicted class labels and the true class labels, as well as incorporates a Bayesian procedure of hyperparameter fitting to efficiently derive the regularisation parameters. The paper contains two elements of novel contribution. Firstly, an original derivation of analytically evaluating the LOOMI efficiently is introduced, which facilitates the automatic model structure selection process with no need of using a predetermined error tolerance to terminate the forward selection process. Secondly, a novel Bayesian framework of calculating local regularisation parameters is designed specifically for the forward selection process, which leads to a very sparse classifier. Classification results for a number of benchmark examples demonstrate that our proposed approach efficiently construct very sparse RBF classifiers with excellent generalisation performance.

It is worthy emphasising that our contributions are significant. In the existing literature, the MI is used for training regression models and classifiers, but not used for model structure selection by optimising model generalisation capability. Instead of focusing on the usual training performance, to the best of our knowledge, our work is the first one that applies the MI for the effective model structure determination by introducing the novel LOOMI to incrementally maximise the classifier's model generalisation capability directly. Bayesian regularisation is also a well-known and widely used technique, e.g. in the support vector machine (SVM) and the relevance vector machine (RVM) [24] as well as in our previous orthogonal forward selection (OFS) based learning algorithms [22,25–27]. All these existing Bayesian regularisation approaches however involve an iterative procedure for updating the set of regularisation parameters. Specifically, given the values of all the regularisation parameters, model selection is carried out, and the resulting model is then used to update the set of regularisation parameters. This procedure iterates until both the selected model and the set of regularisation parameters converge. In this study, we introduce a novel Bayesian analysis for local regularisation parameter selection effectively nested within the OFS step. More particularly, each OFS stage also effectively fits one regularisation parameter from the data and this task is computationally very fast. Thus there is no need for iteratively performing the model selection and fitting the regularisation parameters several times. This paper is organised as follows. Section 2 introduces the two-class classifier construction using the OFS procedure and the concept of mutual information. In Section 3, we introduce model selection based on fast computing of the LOOMI. In Section 4, we carry out a Bayesian analysis for local regularisation parameter selection nested within the forward selection step. Section 5 presents the complete OFS algorithm that integrates joint parameter estimation with Bayesian regularisation and LOOMI model term selection. In Section 6, experimental results are employed to demonstrate the effectiveness of our proposed approach. Our conclusions are given in Section 7.

2. RBF classifier and mutual information

Consider the N labelled training data samples that belong to an approximately balanced two-class data set, denoted as $D_N = \{\boldsymbol{x}(k), y(k)\}_{k=1}^N$, where $\boldsymbol{x}(k) = [x_1(k)x_2(k)\cdots x_m(k)]^{\mathrm{T}} \in \mathbb{R}^m$ are m-dimensional feature vectors, and $y(k) \in \{\pm 1\}$ is the class type of $\boldsymbol{x}(k)$. We use the data set D_N to construct a RBF classifier of the form

$$\begin{cases} \tilde{y}^{(M)}(k) = \mathrm{sgn}(\hat{y}^{(M)}(k)), \\ \hat{y}^{(M)}(k) = f^{(M)}(\boldsymbol{x}(k)) = \displaystyle\sum_{i=1}^{M} \theta_i \phi_i(\boldsymbol{x}(k)), \end{cases} \tag{1}$$

where

$$\mathrm{sgn}(y) = \begin{cases} -1, & y \le 0, \\ 1, & y > 0, \end{cases} \tag{2}$$

$\tilde{y}^{(M)}(k)$ is the estimated class label for $\boldsymbol{x}(k)$ based on the M-term RBF model output $\hat{y}^{(M)}(k)$, and M is total number of regressors or model terms, while θ_i are the model weights, and the regressor $\phi_i(\boldsymbol{x})$ takes the form of Gaussian basis function given by

$$\phi_i(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|^2}{\tau}\right) \tag{3}$$

in which $\boldsymbol{c}_i = [c_{1,i}c_{2,i}\cdots c_{m,i}]^{\mathrm{T}}$ is the centre vector of the ith RBF unit and $\tau > 0$ is a RBF width parameter. We assume that each RBF unit is placed on a training data, namely, all the RBF centre vectors $\boldsymbol{c}_i$ are selected from the training data $\{\boldsymbol{x}(k)\}_{k=1}^N$, and the RBF width τ has been predetermined, for example, using cross validation.

Denote $e^{(M)}(k) = y(k) - \hat{y}^{(M)}(k)$ as the M-term modelling error for the data point $\boldsymbol{x}(k)$. Over the training data set D_N, further denote $\boldsymbol{y} = [y(1)y(2)\cdots y(N)]^{\mathrm{T}}$, $\boldsymbol{e}^{(M)} = [e^{(M)}(1)e^{(M)}(2)\cdots e^{(M)}(N)]^{\mathrm{T}}$, and $\boldsymbol{\Phi}_M = [\boldsymbol{\phi}_1\boldsymbol{\phi}_2\cdots\boldsymbol{\phi}_M]$ with $\boldsymbol{\phi}_l = [\phi_l(\boldsymbol{x}(1))\phi_l(\boldsymbol{x}(2))\cdots\phi_l(\boldsymbol{x}(N))]^{\mathrm{T}}$, $1 \le l \le M$. We have the M-term model in the matrix form of

$$\boldsymbol{y} = \boldsymbol{\Phi}_M\boldsymbol{\theta}_M + \mathbf{e}^{(M)}. \tag{4}$$

Here $\boldsymbol{\theta}_M = [\theta_1\theta_2\cdots\theta_M]^{\mathrm{T}}$. Let an orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}_M$ be

$$\boldsymbol{\Phi}_M = \boldsymbol{W}_M\boldsymbol{A}_M, \tag{5}$$

where

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

and

$$\boldsymbol{W}_M = [\boldsymbol{w}_1\boldsymbol{w}_2\cdots\boldsymbol{w}_M] \tag{7}$$

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

$$\boldsymbol{y} = \boldsymbol{W}_M \boldsymbol{g}_M + \boldsymbol{e}^{(M)}, \tag{8}$$

where the "orthogonal" model's weight vector $\boldsymbol{g}_M = [g_1 g_2 \cdots g_M]^{\mathrm{T}}$ satisfies the triangular system $\boldsymbol{A}_M \boldsymbol{\theta}_M = \boldsymbol{g}_M$, which can be used to determine the original model parameter vector $\boldsymbol{\theta}_M$, given $\boldsymbol{A}_M$ and $\boldsymbol{g}_M$.

Further consider the following l^2-norm regularised orthogonal least squares criterion for the model (8)

$$L_e(\boldsymbol{\Lambda}_M, \boldsymbol{g}_M) = \|\boldsymbol{y} - \boldsymbol{W}_M \boldsymbol{g}_M\|^2 + \boldsymbol{g}_M^{\mathrm{T}} \boldsymbol{\Lambda}_M \boldsymbol{g}_M, \tag{9}$$

where $\boldsymbol{\Lambda}_M = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_M\}$, which contains the local regularisation parameters $\lambda_i \geq 0$, for $1 \leq i \leq M$. The solution for $\boldsymbol{g}_M$ is obtained by solving $\partial L_e / \partial \boldsymbol{g}_M = \boldsymbol{0}$, yielding

$$g_i^{(R)} = \frac{\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_i}{\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_i + \lambda_i} g_i^{(LS)}, \tag{10}$$

with the usual least squares solution given by $g_i^{(LS)} = \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{y} / \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_i$.

The approach taken in this study is to construct a classifier in a forward selection manner, i.e. $\phi_i(\boldsymbol{x})$ is selected from a pool of candidate set and added one at a time to the classifier with some objective that is directly related to the classification performance, such as the misclassification rate [36] or the area under curve (AUC) of receiver operating characteristics (ROC) [37]. For example, by defining the M-term signed decision variable as

$$s_k^{(M)} = \mathrm{sgn}(y(k))\hat{y}^{(M)}(k) = y(k)\hat{y}^{(M)}(k), \tag{11}$$

the misclassification rate over the training data set D_N can be evaluated according to

$$MR(y, \hat{y}^{(M)}) = \frac{1}{N} \sum_{k=1}^{N} \mathcal{I}_d(s_k^{(M)}), \tag{12}$$

where the indication function $\mathcal{I}_d$ is defined as

$$\mathcal{I}_d(s) = \begin{cases} 1, & s \leq 0, \\ 0, & s > 0. \end{cases} \tag{13}$$

Alternatively, in this study, the MI between the two binary variables $y(k) \in \{\pm 1\}$ and $\tilde{y}^{(M)}(k) \in \{\pm 1\}$ is used, and this is defined by

$$MI(y, \tilde{y}^{(M)}) = \sum_{y(k)} \sum_{\tilde{y}^{(M)}(k)} p(y(k), \tilde{y}^{(M)}(k)) \times \log_2 \frac{p(y(k), \tilde{y}^{(M)}(k))}{p(y(k)) p(\tilde{y}^{(M)}(k))}, \tag{14}$$

where $p(\bullet)$ denotes the associated probabilities and $p(\bullet, \bullet)$ denotes the associated joint probabilities, respectively. Over the training data set D_N, these probabilities can be specifically calculated as

$$\begin{cases} p(y(k) = -1) = \dfrac{1}{N} \displaystyle\sum_{k=1}^{N} \mathcal{I}_d(y(k)), \\ p(y(k) = 1) = 1 - p(y(k) = -1), \end{cases} \tag{15}$$

$$\begin{cases} p(y(k) = 1, \tilde{y}^{(M)}(k) = 1) = \dfrac{1}{N} \displaystyle\sum_{k=1}^{N} \left((1 - \mathcal{I}_d(s_k^{(M)})) \left(\frac{y(k)+1}{2} \right) \right), \\ p(y(k) = 1, \tilde{y}^{(M)}(k) = -1) = \dfrac{1}{N} \displaystyle\sum_{k=1}^{N} \left(\mathcal{I}_d(s_k^{(M)}) \left(\frac{y(k)+1}{2} \right) \right), \\ p(y(k) = -1, \tilde{y}^{(M)}(k) = 1) = \dfrac{1}{N} \displaystyle\sum_{k=1}^{N} \left(\mathcal{I}_d(s_k^{(M)}) \left(\frac{1-y(k)}{2} \right) \right), \\ p(y(k) = -1, \tilde{y}^{(M)}(k) = -1) = \dfrac{1}{N} \displaystyle\sum_{k=1}^{N} \left((1 - \mathcal{I}_d(s_k^{(M)})) \left(\frac{1-y(k)}{2} \right) \right), \end{cases} \tag{16}$$

and

$$\begin{cases} p(\tilde{y}^{(M)}(k) = -1) = p(y(k) = 1, \tilde{y}^{(M)}(k) = -1) + p(y(k) = -1, \tilde{y}^{(M)}(k) = -1), \\ p(\tilde{y}^{(M)}(k) = 1) = 1 - p(\tilde{y}^{(M)}(k) = -1). \end{cases} \tag{17}$$

However, note that both the criteria (12) and (14) measure the classifiers' performance on the training data set only. In order to measure the model's generalisation capability, the expected classification performance over a fresh data set that has not been used in training should be employed. For the classifier construction based on the misclassification rate, this can be achieved based on the LOO misclassification rate [36]. Similarly, it is possible to develop the leave-one-out MI (LOOMI) by combining the concept of LOO cross validation with the MI. This is derived in the following section.

3. Model term selection based on LOOMI

When building a classifier, the ultimate goal is the best classification performance over unseen data. In our case, at each forward selection stage, we are faced with the task of model term selection aimed incrementally at this goal. The concept of leave-one-out (LOO) cross validation is often used to estimate generalisation error by choosing amongst different model architectures [1]. In the following, we develop the concept of LOOMI measure specifically for a forward selection stage.

At the lth forward selection step, where $l > 1$, the proposed algorithm selects the lth RBF unit based on a fast calculation of the LOOMI as detailed in this section. Consider the forward selection process at the stage where this l-unit model is produced. Let us denote the l-unit classifier, identified using the entire training data set D_N, as $f^{(l)}(\boldsymbol{x})$. The modelling error of this l-term classifier for the kth data point is given by

$$e^{(l)}(k) = y(k) - f^{(l)}(\boldsymbol{x}(k)) = y(k) - \hat{y}^{(l)}(k). \tag{18}$$

If we "remove" the kth data point from the training data set and use the remaining $(N-1)$ data points to identify the l-unit classifier instead, then the "test" error of the resulting model, which is denoted as $f^{(l,-k)}(\boldsymbol{x})$ for notational convenience, can be calculated on the data point removed from training. Specifically, the test output of this l-unit classifier at the kth data point not used in training is computed by

$$\hat{y}^{(l,-k)}(k) = f^{(l,-k)}(\boldsymbol{x}(k)), \tag{19}$$

and the associated predicted label is given by

$$\tilde{y}^{(l,-k)}(k) = \mathrm{sgn}(\hat{y}^{(l,-k)}(k)). \tag{20}$$

The test error at the kth data point, referred to as the LOO modelling error, is denoted as

$$e^{(l,-k)}(k) = y(k) - \hat{y}^{(l,-k)}(k), \tag{21}$$

and the associated LOO signed decision variable is then defined by

$$s_k^{(l,-k)} = y(k)\hat{y}^{(l,-k)}(k). \tag{22}$$

Denote the set $\mathcal{S}^{(l)} = \{s_k^{(l,-k)}\}_{k=1}^N$. The LOOMI is defined as the MI between the two binary variables $y(k) \in \{\pm 1\}$ and $\tilde{y}^{(l,-k)}(k) \in \{\pm 1\}$, and is a functional of $\mathcal{S}^{(l)}$, given by

$$J_l = MI(\mathcal{S}^{(l)}) = \sum_{y(k)} \sum_{\tilde{y}^{(l,-k)}(k)} p(y(k), \tilde{y}^{(l,-k)}(k)) \times \log_2 \frac{p(y(k), \tilde{y}^{(l,-k)}(k))}{p(y(k))p(\tilde{y}^{(l,-k)}(k))} \tag{23}$$

in which the associated probabilities are calculated based on (15) as well as (24) and (25) given below

$$\begin{cases} p(y(k)=1, \tilde{y}^{(l,-k)}(k)=1) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \left((1-\mathcal{I}_d(s_k^{(l,-k)}))\left(\frac{y(k)+1}{2}\right)\right), \\ p(y(k)=1, \tilde{y}^{(l,-k)}(k)=-1) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \left(\mathcal{I}_d(s_k^{(l,-k)})\left(\frac{y(k)+1}{2}\right)\right), \\ p(y(k)=-1, \tilde{y}^{(l,-k)}(k)=1) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \left(\mathcal{I}_d(s_k^{(l,-k)})\left(\frac{1-y(k)}{2}\right)\right), \\ p(y(k)=-1, \tilde{y}^{(l,-k)}(k)=-1) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \left((1-\mathcal{I}_d(s_k^{(l,-k)}))\left(\frac{1-y(k)}{2}\right)\right), \end{cases} \tag{24}$$

$$\begin{cases} p(\tilde{y}^{(l,-k)}(k)=-1) = p(y(k)=1, \tilde{y}^{(l,-k)}(k)=-1) + p(y(k)=-1, \tilde{y}^{(l,-k)}(k)=-1), \\ p(\tilde{y}^{(l,-k)}(k)=1) = 1 - p(\tilde{y}^{(l,-k)}(k)=-1). \end{cases} \tag{25}$$

For linear-in-the-parameters models, the LOO metrics, such as the LOO mean square error (LOOMSE) [26] and the LOO misclassification rate [36], can be generated without actually splitting the training data set and estimating the associated models, by making use of the Sherman–Morrison–Woodbury theorem [2]. Similarly, the LOOMI can also be obtained analytically without actually splitting the training data set and estimating the associated models. Specifically, we point out that the evaluation of the LOOMI given by (23) makes use of (15), (24) and (25), in which only the signed variable $s_k^{(l,-k)}$ and the class label $y(k)$ are needed. Since $s_k^{(l,-k)}$ can be analytically generated, the LOOMI of (23) can also be analytically computed. Clearly this helps computational efficiency significantly. Moreover, we show how the recursive computation as a consequence of orthogonal decomposition contribute further to computational efficiency in the model construction procedure based on maximising the LOOMI.

Specifically, let us represent the l-unit model identified using the entire training data set as

$$\hat{y}^{(l)}(k) = \sum_{i=1}^{l} g_i^{(R)} w_i(k), \tag{26}$$

where $w_i(k)$ is the kth element of $\boldsymbol{w}_i$ and $g_i^{(R)}$ is given in (10). Following the concept of LOO cross validation discussed above, it can be shown that the LOO modelling error at the kth data point is given by [2]

$$e^{(l,-k)}(k) = \frac{e^{(l)}(k)}{\eta_k^{(l)}} \tag{27}$$

where $e^{(l)}(k)$ is the l-term modelling error defined in (18) and $\eta_k^{(l)}$ is referred to as the LOO error weighting, which can be calculated by [26,36]

$$\eta_k^{(l)} = 1 - \sum_{i=1}^{l} \frac{w_i^2(k)}{\kappa_i + \lambda_i}, \tag{28}$$

where $\kappa_i = \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_i$. Eq. (27) is equivalent to

$$y(k) - \hat{y}^{(l,-k)}(k) = \frac{y(k) - \hat{y}^{(l)}(k)}{\eta_k^{(l)}}. \tag{29}$$

Multiplying the both sides of (29) with $y(k)$ and applying $y^2(k) = 1$ yield

$$1 - s_k^{(l,-k)} = \frac{1 - y(k)\hat{y}^{(l)}(k)}{\eta_k^{(l)}}. \tag{30}$$

that is,

$$s_k^{(l,-k)} = \frac{\sum_{i=1}^{l} y(k) g_i^{(R)} w_i(k) - \sum_{i=1}^{l} (w_i^2(k)/(\kappa_i + \lambda_i))}{\eta_k^{(l)}} = \frac{\psi_k^{(l)}}{\eta_k^{(l)}}. \tag{31}$$

It follows that the signed variable $s_k^{(l,-k)}$ of (31) for $1 \le k \le N$ can be obtained very efficiently via the recursive formula

$$\psi_k^{(l)} = \psi_k^{(l-1)} + y(k) g_l^{(R)} w_l(k) - \frac{w_l^2(k)}{\kappa_l + \lambda_l}, \tag{32}$$

$$\eta_k^{(l)} = \eta_k^{(l-1)} - \frac{w_l^2(k)}{\kappa_l + \lambda_l}. \tag{33}$$

Hence the LOOMI J_l defined in (23) can be calculated efficiently.

3.1. Initialisation

The initial condition of the forward selection is referred to as the forward selection step one when the classifier (1) has only one term. It is noted that at this stage the predicted class labels for all the data samples are identical to be either 1 or −1, dependent only on the sign of $g_1^{(R)} = \theta_1$, regardless of which candidate regressor is selected. Hence for the first step of forward selection, we select the

first regressor $\boldsymbol{\phi}_1$ based on minimising the LOOMSE [26], defined as

$$\frac{1}{N}\sum_{k=1}^{N}(e^{(l,-k)}(k))^2 = \frac{1}{N}\sum_{k=1}^{N}\frac{(e^{(l)}(k))^2}{(\eta_k^{(l)})^2}.$$

Since at the first stage $\boldsymbol{w}_1 = \boldsymbol{\phi}_1$, we have

$$e^{(1)}(k) = y(k) - g_1^{(R)}\phi_1(k),$$

$$\eta_k^{(1)} = 1 - \frac{\phi_1^2(k)}{\kappa_1 + \lambda_1},$$

where $\phi_1(k)$ is the kth element of $\boldsymbol{\phi}_1$, $\kappa_1 = \boldsymbol{\phi}_1^{\mathrm{T}}\boldsymbol{\phi}_1$ and $g_1^{(R)} = (\boldsymbol{\phi}_1^{\mathrm{T}}\boldsymbol{y}/(\kappa_1 + \lambda_1))$.

4. Bayesian local regularisation

The regularised OFS algorithm has two essential elements, model term selection and parameter estimation. Based on the l^2 regularisation, the closed-form solution of (10) can be interpreted as a maximum *a posterior* probability (MAP) estimate of the parameters with a Gaussian prior. It is known that the regularisation parameter is equivalent to the ratio of the related hyperparameter to the noise parameter within Bayesian framework, and can be optimised by maximising the marginal probability (evidence) as detailed below. In this section, we link the lth forward regression step to Bayesian learning framework and then derive the local regularisation parameter by maximising the evidence. Specifically consider the OFS modelling process that has produced the $(l-1)$-node RBF model. Let us denote the constructed $(l-1)$-column regression matrix as $\boldsymbol{W}_{l-1} = [\boldsymbol{w}_1\boldsymbol{w}_2\cdots\boldsymbol{w}_{l-1}]$. The model output vector of this $(l-1)$-node RBF is given by

$$\hat{\boldsymbol{y}}^{(l-1)} = \sum_{i=1}^{l-1} g_i^{(R)}\boldsymbol{w}_i, \tag{34}$$

and the corresponding modeling error vector can be obtained as $\boldsymbol{e}^{(l-1)} = \boldsymbol{y} - \hat{\boldsymbol{y}}^{(l-1)}$. The lth stage forward regression step is aimed at forming a l-node RBF model by adding the lth model column $\mathbf{w}_l$. Clearly this step can be represented by

$$\boldsymbol{e}^{(l-1)} = g_l\boldsymbol{w}_l + \boldsymbol{e}^{(l)}. \tag{35}$$

In a standard Bayesian two-level inference framework, the first level of inference infers the model parameters according to the principle of MAP estimation [28]. Specifically, consider (35) where $\boldsymbol{w}_l$ is assumed to be known, and the prior over g_l is assumed to be Gaussian

$$p(g_l|h_l) = \sqrt{\frac{h_l}{2\pi}}\exp\left(-\frac{h_l}{2}g_l^2\right), \tag{36}$$

with $h_l > 0$ denoting the hyperparameter. The optimal $g_l^{(R)}$ is obtained by maximising the posterior probability of g_l. The posterior probability of g_l is given by

$$\begin{aligned} p(g_l|\boldsymbol{e}^{(l-1)}, h_l, \varepsilon_l) &= \frac{p(\boldsymbol{e}^{(l-1)}, g_l|h_l, \varepsilon_l)}{p(\boldsymbol{e}^{(l-1)}|h_l, \varepsilon_l)} \\ &= \frac{p(\boldsymbol{e}^{(l-1)}|g_l, \varepsilon_l)p(g_l|h_l)}{p(\boldsymbol{e}^{(l-1)}|h_l, \varepsilon_l)}, \end{aligned} \tag{37}$$

where the likelihood is assumed to be Gaussian

$$p(\boldsymbol{e}^{(l-1)}|g_l, \varepsilon_l) = \left(\frac{\varepsilon_l}{2\pi}\right)^{N/2}\exp\left(-\frac{\varepsilon_l}{2}\|\boldsymbol{e}^{(l-1)} - g_l\boldsymbol{w}_l\|^2\right), \tag{38}$$

and $\varepsilon_l > 0$ denotes the inverse of the noise variance in the target.

Maximising $\log p(g_l|\,\boldsymbol{e}^{(l-1)}, h_l, \varepsilon_l)$ with respect to g_l is equivalent to minimising the following Bayesian cost function

$$L_B(h_l, \varepsilon_l, g_l) = \varepsilon_l\|\boldsymbol{e}^{(l-1)} - \boldsymbol{w}_l g_l\|^2 + h_l g_l^2. \tag{39}$$

It can easily verified that the criterion (39) is equivalent to (9) with the relationship $\lambda_l = h_l/\varepsilon_l$.

In order to infer from the data which value of λ_l is more plausible given the data, the second level inference with Bayesian framework is, for a given model basis vector $\boldsymbol{w}_l$, to evaluate the evidence $p(\boldsymbol{e}^{(l-1)}|h_l, \varepsilon_l)$ given by

$$E_l(h_l, \varepsilon_l) = p(\boldsymbol{e}^{(l-1)}|h_l, \varepsilon_l) = \int p(\boldsymbol{e}^{(l-1)}, g_l|h_l, \varepsilon_l)\,dg_l, \tag{40}$$

where

$$\begin{aligned} p(\boldsymbol{e}^{(l-1)}, g_l|h_l, \varepsilon_l) &= p(\boldsymbol{e}^{(l-1)}|g_l, \varepsilon_l)p(g_l|h_l) = \left(\frac{\varepsilon_l}{2\pi}\right)^{N/2} \\ &\times \sqrt{\frac{h_l}{2\pi}}\exp\left(-\frac{\varepsilon_l}{2}\|\boldsymbol{e}^{(l-1)} - \boldsymbol{w}_l g_l\|^2 - \frac{h_l g_l^2}{2}\right). \end{aligned} \tag{41}$$

Noting $\kappa_l = \boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{w}_l$ and $g_l^{(LS)} = \boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{e}^{(l-1)}/\kappa_l$ as well as $g_l^{(R)} = (\varepsilon_l\kappa_l/(\varepsilon_l\kappa_l + h_l g_l^{(LS)}))$, the evidence is derived in Eq. (42):

$$\begin{aligned} E_l(h_l, \varepsilon_l) &= \left(\frac{\varepsilon_l}{2\pi}\right)^{N/2}\sqrt{\frac{h_l}{2\pi}}\exp\left(-\frac{\varepsilon_l\|\boldsymbol{e}^{(l-1)}\|^2}{2}\right)\int\exp\left(-\frac{\varepsilon_l}{2}\left(-2g_l^{(LS)}\kappa_l g_l + \frac{\varepsilon_l\kappa_l + h_l}{\varepsilon_l}g_l^2\right)\right)dg_l \\ &= \left(\frac{\varepsilon_l}{2\pi}\right)^{N/2}\sqrt{\frac{h_l}{2\pi}}\exp\left(-\frac{\varepsilon_l\|\boldsymbol{e}^{(l-1)}\|^2}{2}\right)\exp\left(\frac{\varepsilon_l^2(g_l^{(LS)})^2\kappa_l^2}{2(\varepsilon_l\kappa_l + h_l)}\right)\int\exp\left(-\left(\sqrt{\frac{\varepsilon_l\kappa_l + h_l}{2}}g_l - \frac{\varepsilon_l g_l^{(LS)}\kappa_l}{2\sqrt{((\varepsilon_l\kappa_l + h_l)/2)}}\right)^2\right)dg_l \\ &= \left(\frac{\varepsilon_l}{2\pi}\right)^{N/2}\sqrt{\frac{h_l}{\varepsilon_l\kappa_l + h_l}}\exp\left(-\frac{\varepsilon_l\|\boldsymbol{e}^{(l-1)}\|^2}{2}\right)\exp\left(\frac{\varepsilon_l^2(g_l^{(LS)})^2\kappa_l^2}{2(\varepsilon_l\kappa_l + h_l)}\right) \\ &= \left(\frac{\varepsilon_l}{2\pi}\right)^{N/2}\sqrt{\frac{h_l}{\varepsilon_l\kappa_l + h_l}}\exp\left(-\frac{\varepsilon_l\|\boldsymbol{e}^{(l-1)}\|^2}{2}\right)\exp\left(\frac{(g_l^{(R)})^2(\varepsilon_l\kappa_l + h_l)}{2}\right), \end{aligned} \tag{42}$$

The log evidence is given by

$$\begin{aligned} \log E_l(h_l, \varepsilon_l) = \frac{N}{2}\log\frac{\varepsilon_l}{2\pi} + \frac{1}{2}\log h_l - \frac{1}{2}\log(\varepsilon_l\kappa_l + h_l) \\ - \frac{\varepsilon_l\|\boldsymbol{e}^{(l-1)}\|^2}{2} + \frac{(g_l^{(R)})^2(\varepsilon_l\kappa_l + h_l)}{2}. \end{aligned} \tag{43}$$

Setting $\partial \log E_l(h_l, \varepsilon_l)/\partial \varepsilon_l = 0$ and recalling $\lambda_l = h_l/\varepsilon_l$ yield

$$\frac{\partial \log E_l(h_l, \varepsilon_l)}{\partial \varepsilon_l} = \frac{N}{2\varepsilon_l} - \frac{\kappa_l}{2(\varepsilon_l\kappa_l + h_l)} - \frac{\|\boldsymbol{e}^{(l-1)}\|^2}{2} + \frac{(g_l^{(R)})^2\kappa_l}{2} + (\varepsilon_l\kappa_l + h_l)g_l^{(R)}\frac{\partial g_l^{(R)}}{\partial \varepsilon_l} = 0 \tag{44}$$

so that

$$\varepsilon_l = \frac{N - (\kappa_l/(\kappa_l + \lambda_l))}{\|\boldsymbol{e}^{(l-1)}\|^2 - (g_l^{(R)})^2(\kappa_l + 2\lambda_l)}. \tag{45}$$

By setting $(\partial \log E_l(h_l, \varepsilon_l))/\partial h_l = 0$, we have

$$\frac{\partial E_l(h_l, \varepsilon_l)}{\partial h_l} = \frac{1}{2h_l} - \frac{1}{2(\varepsilon_l\kappa_l + h_l)} + \frac{(g_l^{(R)})^2}{2} + (\varepsilon_l\kappa_l + h_l)g_l^{(R)}\frac{\partial g_l^{(R)}}{\partial h_l} = 0, \tag{46}$$

yielding

$$h_l = \frac{\kappa_l}{(g_l^{(R)})^2(\kappa_l + \lambda_l)}. \tag{47}$$

(45) and (47) constitute the recalculation formula for maximising the log evidence for a given $\mathbf{w}_l$. The above algorithm is simply fitting one regularisation parameter to one stage of the OFS, which selects a single term for the regression model in an orthogonal space. Therefore, it is computationally very efficient. Note that this regularisation parameter fitting is very different to all the existing regularisation based OFS algorithms [22,25–27] which involve an iterative procedure between the OFS model selection and the updating of all the regularisation parameters. Our proposed novel approach is computationally much more attractive. Similar to any Bayesian approach, the question as to whether the Gaussian prior is suitable can be argued. Indeed the convergence of the solution, (45) and (47), is data dependent. When $\boldsymbol{e}^{(l-1)}$ appears as random noise, the regularisation parameter will be driven to a high value and this yields a zero-value associated model parameter, leading to sparse models. On the other hand, this may be undesirable for some cases. For example, some low noise data sets with a complicated decision boundary requires little regularisation. Otherwise, the data sets may become ill-conditioned causing numerically instability for (45) and (47). Hence in our algorithm, at any stage if λ_l diverges or is above a very high value, it is reset as a small number, e.g. 10^{-6}, to allow the OFS to continue. This strategy proves to be useful for the overall numerically stability of the algorithm. Note that the termination of model selection is determined entirely by the LOOMI, unrelated to this strategy.

5. The proposed algorithm

The complete algorithm is presented below integrating (i) the model term selection criterion based on maximising the LOOMI, (ii) Bayesian local parameter regularisation, and (iii) the modified Gram–Schmidt orthogonalisation procedure [3]. Since every training data point is considered as a candidate centre, the candidate regression matrix $\boldsymbol{\Phi}_N \in \mathbb{R}^{N\times N}$. Define

$$\boldsymbol{\Phi}_N^{<l-1} = [\boldsymbol{w}_1\boldsymbol{w}_2\cdots\boldsymbol{w}_{l-1}\boldsymbol{\phi}_l^{<l-1}\cdots\boldsymbol{\phi}_N^{<l-1}], \tag{48}$$

with $\boldsymbol{\Phi}_N^{<0} = \boldsymbol{\Phi}_N$. If some of the columns in $\boldsymbol{\Phi}_N^{<l-1}$ have been interchanged, this will still be referred to as $\boldsymbol{\Phi}_N^{<l-1}$ for notational simplicity.

Initialisation. As explained at the end of Section 3, denote the first selected model term based on the LOOMSE minimisation as $\boldsymbol{\phi}_1$. Set $\mathbf{w}_1 = \boldsymbol{\phi}_1$. Given a very small positive value λ_1 (e.g. 10^{-6}), perform the following Bayesian iteration procedure for a predetermined number of times (e.g. 10):

$$g_1^{(R)} = \frac{\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{y}}{\kappa_1 + \lambda_1},$$
$$\varepsilon_1 = \frac{N - (\kappa_1/(\kappa_1 + \lambda_1))}{\|\boldsymbol{y}\|^2 - (g_1^{(R)})^2(\kappa_1 + 2\lambda_1)},$$
$$h_1 = \frac{\kappa_1}{(g_1^{(R)})^2(\kappa_1 + \lambda_1)},$$
$$\lambda_1 = \frac{h_1}{\varepsilon_1}.$$

Then recalculate $g_1^{(R)} = \boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{y}/(\kappa_1 + \lambda_1)$, set $\boldsymbol{e}^{(1)} = \boldsymbol{y} - g_1^{(R)}\boldsymbol{w}_1$, and for $1 \le k \le N$ calculate

$$\psi_k^{(1)} = y(k)g_1^{(R)}\phi_1(k) - \frac{\phi_1^2(k)}{\kappa_1 + \lambda_1},$$
$$\eta_k^{(1)} = 1 - \frac{\phi_1^2(k)}{\kappa_1 + \lambda_1},$$

where $\phi_1(k)$ denotes the kth element of $\boldsymbol{\phi}_1$.

5.1. The lth stage of selection procedure

At the beginning of the lth selection stage, we have the regression matrix given in (48). Perform the following steps:

Step 1): Set λ_l to a very small positive value (e.g. 10^{-6}). For $l \le j \le N$, denote the kth element of $\boldsymbol{\phi}_j^{<l-1}$ as $\phi_j^{<l-1}(k)$, and compute

$$\kappa_l^{[j]} = (\boldsymbol{\phi}_j^{<l-1})^{\mathrm{T}}\boldsymbol{\phi}_j^{<l-1},$$
$$g_l^{(R),[j]} = (\boldsymbol{\phi}_j^{<l-1})^{\mathrm{T}}\boldsymbol{e}^{(l-1)}/(\kappa_l^{[j]} + \lambda_l),$$
$$\psi_k^{(l),[j]} = \psi_k^{(l-1)} + y(k)g_l^{(R),[j]}\phi_j^{<l-1}(k) - \frac{(\phi_j^{<l-1}(k))^2}{\kappa_l^{[j]} + \lambda_l},$$
$$\eta_k^{(l),[j]} = \eta_k^{(l-1)} - \frac{(\phi_j^{<l-1}(k))^2}{\kappa_l^{[j]} + \lambda_l},$$
$$s_k^{(l,-k),[j]} = \frac{\psi_k^{(l),[j]}}{\eta_k^{(l),[j]}},$$

for $1 \le k \le N$. Then calculate

$$J_l^{[j]} = \sum_{y(k)}\sum_{\tilde{y}^{(l,-k),[j]}(k)} p(y(k), \tilde{y}^{(l,-k),[j]}(k)) \times \log_2 \frac{p(y(k), \tilde{y}^{(l,-k),[j]}(k))}{p(y(k))p(\tilde{y}^{(l,-k),[j]}(k))} \tag{49}$$

in which the associated probabilities $p(\bullet)$, $p(\bullet, \bullet)$ are calculated based on (15), (24) and (25), respectively, with $s_k^{(l,-k)}$ being replaced by $s_k^{(l,-k),[j]}$ as appropriate. Here $\tilde{y}^{(l,-k),[j]}(k)$ is conceptually used to denote the LOO predicted class label. Note that only $s_k^{(l,-k),[j]}$ are required in the calculation of the LOOMI (49).

Step 2): Find

$$J_l = J_l^{[j_l]} = \max\left\{J_l^{[j]}, l \le j \le N\right\}. \tag{50}$$

Then the j_lth column and the lth column of $\boldsymbol{\Phi}_N^{<l-1}$ are interchanged. The j_lth column and the lth column of $\boldsymbol{A}_N$ are interchanged up to the $(l-1)$th row. This effectively selects the resulting lth regressor $\boldsymbol{\phi}_l^{<l-1}$ in the subset model.

Step 3): Set $\boldsymbol{w}_l = \boldsymbol{\phi}_l^{<l-1>}$, and perform the following Bayesian iteration procedure for a predetermined number of times (e.g. 10)

$$g_l^{(R)} = \frac{\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{y}}{\kappa_l + \lambda_l},$$
$$\varepsilon_l = \frac{N - (\kappa_l/(\kappa_l + \lambda_l))}{\|\boldsymbol{e}^{(l-1)}\|^2 - (g_l^{(R)})^2(\kappa_l + 2\lambda_l)},$$
$$h_l = \frac{\kappa_l}{(g_l^{(R)})^2(\kappa_l + \lambda_l)},$$
$$\lambda_l = \frac{h_l}{\varepsilon_l}.$$

λ_l will be reset as 10^{-6} if it diverges or larger than a threshold (e.g. 10^6). Then set $\boldsymbol{e}^{(l)} = \boldsymbol{e}^{(l-1)} - g_l^{(R)}\boldsymbol{w}_l$. Calculate $\psi_k^{(l)}$ and $\eta_k^{(l)}$ for $1 \le k \le N$ using (32) and (33), respectively, and compute $s_k^{(l,-k)} = \psi_k^{(l)}/\eta_k^{(l)}$ for $1 \le k \le N$. This is followed by updating the value of the LOOMI J_l accordingly.

Step 4): Use the modified Gram–Schmidt orthogonalisation procedure [3] to derive the lth row of $\boldsymbol{A}_N$ and to transform $\boldsymbol{\Phi}_N^{<l-1>}$ into $\boldsymbol{\Phi}_N^{<l>}$

$$\boldsymbol{w}_l = \boldsymbol{\phi}_l^{<l-1>},$$
$$a_{l,j} = \frac{\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{\phi}_j^{<l-1>}}{\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{w}_l}, \quad l+1 \le j \le N,$$
$$\boldsymbol{\phi}_j^{<l>} = \boldsymbol{\phi}_j^{<l-1>} - a_{l,j}\boldsymbol{w}_l, \quad l+1 \le j \le N.$$

Termination. The selection procedure is terminated with the subset model of the M significant regressors when the following condition is detected

$$J_{M+i} \le J_M, \quad 1 \le i \le p, \tag{51}$$

subject to a minimum model size, where p is a preset number of steps.

Similar to any model selection, the model construction using forward selection deals with a tradeoff between overfitting and underfitting. The underfitting occurs if the model is too simple to handle a complex problem. The overfitting scenario is related to fitting a model to noisy training data by overly increasing model complexity. This overfitted model however is unlikely to have good classification performance for the unseen data. Since the LOOMI is based on cross validation, provided there is a sufficient number of model terms, J_M will be monotonically increasing until it reaches a global maximum, indicating that a suitable model size M has been achieved.

Since generally $M \ll N$, we can approximately estimate the computational cost of the proposed algorithm and conclude that it has a similar computational complexity to our previously proposed LOO cross validation based OFS algorithms, e.g. [36,37,38]. These algorithms have a computational complexity at $O(N^2)$, scaled by a small number of variables used in the algorithms. This is because at each OFS stage, there are also $O(N)$ candidates for model term selection in our algorithm, and the operations at each stage are based on vector operation with the size N. Thus, each OFS stage has a complexity of $O(N^2)$. The total cost of the algorithm is therefore $O(N^2)$ scaled by M which is far smaller than N.

6. Experimental results

6.1. Synthetic data set

The two-dimensional synthetic two-class problem [39] has 250 training data samples and 1000 test data samples. The Gaussian kernel function $\phi_i(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{c}_i\|^2/\tau)$ was employed, using all the 250 training data samples as the candidate centres $\boldsymbol{c}_i$ with the kernel width $\tau = 0.06$. The proposed algorithm was applied and a very sparse 4-centre RBF model was found. The classification boundary of the resulting RBF classifier was plotted in Fig. 1 together with the training data set. The misclassification rate over the test data set for this model was 9.7%, which is similar to 9.3% and 10.6% for the 4-term RVM and the 38-term SVM, respectively, reported in [24].

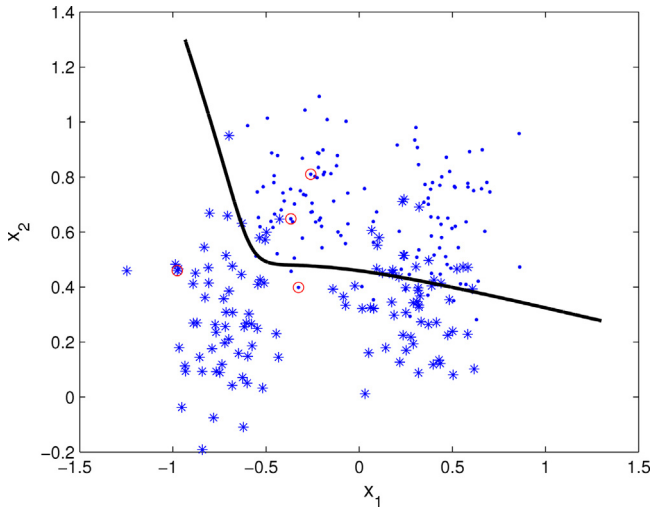

Fig. 1. Decision boundary produced by the proposed algorithm for synthetic data set, where stars and dots represent two-class data points, respectively, while circles are the selected centres.

6.2. Benchmark data sets

Nine data sets, Breast Cancer, Diabetes, German, Heart, Flare Solar, Titanic, Banana, Waveform and Thyroid data, taken from [40,41], were used in our experiment. Each data set contains 100 realisations, and each realisation has N training patterns and N_{test} test patterns. The feature space dimension m as well as the values of the training samples N and test samples N_{test} for each data set are summarised in Table 1.

The Gaussian kernel function $\phi_i(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{c}_i\|^2/\tau)$ is employed in the experiment. For each data set, models are constructed over the 100 training data realisation sets and generalisation performance is evaluated using the average misclassification rate of the corresponding models over the 100 test data realisation sets. For each realisation of all the nine benchmark data sets, the full training data set of N samples were used as the candidate RBF centres $\boldsymbol{c}_i$ to form the candidate regressor set. A common kernel width value τ was predetermined to derive individual models for all the 100 training realisations. Specifically, for the first training realisation data set, we evaluated the LOOMI performance of several kernel width values and chose the value that yielded a satisfactory

Table 1
Description of the benchmark data sets used.

Data set	m	N	N_{test}
Breast Cancer	9	200	77
Diabetes	8	468	300
German	20	700	300
Heart	13	170	100
Flare Solar	9	666	400
Titanic	3	150	2051
Banana	2	400	4900
Waveform	21	400	4600
Thyroid	5	140	75

Table 2
Average test misclassification rate in % and model size over 100 realisations of the Breast Cancer data set.

Algorithm	Test 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
Proposed algorithm	26.1 ± 4.7	3.7 ± 1.4

Table 3
Average test misclassification rate in % and model size over 100 realisations of the Diabetes data set.

Algorithm	Test 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
Proposed algorithm	23.7 ± 1.9	3.7 ± 0.8

Table 4
Average test misclassification rate in % and model size over 100 realisations of the German data set.

Algorithm	Test 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
Proposed algorithm	24.7 ± 2.1	4.9 ± 1.4

modelling performance. This kernel width value τ was then used in constructing the models for the other 99 training realisation data sets. We point out that the SVM results presented in [40,41] were also obtained with a same common kernel width for all the 100 realisations. As both the SVM and our method belong to the same class of fixed kernel models which choose training data points as kernel centres and use a single common kernel width for every kernel, we also use a same common kernel width for all the 100 realisations, for a fair comparison with the SVM results given in [40,41]. Alternatively, different kernel width values could be chosen for individual training realisation data sets, which would potentially result in better classification performance at the cost of increasing modelling complexity.[1]

The performance achieved by our proposed algorithm for the nine benchmark data sets are summarised in Tables 2–10, respectively, in comparison with those of the six existing algorithms quoted from [40,41]. The classification performance of the proposed approach is shown to be similar to the other existing state-of-the-arts methods. We point out that, in the work [40], the model size for the first five methods listed in Tables 2–10 are preset for each data set, and this model size has to be determined separately by some other means, e.g. cross validation. In other words,

[1] In fact, we also performed the same experiments with our method by using the LOOMI based cross validation to determine the kernel width for individual training realisation. The results obtained were slightly better than those obtained with a same kernel width for all the 100 training realisations, in terms of both model size and test misclassification rate. For a fair comparison with the SVM results, we only used the results obtained with the same kernel width for all the 100 training realisations.

Table 5
Average test misclassification rate in % and model size over 100 realisations of the Heart data set.

Algorithm	Test 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
Proposed algorithm	16.2 ± 3.4	3.7 ± 0.9

Table 6
Average test misclassification rate in % and model size over 100 realisations of the Flare Solar data set.

Algorithm	Test 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
Proposed algorithm	33.8 ± 2.5	3.9 ± 1.0

Table 7
Average test misclassification rate in % and model size over 100 realisations of the Titanic data set.

Algorithm	Test 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
Proposed algorithm	22.7 ± 0.9	3.3 ± 0.9

Table 8
Average test misclassification rate in % and model size over 100 realisations of the Banana data set.

Algorithm	Test 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
Proposed algorithm	10.8 ± 0.6	40.2 ± 0.6

Table 9
Average test misclassification rate in % and model size over 100 realisations of the waveform data set.

Algorithm	Test misclassification rate	Model size
RBF	10.7 ± 1.1	10
Adaboost with RBF	10.8 ± 0.6	10
AdaBoost$_{Reg}$	9.8 ± 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
Proposed algorithm	10.5 ± 0.6	50.1 ± 0.3

Table 10
Average test misclassification rate in % and model size over 100 realisations of the Thyroid data set.

Algorithm	Test misclassification rate	Model size
RBF	4.5 ± 2.1	10
Adaboost with RBF	4.4 ± 2.2	8
AdaBoost$_{Reg}$	4.6 ± 2.2	8
LP$_{Reg}$-AdaBoost	4.6 ± 2.2	8
QP$_{Reg}$-AdaBoost	4.4 ± 2.2	8
SVM with RBF kernel	4.8 ± 2.2	Not available
Proposed algorithm	4.8 ± 2.4	100 ± 0.2

Table 11
Total recorded running time used in training and evaluation.

Data set	Breast Cancer	Diabetes	German	Heart	Flare Solar	Titanic	Banana	Waveform	Thyroid
Running time (s)	7.88	32.8	86.3	6.75	63.99	6.78	130.52	227.72	51.85

all these five algorithms cannot perform model structure selection automatically. By contrast, our proposed algorithm automatically determines the appropriate model size. The sixth algorithm used for comparison, the SVM, is capable of automatically determining an appropriate model size. However, the models produced by the SVM are in fact not very sparse, and no average model size was given for the SVM in [40,41]. Our experience suggests that typical model size produced by the SVM is likely to be in the range of tens to hundreds, much much larger than the model size determined by our algorithm.

The results shown in Tables 2–7 demonstrate that our proposed algorithm can automatically construct very parsimonious classifiers with comparable classification accuracy to those benchmark algorithms investigated in [40,41] for these data sets. We point out that during the OFS procedure for these first six examples, the Bayesian local regularisation via hyperparameter fitting works in full in the sense that regularisation parameters are not being reset to small values. Note that these six data sets are very noisy, as shown by their relatively high achievable test misclassification rates. By contrast, the results shown in Tables 8–10 show that our models are not as sparse as the models produced by the first five approaches. Note that these three data sets are less noisy compared with the previous six data sets, as shown by smaller achievable test misclassification rates. For the fixed kernel modellings, such as the SVM and our method, these data sets require larger oversized models for all the training realisation data sets in order to avoid underfitting. Our model size is determined by the LOOMI together with the constraint of a minimum model size. For these three data sets, the regularisation parameter is often reset to a small value as the model size grows during the OFS procedure. Finally we list the total running time of the training and evaluation for each data set using Matlab on a computer Intel®Core(™)2 CPU 6400@2.13 GHz in Table 11, which confirms that the run time of our algorithm is clearly very low.

7. Conclusions

An OFS algorithm for automatically constructing RBF classifiers has been proposed based on a new model-term selection criterion that maximises the leave-one-out mutual information between the classifier's predicted class labels and the true class labels. Integrated within each OFS step, a Bayesian procedure of hyperparameter fitting has been introduced to infer the l^2-norm local regularisation parameter from the data. In our algorithm, model terms are selected by directly optimising the classifier's generalisation performance, and Bayesian evidence procedure for fitting the local regularisation parameter significantly enhances the sparsity of the constructed RBF classifier. Consequently, our RBF classifier construction procedure automatically terminates without any additional stopping criterion to yield very parsimonious RBF classifiers with excellent classification generalisation performance. Several benchmark examples have been employed to demonstrate the effectiveness of our proposed approach, in particular the ability of constructing very sparse models automatically with similar good generalisation performance as some well-known existing state-of-the-arts methods reported in the literature.

Our future work will further investigate this sparse classifier construction algorithm using other real-life benchmark data sets. For the challenging class of high-dimensional classification problems, where the feature space dimension is extremely large, in thousands or even tens of thousands, but the sample size is extremely small by comparison, in hundreds or even in tens, efficient feature selection becomes essential. We are currently investigating suitable feature selection techniques for integrating with the proposed algorithm in order to tackle this type of challenging high-dimensional classification problems effectively.

References

[1] M. Stone, Cross-validatory choice and assessment of statistical predictions, J. R. Stat. Soc. Ser. B 36 (2) (1974) 117–147.
[2] R.H. Myers, Classical and Modern Regression with Applications, 2nd ed., PWS-KENT, Boston, 1990.
[3] 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.
[4] M.J. Korenberg, Identifying nonlinear difference equation and functional expansion representations: the fast orthogonal algorithm, Ann. Biomed. Eng. 16 (1) (1988) 123–142.
[5] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Netw. 2 (2) (1991) 302–309.
[6] L.-X. Wang, J.M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, IEEE Trans. Neural Netw. 5 (5) (1992) 807–814.
[7] 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.
[8] Q. Zhang, Using wavelets network in nonparametric estimation, IEEE Trans. Neural Netw. 8 (2) (1997) 227–236.
[9] 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.
[10] N. Chiras, C. Evans, D. Rees, Nonlinear gas turbine modeling using NARMAX structures, IEEE Trans. Instrum. Meas. 50 (4) (2001) 893–898.
[11] Y. Gao, M.J. Er, Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems, IEEE Trans. Fuzzy Syst. 11 (4) (2003) 462–477.
[12] 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.
[13] G.-C. Luh, W.-C. Cheng, Identification of immune models for fault detection, Proc. Inst. Mech. Eng. I: J. Syst. Control Eng. 218 (2004) 353–367.
[14] G.W. Chang, C. Chen, Y. Liu, A neural-network-based method of modeling electric arc furnace load for power engineering study, IEEE Trans. Power Syst. 25 (1) (2010) 138–146.
[15] B. Mutnury, M. Swaminathan, J.P. Libous, Macromodeling of nonlinear digital I/O drivers, IEEE Trans. Adv. Pack. 29 (1) (2006) 102–113.
[16] C. Huang, F. Wang, An RBF network with OLS and EPSO algorithms for real-time power dispatch, IEEE Trans. Power Syst. 22 (1) (2007) 96–104.
[17] R. Mukai, V.A. Vilnrotter, P. Arabshahi, V. Jammejad, Adaptive acquisition and tracking for deep space array feed antennas, IEEE Trans. Neural Netw. 13 (5) (2002) 1149–1162.
[18] V.S. Kodogiannis, J.N. Lygouras, A. Tarczynski, H.S. Chowdrey, Artificial odor discrimination system using electronic nose and neural networks for the identification of urinary tract infection, IEEE Trans. Inf. Technol. Biomed. 12 (6) (2008) 707–713.
[19] C. Kauffmann, P. Motreff, L. Sarry, In vivo supervised analysis of stent reendothelialization from optical coherence tomography, IEEE Trans. Med. Imaging 29 (3) (2010) 807–818.
[20] G.P. Asner, R.E. Martin, R. Tupayachi, et al., Taxonomy and remote sensing of leaf mass per area (LMA) in humid tropical forests, Ecol. Appl. 21 (1) (2011) 85–98.
[21] M.J.L. Orr, Regularization in the selection of radial basis function centers, Neural Comput. 7 (3) (1995) 606–623.
[22] 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.
[23] 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 (5) (1999) 1239–1243.
[24] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.
[25] 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.

[26] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modelling using forward regression with PRESS statistic and regularization, IEEE Trans. Syst. Man Cybern. B 34 (2) (2004) 898–911.
[27] S. Chen, Local regularization assisted orthogonal least squares regression, Neurocomputing 69 (4-6) (2006) 559–585.
[28] D.J.C. MacKay, Bayesian Methods for Adaptive Models (Ph.D. Thesis), California Institute of Technology, USA, 1992.
[29] C.E. Shannon, A mathematical theory of information, Bell Syst. Tech. J. 27 (1948) 379–423.
[30] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley & Sons, 1991.
[31] G.L. Zheng, S.A. Billings, Radial basis function network configuration using mutual information and the orthogonal least squares algorithm, Neural Netw. 9 (9) (1996) 1619–1637.
[32] F. Rossi, A. Lendasse, D. François, V. Wertz, M. Verleysen, Mutual information for the selection of relevant variables in spectrometric nonlinear modelling, Chemom. Intell. Lab. Syst. 80 (2) (2006) 215–226.
[33] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[34] J.P.W. Pluim, J.B.A. Maintz, M.A. Viergever, Mutual information based registration of medical images: a survey, IEEE Trans. Med. Imaging 22 (8) (2003) 986–1004.
[35] X. Zhou, X. Wang, E.R. Dougherty, Nonlinear probit gene classification using mutual information and wavelet-based feature selection, J. Biol. Syst. 12 (3) (2004) 371–386.
[36] 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.
[37] X. Hong, S. Chen, C.J. Harris, A kernel-based two class classifier for imbalanced data sets, IEEE Trans. Neural Netw. 18 (1) (2007) 28–41.
[38] X. Hong, P.M. Sharkey, K. Warwick, Automatic nonlinear predictive model construction algorithm using forward regression and the PRESS statistic, IEE Proc. Control Theory Appl. 150 (3) (2003) 245–254.
[39] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.
[40] G. Rätsch, T. Onoda, K.R. Müller, Soft margins for AdaBoost, Mach. Learn. 42 (3) (2001) 287–320.
[41] G. Rätsch, http://www.raetschlab.org/members/raetsch