Neurocomputing 74 (2011) 3456–3466

Contents lists available at ScienceDirect

Neurocomputing

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

A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems

Ming Gao [a], Xia Hong [a,*], Sheng Chen [b], Chris J. Harris [b]

[a] School of Systems Engineering, University of Reading, Reading RG6 6AY, UK
[b] School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

ARTICLE INFO

Article history:
Received 7 March 2011
Received in revised form
26 May 2011
Accepted 5 June 2011
Communicated by K. Li
Available online 28 June 2011

Keywords:
Imbalanced classification
Synthetic minority over-sampling technique
Radial basis function classifier
Orthogonal forward selection
Particle swarm optimisation

ABSTRACT

This contribution proposes a powerful technique for two-class imbalanced classification problems by combining the synthetic minority over-sampling technique (SMOTE) and the particle swarm optimisation (PSO) aided radial basis function (RBF) classifier. In order to enhance the significance of the small and specific region belonging to the positive class in the decision region, the SMOTE is applied to generate synthetic instances for the positive class to balance the training data set. Based on the over-sampled training data, the RBF classifier is constructed by applying the orthogonal forward selection procedure, in which the classifier's structure and the parameters of RBF kernels are determined using a PSO algorithm based on the criterion of minimising the leave-one-out misclassification rate. The experimental results obtained on a simulated imbalanced data set and three real imbalanced data sets are presented to demonstrate the effectiveness of our proposed algorithm.

1. Introduction

A classification problem is referred to as imbalanced when the instances in one or several classes, known as the majority classes, outnumber the instances of the other classes, called the minority classes. Such an imbalance in the data represents the so-called between-class imbalance [1], in contrast to the related issue of within-class imbalance [2,3]. Imbalanced problems widely exist in the fields of medical diagnosis, science and engineering, and some examples includes surveillance of nosocomial infection [4], cardiac care [5] and elucidating protein–protein interactions [6] as well as fraud detection [7,8], network intrusion detection [9] and telecommunication management [10]. Note that, in an imbalance problem, the minority classes are usually the more important classes. For instance, 11% of patients suffer from one or more nosocomial infections [4]. In the study of two-class imbalanced problems, the instances in the majority class are referred to as negative, while in its counterpart, the minority class, the instances are referred to as positive. Since in practice the minority class is more important, one should be more concerned with the positive instances. Imbalanced data learning has been widely researched [11–16]. Typically, the approaches for solving the imbalanced problem can be divided into two categories: re-sampling methods and imbalanced learning algorithms.

The re-sampling approach is actually a re-balancing process to balance the given imbalanced data set. The studies [17,18] on class distribution have shown that balanced data sets provide better classification performance than imbalanced ones, though some other studies [1,19] have argued that imbalanced data sets are not necessarily responsible for the poor performance of some classifiers. Re-sampling techniques are attractive under most imbalanced circumstances. This is because re-sampling adjusts only the original training data set, instead of modifying the learning algorithm. Thus, this approach is external and transportable [18,20], and it provides a convenient and effective way to deal with imbalanced learning problems using standard classifiers. Specifically, the re-sampling methods include the random over-sampling, which randomly appends replicated instances to the positive class, and the random under-sampling, which randomly removes instances from the majority class. Alternatively, there exist the guided over-sampling and under-sampling, respectively, of which the choices to replicate or to eliminate are informed rather than random. In addition, the synthetic minority over-sampling technique (SMOTE) [21] is a well acknowledged over-sampling method. In the SMOTE, instead of mere data oriented duplicating, the positive class is over-sampled by creating synthetic instances in the feature space formed by the positive instances and their K-nearest neighbours.

* Corresponding author.
E-mail addresses: ming.gao@pgr.reading.ac.uk (M. Gao), x.hong@reading.ac.uk (X. Hong), sqc@ecs.soton.ac.uk (S. Chen), cjh@ecs.soton.ac.uk (C.J. Harris).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.06.010

The second category, consisting of imbalanced learning algorithms, can be regarded as a process to modify or re-balance the existing learning algorithms so that they can deal with imbalanced problems effectively. The imbalanced learning algorithms include the cost-sensitive method [22–25], the discrimination-based and recognition-based approaches [3]. An alternative is to adapt standard kernel-based or radial basis function (RBF) classifiers, which use a fixed common variance for every RBF kernel and choose RBF centres from input data, to imbalanced data sets by modifying the kernel construction and model selection procedure. A representative work [26] of this imbalanced learning proposes a regularised weighted least square estimator (LSE) using the orthogonal forward selection (OFS) based on the model selection criterion of maximising the leave-one-out (LOO) area under the curve (AUC) of receiver operating characteristics (ROC). In this LOO-AUC+OFS algorithm [26], the cost function of the LSE is made sensitive to the class labels, such that the errors due to minority class data samples are given a higher weight $\rho \geq 1$, and this weighted LSE (WLSE) reduces to the standard LSE with the weight $\rho = 1$. A well-known RBF modelling is the two staged procedure [27], in which the RBF centres are first determined using the κ-means clustering [28] and the RBF weights are then obtained using the LSE. To cope with imbalanced data sets, a natural extension of [27] is to modify the latter stage as the WLSE, where the same weighted cost function of [26] is used. This κ-means +WLSE algorithm provides a viable alternative within this imbalanced learning category.

Kernel-based learning, such as support vector machine (SVM) and RBF, is widely used for solving balanced learning problems. In particular, a powerful approach for constructing the RBF and other sparse kernel classifiers is to assign a fixed common variance for every kernel and to select input data as the candidate centres for RBF kernels by minimising the leave-one-out (LOO) misclassification rate in the efficient OFS procedure [29]. This approach has its root in regression application [30–33]. Two limitations may be associated with this "fixed" RBF kernel approach. Firstly, RBF kernels cannot be flexibly tuned, as the position of each kernel is restricted to the input data and the shape of each kernel is fixed rather than determined by the learning procedure. Secondly, the common kernel variance has to be determined via cross validation, which inevitably increases the computational cost. The previous studies [34–36] have proposed to construct the tunable RBF classifier based on the OFS procedure using a global search optimisation algorithm [37] to optimise the RBF kernels one by one. This tunable RBF kernel approach is observed to produce sparser classifiers with better performance but higher computational complexity in classifier construction, in comparison with the standard fixed kernel approach. Recently, the particle swarm optimisation (PSO) algorithm [38] is adopted to minimise the LOO misclassification rate in the OFS construction of tunable RBF classifier [39,40]. PSO [38] is an efficient population-based stochastic optimisation technique inspired by social behaviour of bird flocks or fish schools, and it has been successfully applied to wide-ranging optimisation applications [41–46]. Owing to the efficiency of PSO, the tunable RBF modelling approach advocated in [39,40] offers significant advantages in terms of better generalisation performance and smaller classifier size as well as lower complexity in learning process, compared with the standard fixed kernel approach. This PSO aided tunable RBF classifier offers the state-of-the-art for balanced data sets.

Although the study [1] has shown that kernel-based methods provide a relatively robust classification to imbalanced problems, the detrimental effects of a highly imbalanced data set can seriously degrade the generalisation performance of kernel-based classifiers. In order to achieve better classification performance for highly imbalanced data, an effective approach is to integrate kernel-based classifiers with re-sampling methods. The previous studies [47–49] mainly focused on SVMs. Specifically, the method [47] combined the SMOTE with different costs to bias SVMs by assigning different classes with different costs so as to shift the decision boundary away from the positive instances and to define a better boundary. The work [48] proposed ensemble systems by re-sampling data sets to form the input to the standard SVM classifier, while the method [49] introduced asymmetric misclassification costs in SVMs so as to improve classification performance. Another integration of SVM with under-sampling method used the combination of the granular support vector machine (GSVM) [50] and repetitive under-sampling (RU) to form the GSVM–RU algorithm [51].

Against this background, this contribution proposes an effective alternative to deal with two-class imbalanced classification problems by combining the SMOTE algorithm [21] and the PSO aided RBF classifier [39,40]. Specifically, the SMOTE is first applied to generate synthetic instances in the positive class to balance the training data set. Using the resulting balanced data set, the tunable RBF classifier is then constructed by applying the PSO to minimise the LOO misclassification rate in the computationally efficient OFS procedure. In the experimental study involving a simulated imbalanced data set and three real imbalanced data sets, three benchmarks are used to compare with the proposed SMOTE+PSO-OFS method. The first benchmark combines the SMOTE [21] and the $\overline{K}$ nearest neighbour ($\overline{K}$-NN) classifier [52], which will be denoted as the SMOTE+$\overline{K}$-NN. The $\overline{K}$-NN classifier is a widely used classification method, and this combined SMOTE and $\overline{K}$-NN represents a typical method of the re-sampling approach for imbalanced problems. The second benchmark is the algorithm advocated in [26], denoted by the LOO-AUC+OFS, which is a state-of-the-art representative of the second approach for dealing with imbalanced problems. The third benchmark, the κ-means+WLSE algorithm, as discussed previously, is also a typical method of the imbalanced learning approach. The experimental results obtained demonstrate that the proposed method is competitive to these existing state-of-the-arts methods for two-class imbalanced problems.

The rest of the paper is organised as follows. Section 2 introduces the tunable RBF model for two-class classification and the OFS procedure based on the LOO misclassification rate, while Section 3 presents the PSO algorithm for tuning the RBF kernels by minimising the LOO misclassification rate. Section 4 introduces the SMOTE method and presents the proposed combined SMOTE and PSO based RBF algorithm. The effectiveness of our approach is demonstrated by numerical examples in Section 5, and our conclusions are given in Section 6.

2. RBF classifier for two-class problems

Consider the two-class data set $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$ that contains N data instances, where $y_k = \{\pm 1\}$ denotes the class label for the feature vector $\mathbf{x}_k \in \mathrm{R}^m$, while there are N_+ positive instances and N_- negative instances, with $N = N_+ + N_-$. We use the data set D_N to construct the RBF classifier of the form:

$$\hat{y}_k^{(M)} = \sum_{i=1}^{M} w_i g_i(\mathbf{x}_k) = \mathbf{g}_M^{\mathrm{T}}(k)\mathbf{w}_M$$

$$\tilde{y}_k^{(M)} = \mathrm{sgn}(\hat{y}_k^{(M)}) \tag{1}$$

where M is the number of RBF kernels, $\hat{y}_k^{(M)}$ is the output of the M-term classifier with the M kernels, $g_i(\bullet)$ for $1 \leq i \leq M$, $\mathbf{w}_M = [w_1 w_2 \cdots w_M]^{\mathrm{T}}$ is the weight vector and $\mathbf{g}_M^{\mathrm{T}}(k) = [g_1(\mathbf{x}_k)\ g_2(\mathbf{x}_k) \cdots g_M(\mathbf{x}_k)]$, while $\tilde{y}_k^{(M)}$ denotes the corresponding estimated class label for $\mathbf{x}_k$, and

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

In this study, we use the Gaussian kernel function

$$g_i(\mathbf{x})=\exp(-(\mathbf{x}-\mathbf{c}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\mathbf{c}_i)) \tag{3}$$

where $\mathbf{c}_i\in\mathrm{R}^m$ is the centre vector of the ith RBF kernel and $\boldsymbol{\Sigma}_i=\mathrm{diag}\{\sigma_{i,1}^2,\sigma_{i,2}^2,\cdots,\sigma_{i,m}^2\}$ is the diagonal covariance matrix of the ith kernel. Hence, the position of each kernel, $\mathbf{c}_i$, and the coverage of each kernel, $\boldsymbol{\Sigma}_i$, are both considered as the tunable parameters to be determined in modelling.

From (1), the RBF classifier over D_N can be written in the matrix form as

$$\mathbf{y}=\mathbf{G}_M\mathbf{w}_M+\mathbf{e}^{(M)} \tag{4}$$

where $\mathbf{e}^{(M)}=[e_1^{(M)}e_2^{(M)}\cdots e_N^{(M)}]^{\mathrm{T}}$ is the error vector with the M-term modelling error $e_k^{(M)}=y_k-\hat{y}_k^{(M)}$, $\mathbf{y}=[y_1y_2\cdots y_N]^{\mathrm{T}}$ is the desired class label vector, and the kernel matrix $\mathbf{G}_M=[\mathbf{g}_1\mathbf{g}_2\cdots\mathbf{g}_M]$ with $\mathbf{g}_l=[g_l(\mathbf{x}_1)g_l(\mathbf{x}_2)\cdots g_l(\mathbf{x}_N)]^{\mathrm{T}}$ for $1\le l\le M$. Note that $\mathbf{g}_l$ is the lth column of $\mathbf{G}_M$ while $\mathbf{g}_M^{\mathrm{T}}(k)$ is the kth row of $\mathbf{G}_M$.

Now consider the orthogonal decomposition $\mathbf{G}_M=\mathbf{P}_M\mathbf{A}_M$, where

$$\mathbf{A}_M=\begin{bmatrix}1 & a_{1,2} & \cdots & a_{1,M}\\ 0 & 1 & \ddots & \vdots\\ \vdots & \ddots & \ddots & a_{M-1,M}\\ 0 & \cdots & 0 & 1\end{bmatrix} \tag{5}$$

$$\mathbf{P}_M=[\mathbf{p}_1\ \mathbf{p}_2\cdots\mathbf{p}_M] \tag{6}$$

and the columns in (6) satisfy $\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_j=0$ for $i\ne j$. The RBF classifier (4) can alternatively be represented as

$$\mathbf{y}=\mathbf{P}_M\boldsymbol{\theta}_M+\mathbf{e}^{(M)} \tag{7}$$

where $\boldsymbol{\theta}_M=[\theta_1\ \theta_2\cdots\theta_M]^{\mathrm{T}}$ satisfies $\boldsymbol{\theta}_M=\mathbf{A_M}\mathbf{w}_M$. The space spanned by the original model bases $\mathbf{g}_i$, $1\le i\le M$, is identical to that spanned by $\mathbf{p}_i$, $1\le i\le M$.

The OFS procedure constructs the RBF kernels one by one by minimising the LOO misclassification rate [39,40]. Specifically, at the nth stage, the nth RBF kernel, namely, $\mathbf{p}_n$ and θ_n, is determined. Define the LOO-model output of the n-term RBF model constructed from the LOO data set $D_N\backslash(\mathbf{x}_k,y_k)$, calculated at $\mathbf{x}_k$, as $\hat{y}_k^{(n,-k)}$. Further define the associated LOO decision variable as

$$s_k^{(n,-k)}=\mathrm{sgn}(y_k)\hat{y}_k^{(n,-k)}=y_k\hat{y}_k^{(n,-k)} \tag{8}$$

Then the LOO misclassification rate is defined by [29]

$$J_{\mathrm{LOO}}^{(n)}=\frac{1}{N}\sum_{k=1}^{N}\mathcal{I}_d(s_k^{(n,-k)}) \tag{9}$$

in which the indicator function $\mathcal{I}_d(s)$ is defined as

$$\mathcal{I}_d(s)=\begin{cases}1, & s\le 0\\ 0, & s>0\end{cases} \tag{10}$$

The LOO misclassification rate is a measure of the classifier's generalisation capability [29,35,36,53]. By making use of Sherman–Morrison–Woodbury theorem [53] as well as the orthogonal property, the LOO decision variable can be efficiently calculated according to [29,39,40]

$$s_k^{(n,-k)}=\frac{\psi_k^{(n)}}{\eta_k^{(n)}} \tag{11}$$

in which $\psi_k^{(n)}$ and $\eta_k^{(n)}$ can be computed recursively by

$$\psi_k^{(n)}=\psi_k^{(n-1)}+y_k\theta_np_n(k)-\frac{p_n^2(k)}{\mathbf{p}_n^{\mathrm{T}}\mathbf{p}_n+\lambda} \tag{12}$$

$$\eta_k^{(n)}=\eta_k^{(n-1)}-\frac{p_n^2(k)}{\mathbf{p}_n^{\mathrm{T}}\mathbf{p}_n+\lambda} \tag{13}$$

where $p_n(k)$ is the kth element of $\mathbf{p}_n$, and $\lambda\ge 0$ is a small regularisation parameter if the regularisation is employed.

At the nth stage of the OFS procedure, the nth RBF kernel, namely, its centre vector $\mathbf{c}_n$ and diagonal covariance matrix $\boldsymbol{\Sigma}_n$, are determined by minimising $J_{\mathrm{LOO}}^{(n)}$. The construction terminates at the size of M when $J_{\mathrm{LOO}}^{(M+1)}\ge J_{\mathrm{LOO}}^{(M)}$ [29,39,40].

3. PSO for optimising RBF parameters

Denote $\boldsymbol{\mu}=[\mu(1)\ \mu(2)\cdots\mu(2m)]^{\mathrm{T}}$ as the $2m$-dimensional parameter vector that contains $\mathbf{c}_n$ and $\boldsymbol{\Sigma}_n$. Then, as defined in the previous section, the problem of determining the nth RBF kernel's parameters at the nth OFS stage is to solve the following optimisation problem

$$\hat{\boldsymbol{\mu}}=\arg\min_{\boldsymbol{\mu}\in\boldsymbol{\Gamma}}J_{\mathrm{LOO}}^{(n)}(\boldsymbol{\mu}) \tag{14}$$

where the $2m$-dimensional search space $\boldsymbol{\Gamma}$ is defined by

$$\boldsymbol{\Gamma}\triangleq\prod_{i=1}^{2m}[\Gamma_{i,\min},\Gamma_{i,\max}] \tag{15}$$

Specifically, the search space for $\mathbf{c}_n=[c_{n,1}c_{n,2}\cdots c_{n,m}]^{\mathrm{T}}$ is specified by the distribution of the training data $\{\mathbf{x}_k=[x_{k,1}\ x_{k,2}\cdots x_{k,m}]^{\mathrm{T}}\}_{k=1}^{N}$, namely,

$$c_{n,i}\in[x_{\min,i},x_{\max,i}]\triangleq[\Gamma_{i,\min},\Gamma_{i,\max}] \tag{16}$$

for $1\le i\le m$, with

$$\begin{cases}x_{\min,i}=\min\limits_{1\le k\le N}x_{k,i}\\ x_{\max,i}=\max\limits_{1\le k\le N}x_{k,i}\end{cases} \tag{17}$$

while each element of $\boldsymbol{\Sigma}_n$ is limited in the range

$$\sigma_{n,i}^2\in[\sigma_{\min}^2,\sigma_{\max}^2]\triangleq[\Gamma_{(i+m),\min},\Gamma_{(i+m),\max}] \tag{18}$$

for $1\le i\le m$. When applying a PSO [38] to solve the optimisation (14), a swarm of the candidate particles $\{\boldsymbol{\mu}_i^{[l]}\}_{i=1}^{S}$ are "flying" in the search space $\boldsymbol{\Gamma}$ in order to find a solution $\hat{\boldsymbol{\mu}}$, where S is the size of the swarm and $l\in\{0,1,\cdots,L\}$ denotes the lth movement of the swarm. Each particle $\boldsymbol{\mu}$ has a $2m$-dimensional velocity $\boldsymbol{v}=[v(1)\ v(2)\cdots v(2m)]^{\mathrm{T}}$ to direct its search, and $\boldsymbol{v}\in\mathbf{V}$ with the velocity space defined by

$$\mathbf{V}\triangleq\prod_{i=1}^{2m}[-V_{i,\max},V_{i,\max}] \tag{19}$$

where $V_{i,\max}=\frac{1}{2}(\Gamma_{i,\max}-\Gamma_{i,\min})$.

To start the PSO, the candidate particles $\{\boldsymbol{\mu}_i^{[0]}\}_{i=1}^{S}$ are initialised randomly within $\boldsymbol{\Gamma}$, and the velocity for each candidate particle is initialised to zero, namely, $\{\boldsymbol{v}_i^{[0]}=\mathbf{0}\}_{i=1}^{S}$. The cognitive information $\boldsymbol{pb}_i^{[l]}$ and the social information $\boldsymbol{gb}^{[l]}$ record the best position visited by the particle i and the best position visited by the entire swarm, respectively, during the l movements. The LOO costs associated with $\boldsymbol{pb}_i^{[l]}$ and $\boldsymbol{gb}^{[l]}$ are denoted by $J_{\mathrm{LOO}}^{(n)}(\boldsymbol{pb}_i^{[l]})$ and $J_{\mathrm{LOO}}^{(n)}(\boldsymbol{gb}^{[l]})$, respectively. The cognitive information $\boldsymbol{pb}_i^{[l]}$ and the social information $\boldsymbol{gb}^{[l]}$ are used to update the velocities and positions according to

$$\boldsymbol{v}_i^{[l+1]} = a \cdot \boldsymbol{v}_i^{[l]} + rand() \cdot b \cdot (\boldsymbol{pb}_i^{[l]} - \boldsymbol{\mu}_i^{[l]}) + rand() \cdot c \cdot (\boldsymbol{gb}^{[l]} - \boldsymbol{\mu}_i^{[l]}) \quad (20)$$

$$\boldsymbol{\mu}_i^{[l+1]} = \boldsymbol{\mu}_i^{[l]} + \boldsymbol{v}_i^{[l+1]} \quad (21)$$

where a denotes the inertia weight, $rand()$ is the random number uniformly distributed in [0, 1], b and c are the two acceleration coefficients. Experimental results given in [40] show that a better performance can be achieved by using $a = rand()$ instead of a constant inertia weight. Adopting the time varying acceleration coefficients (TVAC) [41], in which

$$b = 2.5 - (2.5 - 0.5) \cdot l/L$$

$$c = 0.5 + (2.5 - 0.5) \cdot l/L \quad (22)$$

can often enhance the performance of PSO. The search space $\boldsymbol{\Gamma}$ and the velocity space $\mathbf{V}$ are used to confine $\boldsymbol{\mu}_i^{[l+1]}$ and $\boldsymbol{v}_i^{[l+1]}$ derived from (20) and (21), respectively. If $\boldsymbol{v}_i^{[l+1]}$ becomes too close to $\mathbf{0}$, a random re-initialisation is needed, which may take the form $\boldsymbol{v}_i^{[l+1]} = \pm 0.1 \cdot rand() \cdot \mathbf{V}_{\max}$, where $\mathbf{V}_{\max} = [V_{1,\max} V_{2,\max} \cdots V_{2m,\max}]^{\mathrm{T}}$. The detailed PSO aided OFS algorithm can be found in [40], also see the next section.

4. Combined SMOTE and PSO optimised RBF for imbalanced classification

The SMOTE [21] over-samples the positive class by creating synthetic instances by a specified over-sampling ratio of the original minority data size, $\beta\%$. Based on each minority data sample, denoted by $\mathbf{x}_o$, $\beta\%$ synthetic data points are generated by randomly selecting data points on the lines linking $\mathbf{x}_o$ with some of its K nearest neighbours, where K is predetermined. Depending on the required SMOTE amount $\beta\%$, one out of the K nearest positive-class data samples are randomly selected several times. For example, if $\beta\% = 600\%$ and $K=5$, then one out of five nearest neighbours of $\mathbf{x}_o$ is randomly chosen repeatedly for six times. Each time a random kth neighbour is selected to create a line linking $\mathbf{x}_o$ to this neighbour, and then a single synthetic instance is created by randomly selecting a point on the line. Thus any synthetic instance $\mathbf{x}_s$ is given by

$$\mathbf{x}_s = \mathbf{x}_o + \delta \cdot (\mathbf{x}_o^{\{t\}} - \mathbf{x}_o) \quad (23)$$

where $\mathbf{x}_s$ denotes one synthetic instance, $x_o^{\{t\}}$ is the tth nearest neighbors of $\mathbf{x}_o$ in the positive class, and $\delta \in [0,1]$ is a random number. The procedure is repeated for all the positive data points.

A major problem caused by imbalanced data sets is that most classifiers tend to attribute the positive-class instances within the decision region to the negative class, due to insufficient positive-class training instances in the decision region. As a result, the trained decision boundary tends to be far away from the negative class and too close to the positive class. The contribution of SMOTE is to enhance the significance of the small and specific region belonging to the positive class in the decision region, which leads to the better generalisation for classifiers. Fig. 1(a) shows a simulated imbalanced data set, the details of which are specified in Section 5. After the SMOTE over-sampling the positive class by 500% of its original size, the instances from the positive class become more significant in the decision region (the area specified by dash–dot line), as shown in Fig. 1(b), compared with the original data set. Consequently, the trained decision boundary tends to be further away from the positive class.

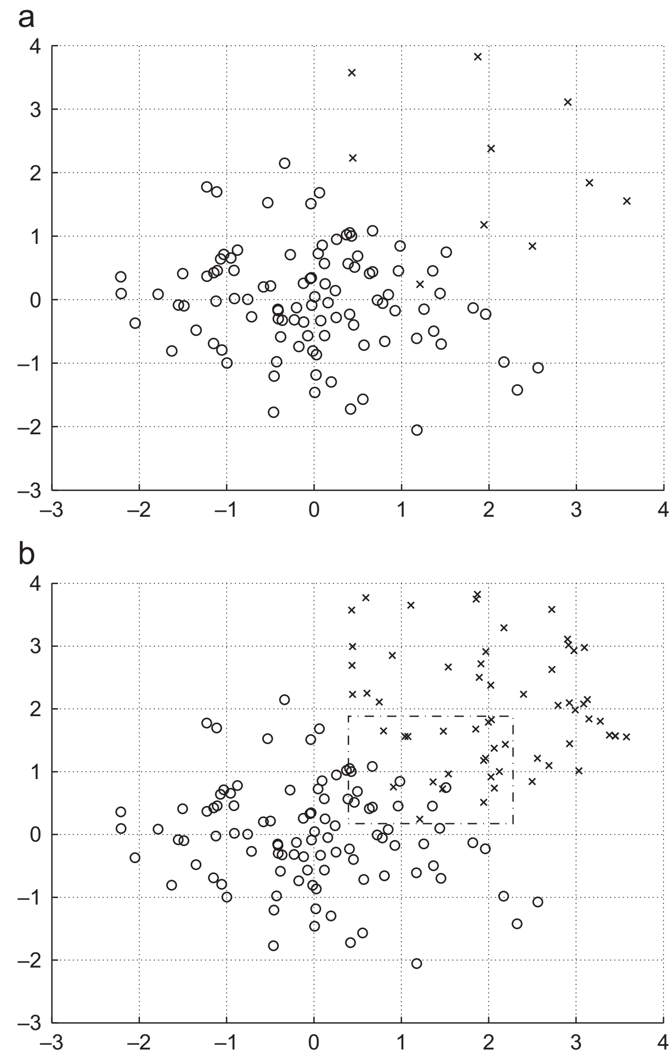

Fig. 1. Simulated 2-dimensional example: (a) original training data space, and (b) training data space after SMOTE over-sampling the positive class by 500% of its original size, where x denotes a positive-class instance while ○ denotes a negative class instance.

We combine this SMOTE with the PSO optimised RBF classifier described in Section 3 to create a powerful algorithm for two-class imbalanced problems. This combined SMOTE and PSO aided RBF is detailed below.

1. Over-sampling the training data set:
   (a) SMOTE initialisation: Specify the balanced degree $\beta\%$ and the value of K.
   (b) Create the new training data set $\tilde{D}_N$ by appending the generated positive training data points to the original training data set via the SMOTE.
2. PSO aided OFS initialisation:
   (a) Specify the search space $\boldsymbol{\Gamma}$ and the velocity space $\mathbf{V}$. Specify the values of L and S.
   (b) Set $J_{\mathrm{LOO}}^{(0)} = 1$, $\psi_k^{(0)} = 0$, and $\eta_k^{(0)} = 1$.
   (c) Set regularisation parameter $\lambda = 10^{-6}$.
3. Construct the nth RBF kernel:
   (a) PSO initialisation: Randomly initialise $\{\boldsymbol{\mu}_i^{[0]}\}_{i=1}^S$ in $\boldsymbol{\Gamma}$, and set $\{\boldsymbol{v}_i^{[0]} = \mathbf{0}\}_{i=1}^S$.
   (b) For $0 \le l < L$:

(b.i) Construct the candidates $\mathbf{g}_n^{\{i\}}$ from $\boldsymbol{\mu}_i^{[l]}$, for $1 \le i \le S$. Then, for $1 \le i \le S$ and $1 \le j < n$, compute:

$$a_{j,n}^{\{i\}} = \begin{cases} 1, & n = 1 \\ \dfrac{\mathbf{p}_j^{\mathrm{T}} \mathbf{g}_n^{\{i\}}}{\mathbf{p}_j^{\mathrm{T}} \mathbf{p}_j}, & n > 1 \end{cases}$$

$$\mathbf{p}_n^{\{i\}} = \begin{cases} \mathbf{g}_n^{\{i\}}, & n = 1 \\ \mathbf{g}_n^{\{i\}} - \sum_{j=1}^{n-1} a_{j,n}^{\{i\}} \mathbf{p}_j, & n > 1 \end{cases}$$

$$\theta_n^{\{i\}} = \frac{(\mathbf{p}_n^{\{i\}})^{\mathrm{T}} \mathbf{y}}{(\mathbf{p}_n^{\{i\}})^{\mathrm{T}} \mathbf{p}_n^{\{i\}} + \lambda}$$

(b.ii) For $1 \le i \le S$ and $1 \le k \le N$, compute:

$$\psi_k^{(n)}\{i\} = \psi_k^{(n-1)} + y(k)\theta_n^{\{i\}} p_n^{\{i\}}(k) - \frac{(p_n^{\{i\}}(k))^2}{(\mathbf{p}_n^{\{i\}})^{\mathrm{T}} \mathbf{p}_n^{\{i\}} + \lambda}$$

$$\eta_k^{(n)}\{i\} = \eta_k^{(n-1)} - \frac{(p_n^{\{i\}}(k))^2}{(\mathbf{p}_n^{\{i\}})^{\mathrm{T}} \mathbf{p}_n^{\{i\}} + \lambda}$$

Then, for $1 \le i \le S$, calculate the LOO costs:

$$J_{\mathrm{LOO}}^{(n)}\{i\} = \frac{1}{N} \sum_{k=1}^{N} \mathcal{I}_d \left(\frac{\psi_k^{(n)}\{i\}}{\eta_k^{(n)}\{i\}} \right)$$

(b.iii) For $1 \le i \le S$:
If $J_{\mathrm{LOO}}^{(n)}\{i\} < J_{\mathrm{LOO}}^{(n)}(\boldsymbol{pb}_i^{[l]})$:

$\boldsymbol{pb}_i^{[l]} = \boldsymbol{\mu}_i^{[l]}$ and $J_{\mathrm{LOO}}^{(n)}(\boldsymbol{pb}_i^{[l]}) = J_{\mathrm{LOO}}^{(n)}\{i\}$

Then find

$$i_* = \arg \min_{1 \le i \le S} J_{\mathrm{LOO}}^{(n)}(\boldsymbol{pb}_i^{[l]})$$

If $J_{\mathrm{LOO}}^{(n)}(\boldsymbol{pb}_{i_*}^{[l]}) < J_{\mathrm{LOO}}^{(n)}(\boldsymbol{gb}^{[l]})$:

$\boldsymbol{gb}^{[l]} = \boldsymbol{pb}_{i_*}^{[l]}$ and $J_{\mathrm{LOO}}^{(n)}(\boldsymbol{gb}^{[l]}) = J_{\mathrm{LOO}}^{(n)}(\boldsymbol{pb}_{i_*}^{[l]})$

(b.iv) For $1 \le i \le S$:

$$\begin{aligned} \boldsymbol{v}_i^{[l+1]} &= a \cdot \boldsymbol{v}_i^{[l]} + rand() \cdot b \cdot (\boldsymbol{pb}_i^{[l]} - \boldsymbol{\mu}_i^{[l]}) \\ &\quad + rand() \cdot c \cdot (\boldsymbol{gb}^{[l]} - \boldsymbol{\mu}_i^{[l]}) \end{aligned}$$

If $v_i^{[l+1]}(j) = 0$:

$v_i^{[l+1]}(j) = \pm 0.1 \cdot rand() \cdot V_{j,\max}$

If $v_i^{[l+1]}(j) > V_{j,\max}$: $v_i^{[l+1]}(j) = V_{j,\max}$
If $v_i^{[l+1]}(j) < -V_{j,\max}$: $v_i^{[l+1]}(j) = -V_{j,\max}$
Then for $1 \le i \le S$:

$\boldsymbol{\mu}_i^{[l+1]} = \boldsymbol{\mu}_i^{[l]} + \boldsymbol{v}_i^{[l+1]}$

If $\mu_i^{[l+1]}(j) > \Gamma_{j,\max}$: $\mu_i^{[l+1]}(j) = \Gamma_{j,\max}$
If $\mu_i^{[l+1]}(j) < \Gamma_{j,\min}$: $\mu_i^{[l+1]}(j) = \Gamma_{j,\min}$

(c) Termination of PSO: $\boldsymbol{gb}^{[L]}$ provides $\mathbf{c}_n$ and $\boldsymbol{\Sigma}_n$ with the associated LOO cost $J_{\mathrm{LOO}}^{(n)} = J_{\mathrm{LOO}}^{(n)}(\boldsymbol{gb}^{[L]})$.
The algorithm also generates $a_{j,n}$ for $1 \le j < n$, $\mathbf{p}_n$ and θ_n as well as $\psi_k^{(n)}$ and $\eta_k^{(n)}$ for $1 \le k \le N$.

4. OFS termination condition checking:
If $J_{\mathrm{LOO}}^{(n)} < J_{\mathrm{LOO}}^{(n-1)}$: $n = n+1$, go to step 3.
Otherwise, $M = n-1$, terminate the OFS procedure.

5. Experimental results

The effectiveness of the proposed SMOTE+PSO-OFS algorithm was investigated using a simulated imbalanced date set and three real imbalanced data sets. The first two real data sets were taken from [54], while the third real data set was from [55]. These three real data sets were chosen in the order of increasing imbalance. For each data set, the positive class was over-sampled at different rates $\beta\%$ of its original size using the SMOTE. For the synthetic data set, a separate test data set was used, while for the three real data sets, P-fold cross validation was used, to indicate the classifier generalisation capability based on multiple specifications, including the true positive rate (TP%) and the false positive rate (FP%) [56], as well as the precision (Pr), the F-measure (F-meas) and the G-mean [57]. These criteria are commonly adopted as the performance metrics for evaluating imbalanced learning classifiers. They were calculated according to the confusion matrix in Table 1 as follows:

$$\mathrm{TP\%} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{24}$$

$$\mathrm{FP\%} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{25}$$

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{26}$$

$$G\text{-mean} = \sqrt{\mathrm{TP\%} \times (1-\mathrm{FP\%})} \tag{27}$$

$$F\text{-meas} = \frac{2 \times \mathrm{Pr} \times \mathrm{TP\%}}{\mathrm{Pr}+\mathrm{TP\%}} \tag{28}$$

As discussed in the introduction section, the three typical methods that represent the two different approaches for dealing with imbalanced problems, respectively, were used as the benchmarks for comparison, and they were the SMOTE+$\overline{K}$-NN with $\overline{K} = 1$ and 3, as well as the LOO-AUC+OFS with different weight ρ and the κ-means +WLSE with different weight ρ. Note that in the SMOTE+$\overline{K}$-NN classifiers if there is any data sample in the test data set that duplicates a data sample in the training data set, this was not counted in the statistics in order to obtain honest cross validation. This is necessary in particular for ADI data sets which are produced by randomly sampling the original data set, causing repetitive data samples. For the κ-means +WLSE algorithm a fixed common variance for every kernel was predetermined empirically (similar to [26]), and in addition the number of centres were also predetermined empirically.

Simulated imbalanced data set: The simulated data set was generated with the m=2 features. The mean vector of the negative class was $[0\ 0]^{\mathrm{T}}$, while the mean vector of the positive class was $[2\ 2]^{\mathrm{T}}$. The covariance matrices of both the negative-class and positive-class instances were the same 2-dimensional identity matrix. The training data set contained 100 instances from the negative class and 10 instances from the positive class, as depicted in Fig. 1(a). The test data set contained 1000 instances from the negative class, and 100 instances from the positive class. The 5-nearest neighbour method was applied to generate synthetic training data in the SMOTE, with the over-sampling rate $\beta\%$ set to 0%, 100%, 500%, 1000%, 1500% and 2000%, respectively. For the SMOTE+PSO-OFS, the swarm size and the number of

Table 1
Confusion matrix.

	Predicted positive	Predicted negative
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

movements were set to $S=10$ and $L=20$. The test results obtained by the various classifiers are shown in Table 2.

It can be seen from the results for the SMOTE+PSO-OFS listed in Table 2 that, as the over-sampling rate $\beta\%$ increases, typically TP% increases but FP% inevitably increases as well. A better tradeoff between TP% and FP% was achieved, however, at the over-sampling rate where the better *G*-mean and *F*-measure were obtained. Since the imbalance degree of the negative class to the positive class was 10:1, the over-sampled positive instances made $\tilde{D}_N$ fully balanced at $\beta\%=1000\%$. From Table 2, it can be seen that the best test performance tradeoff occurred at the over-sampling rate around 500–1000%. Compared with the other benchmark methods, the proposed SMOTE+PSO-OFS showed a competitive test performance. The effect of the SMOTE on the decision boundary is shown in Fig. 2, where it can be seen that the decision boundary trained by the more balanced data set was pushed further away from the positive class.

Table 2
Test classification performance comparison for the synthetic data set.

Method	TP%	FP%	Pr	*G*-mean	*F*-meas
SMOTE+1-NN ($\beta\%=0\%$)	0.830	0.047	0.638	0.899	0.722
SMOTE+1-NN ($\beta\%=100\%$)	0.880	0.094	0.484	0.893	0.624
SMOTE+1-NN ($\beta\%=500\%$)	0.920	0.113	0.449	0.903	0.603
SMOTE+1-NN ($\beta\%=1000\%$)	0.930	0.156	0.373	0.886	0.533
SMOTE+1-NN ($\beta\%=1500\%$)	0.940	0.158	0.373	0.890	0.534
SMOTE+1-NN ($\beta\%=2000\%$)	0.930	0.150	0.383	0.889	0.542
SMOTE+3-NN ($\beta\%=0\%$)	0.780	**0.022**	**0.780**	0.873	0.780
SMOTE+3-NN ($\beta\%=100\%$)	0.900	0.092	0.495	0.904	0.638
SMOTE+3-NN ($\beta\%=500\%$)	0.940	0.134	0.412	0.902	0.573
SMOTE+3-NN ($\beta\%=1000\%$)	**0.950**	0.156	0.378	0.895	0.541
SMOTE+3-NN ($\beta\%=1500\%$)	**0.950**	0.151	0.386	0.898	0.549
SMOTE+3-NN ($\beta\%=2000\%$)	**0.950**	0.174	0.353	0.886	0.515
LOO-AUC+OFS ($\rho=1$)	0.860	0.049	0.637	0.904	0.732
LOO-AUC+OFS ($\rho=5$)	0.840	0.028	0.750	0.903	0.792
LOO-AUC+OFS ($\rho=10$)	0.900	0.063	0.588	**0.918**	0.712
LOO-AUC+OFS ($\rho=15$)	0.870	0.046	0.654	0.911	0.747
LOO-AUC+OFS ($\rho=20$)	0.870	0.049	0.640	0.909	0.737
κ-means +WLSE ($\rho=1$)	0.810	0.030	0.730	0.886	0.768
κ-means +WLSE ($\rho=5$)	0.840	0.041	0.672	0.898	0.747
κ-means+WLSE ($\rho=10$)	0.860	0.078	0.524	0.890	0.652
κ-means +WLSE ($\rho=15$)	0.940	0.131	0.418	0.904	0.578
κ-means +WLSE ($\rho=20$)	**0.950**	0.185	0.339	0.880	0.500
SMOTE+PSO-OFS ($\beta\%=0\%$)	0.860	0.044	0.662	0.907	0.748
SMOTE+PSO-OFS ($\beta\%=100\%$)	0.880	0.055	0.615	0.912	0.724
SMOTE+PSO-OFS ($\beta\%=500\%$)	0.810	0.023	**0.780**	0.890	**0.794**
SMOTE+PSO-OFS ($\beta\%=1000\%$)	0.890	0.053	0.627	**0.918**	0.736
SMOTE+PSO-OFS ($\beta\%=1500\%$)	0.930	0.102	0.476	0.914	0.631
SMOTE+PSO-OFS ($\beta\%=2000\%$)	0.940	0.110	0.461	0.915	0.618

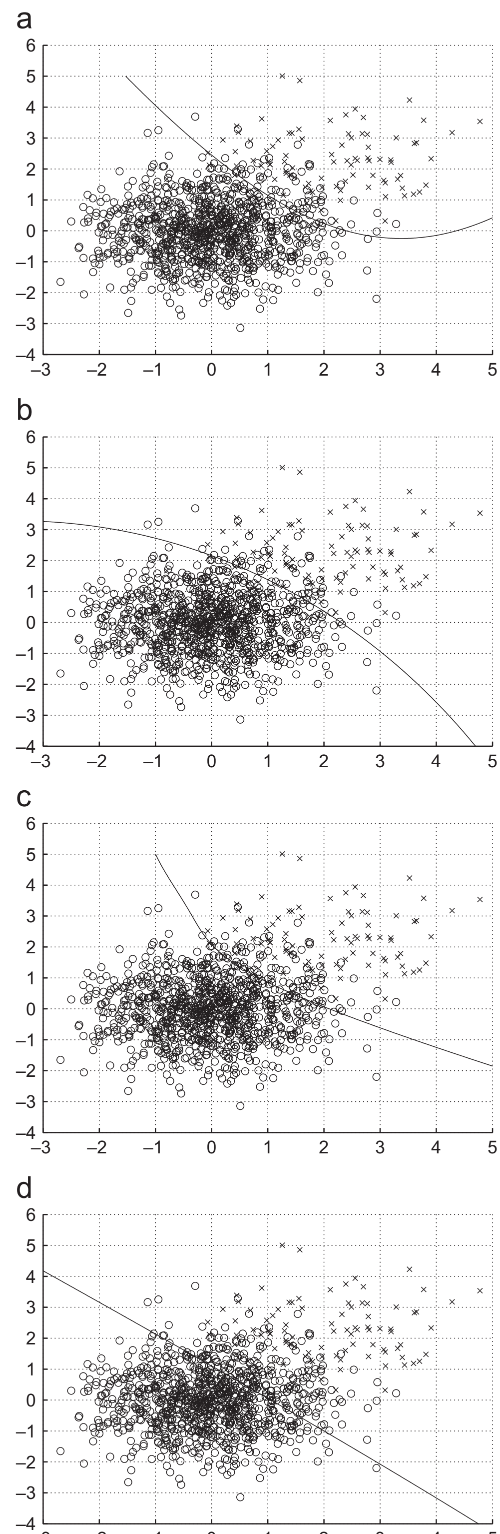

Fig. 2. Decision boundaries obtained by the SMOTE+PSO-OFS with different over-sampling rates for the simulated 2-dimensional example: (a) $\beta\%=0\%$, (b) $\beta\%=100\%$, (c) $\beta\%=1000\%$, and (d) $\beta\%=2000\%$, where x denotes a positive-class test instance while ○ denotes a negative-class test instance.

Pima Indians diabetes data set: The data set was obtained from the UCI repository [54], and it contained 768 instances from the two classes with 500 negative instances and 268 positive instances. This data set was used to identify the positive diabetes cases in a population near Phoenix, Arizona. The feature space dimension was $m=8$. All the eight input features were normalised to the range [0, 1] using the operation

$$\bar{x}_{k,i} = \frac{x_{k,i} - x_{\min,i}}{x_{\max,i} - x_{\min,i}}, \quad 1 \le k \le N, \ 1 \le i \le m \tag{29}$$

The 5-nearest neighbour scheme was applied to generate synthetic training data in the SMOTE. The over-sampling rate $\beta\%$ was

Table 3
Eight-fold cross validation classification performance and standard deviations for Pima Indians diabetes data set.

Method	TP%	FP%	Pr	*G*-mean	*F*-meas
SMOTE+1-NN ($\beta\%=0\%$)	0.54 ± 0.04	0.21 ± 0.04	0.58 ± 0.06	0.65 ± 0.02	0.56 ± 0.04
SMOTE+1-NN ($\beta\%=50\%$)	0.58 ± 0.06	0.24 ± 0.04	0.56 ± 0.07	0.66 ± 0.03	0.57 ± 0.05
SMOTE+1-NN ($\beta\%=75\%$)	0.59 ± 0.06	0.25 ± 0.04	0.56 ± 0.07	0.66 ± 0.02	0.57 ± 0.06
SMOTE+1-NN ($\beta\%=100\%$)	0.63 ± 0.06	0.27 ± 0.05	0.55 ± 0.08	0.67 ± 0.02	0.58 ± 0.05
SMOTE+1-NN ($\beta\%=150\%$)	0.66 ± 0.05	0.27 ± 0.05	0.57 ± 0.07	0.70 ± 0.03	0.61 ± 0.05
SMOTE+1-NN ($\beta\%=200\%$)	0.68 ± 0.07	0.28 ± 0.04	0.56 ± 0.07	0.70 ± 0.04	0.61 ± 0.06
SMOTE+1-NN ($\beta\%=250\%$)	0.70 ± 0.04	0.30 ± 0.04	0.55 ± 0.07	0.70 ± 0.03	0.61 ± 0.04
SMOTE+1-NN ($\beta\%=500\%$)	0.72 ± 0.04	0.36 ± 0.07	0.52 ± 0.09	0.68 ± 0.04	0.60 ± 0.05
SMOTE+3-NN ($\beta\%=0\%$)	0.58 ± 0.06	0.17 ± 0.06	0.65 ± 0.07	0.69 ± 0.04	0.61 ± 0.04
SMOTE+3-NN ($\beta\%=50\%$)	0.62 ± 0.07	0.19 ± 0.05	0.63 ± 0.05	0.70 ± 0.04	0.62 ± 0.04
SMOTE+3-NN ($\beta\%=75\%$)	0.67 ± 0.07	0.25 ± 0.05	0.59 ± 0.06	0.71 ± 0.04	0.63 ± 0.06
SMOTE+3-NN ($\beta\%=100\%$)	0.70 ± 0.08	0.29 ± 0.05	0.56 ± 0.05	0.71 ± 0.05	0.62 ± 0.04
SMOTE+3-NN ($\beta\%=150\%$)	0.74 ± 0.08	0.30 ± 0.04	0.56 ± 0.05	0.72 ± 0.04	0.64 ± 0.04
SMOTE+3-NN ($\beta\%=200\%$)	0.76 ± 0.09	0.33 ± 0.06	0.55 ± 0.06	0.71 ± 0.05	0.64 ± 0.06
SMOTE+3-NN ($\beta\%=250\%$)	0.78 ± 0.07	0.34 ± 0.05	0.55 ± 0.07	0.72 ± 0.03	0.64 ± 0.06
SMOTE+3-NN ($\beta\%=500\%$)	0.82 ± 0.06	0.41 ± 0.05	0.52 ± 0.07	0.70 ± 0.03	0.63 ± 0.06
LOO-AUC+OFS ($\rho=1.0$)	0.58 ± 0.03	0.13 ± 0.05	0.70 ± 0.09	0.71 ± 0.03	0.63 ± 0.05
LOO-AUC+OFS ($\rho=1.5$)	0.68 ± 0.06	0.20 ± 0.07	0.65 ± 0.08	0.73 ± 0.04	0.66 ± 0.05
LOO-AUC+OFS ($\rho=2.0$)	0.73 ± 0.05	0.24 ± 0.07	0.62 ± 0.07	0.74 ± 0.04	0.67 ± 0.05
LOO-AUC+OFS ($\rho=2.5$)	0.77 ± 0.05	0.31 ± 0.06	0.57 ± 0.05	0.73 ± 0.03	0.66 ± 0.07
κ-means+WLSE ($\rho=1.0$)	0.60 ± 0.06	0.13 ± 0.05	0.72 ± 0.07	0.72 ± 0.04	0.65 ± 0.03
κ-means+WLSE ($\rho=1.5$)	0.70 ± 0.08	0.20 ± 0.07	0.65 ± 0.07	0.74 ± 0.06	0.67 ± 0.05
κ-means+WLSE ($\rho=2.0$)	0.77 ± 0.07	0.29 ± 0.07	0.59 ± 0.07	0.74 ± 0.06	0.66 ± 0.06
κ-means+WLSE ($\rho=2.5$)	0.84 ± 0.05	0.34 ± 0.07	0.57 ± 0.06	0.75 ± 0.06	0.68 ± 0.06
SMOTE+PSO-OFS ($\beta\%=0\%$)	0.57 ± 0.04	$\mathbf{0.11}\pm0.04$	$\mathbf{0.73}\pm0.10$	0.71 ± 0.03	0.64 ± 0.06
SMOTE+PSO-OFS ($\beta\%=50\%$)	0.70 ± 0.07	0.19 ± 0.09	0.67 ± 0.07	0.75 ± 0.03	0.68 ± 0.04
SMOTE+PSO-OFS ($\beta\%=75\%$)	0.73 ± 0.12	0.23 ± 0.19	0.68 ± 0.14	0.73 ± 0.06	0.69 ± 0.04
SMOTE+PSO-OFS ($\beta\%=100\%$)	0.79 ± 0.07	0.25 ± 0.10	0.64 ± 0.06	$\mathbf{0.76}\pm0.05$	$\mathbf{0.70}\pm0.04$
SMOTE+PSO-OFS ($\beta\%=150\%$)	0.81 ± 0.07	0.29 ± 0.09	0.60 ± 0.06	$\mathbf{0.76}\pm0.04$	0.69 ± 0.05
SMOTE+PSO-OFS ($\beta\%=200\%$)	0.83 ± 0.04	0.33 ± 0.07	0.58 ± 0.06	0.75 ± 0.04	0.68 ± 0.05
SMOTE+PSO-OFS ($\beta\%=250\%$)	0.85 ± 0.07	0.35 ± 0.07	0.57 ± 0.07	0.74 ± 0.06	0.68 ± 0.06
SMOTE+PSO-OFS ($\beta\%=500\%$)	$\mathbf{0.91}\pm0.05$	0.44 ± 0.06	0.52 ± 0.05	0.71 ± 0.04	0.67 ± 0.05

set to 0%, 50%, 75%, 100%, 150%, 200%, 250% and 500%, respectively. The swarm size and the number of movements were set to $S=10$ and $L=20$ for the PSO. The 8-fold cross validation was used to investigate the test performance of a classifier. The 8-fold cross validation results for the various classifiers are shown in Table 3.

For the SMOTE+PSO-OFS, it can be seen that the best TP%, that is, the best detection capability for diabetes, occurred at $\beta\%=500\%$, while the best FP% occurred at $\beta\%=0\%$. But the best TP% was obtained at the expense of the worst FP%, and the best FP% was obtained at the expense of the worst TP%, as indicated by the poor values of the *G*-mean and *F*-measure. The best tradeoff between TP% and FP% occurred around $\beta\%=100-150\%$, which enabled to detect as many positive diabetes patients as possible while ensuring the minimum incorrect diagnose of healthy people. As expected, this best over-sampling rate made the enlarged data set fully balanced. The results of Table 3 also show that the test performance of the proposed SMOTE+PSO-OFS compare favourably with the other classifiers.

Haberman survival data set: This data set in the UCI repository [54] contained 306 instances from the two classes with 225 negative instances and 81 positive instances. It came from a study on the survival of patients after surgery for breast cancer. The feature space dimension was $m=3$. All the three input features were normalised to the range [0, 1] using the operation (29). The 5-nearest neighbour method was adopted to generate synthetic training data in the SMOTE. The over-sampling rate $\beta\%$ was set to 0%, 100%, 200%, 300% and 400%, respectively. The swarm size and the number of movements were chosen to be $S=10$ and $L=20$. The 3-fold cross validation was used to calculate test performance, and the results obtained for the various classifiers are shown in Table 4. Compared with the other benchmark classifiers, the SMOTE+PSO-OFS demonstrated its competitive performance. For the SMOTE+PSO-OFS, the best tradeoff between TP% and FP% occurred around $\beta\%=150\%$, which was again close to the imbalanced degree of the original data set.

ADI data set: The austempered ductile iron (ADI) material data set was obtained from a study on fatigue cracks from the graphite nodules within the microstructure in an automotive camshaft application [55]. This two-class data set contained 2923 instances in the feature space of dimension $m=9$, with 2807 negative instances and 116 positive instances. As in [55,26], 700 negative-class instances and 90 positive-class instances were randomly selected from the original data set to form the 8-fold cross validation set. Initially, all the nine input features were

Table 4
Three-fold cross validation classification performance and standard deviations for Haberman survival data set.

Method	TP%	FP%	Pr	*G*-mean	*F*-meas
SMOTE+1-NN ($\beta\%=0\%$)	0.28 ± 0.06	0.20 ± 0.13	0.38 ± 0.12	0.47 ± 0.02	0.31 ± 0.02
SMOTE+1-NN ($\beta\%=100\%$)	0.36 ± 0.11	0.22 ± 0.16	0.41 ± 0.12	0.52 ± 0.03	0.36 ± 0.01
SMOTE+1-NN ($\beta\%=200\%$)	0.41 ± 0.13	0.28 ± 0.14	0.36 ± 0.05	0.53 ± 0.04	0.37 ± 0.04
SMOTE+1-NN ($\beta\%=300\%$)	0.44 ± 0.29	0.27 ± 0.21	0.40 ± 0.09	0.53 ± 0.08	0.38 ± 0.09
SMOTE+1-NN ($\beta\%=400\%$)	0.47 ± 0.20	0.29 ± 0.22	0.40 ± 0.11	0.55 ± 0.02	0.40 ± 0.04
SMOTE+3-NN ($\beta\%=0\%$)	0.30 ± 0.10	0.15 ± 0.10	0.45 ± 0.14	0.49 ± 0.08	0.35 ± 0.09
SMOTE+3-NN ($\beta\%=100\%$)	0.41 ± 0.07	0.21 ± 0.14	0.45 ± 0.13	0.56 ± 0.04	0.41 ± 0.06
SMOTE+3-NN ($\beta\%=200\%$)	0.49 ± 0.15	0.24 ± 0.18	0.46 ± 0.10	0.60 ± 0.03	0.45 ± 0.04
SMOTE+3-NN ($\beta\%=300\%$)	0.53 ± 0.19	0.28 ± 0.20	0.44 ± 0.09	0.60 ± 0.04	0.46 ± 0.05
SMOTE+3-NN ($\beta\%=400\%$)	0.56 ± 0.16	0.31 ± 0.21	0.43 ± 0.09	0.60 ± 0.02	0.46 ± 0.01
LOO-AUC+OFS ($\rho=1$)	0.21 ± 0.02	0.05 ± 0.01	0.61 ± 0.05	0.45 ± 0.02	0.31 ± 0.03
LOO-AUC+OFS ($\rho=2$)	0.38 ± 0.08	0.13 ± 0.02	0.51 ± 0.02	0.57 ± 0.05	0.44 ± 0.06
LOO-AUC+OFS ($\rho=3$)	0.62 ± 0.08	0.27 ± 0.03	0.45 ± 0.05	0.67 ± 0.05	0.52 ± 0.06
LOO-AUC+OFS ($\rho=4$)	0.67 ± 0.02	0.42 ± 0.08	0.36 ± 0.03	0.62 ± 0.03	0.47 ± 0.02
κ-means+WLSE ($\rho=1$)	0.20 ± 0.02	$\mathbf{0.02}\pm0.00$	$\mathbf{0.63}\pm0.05$	0.44 ± 0.02	0.30 ± 0.03
κ-means+WLSE ($\rho=2$)	0.36 ± 0.06	0.05 ± 0.01	0.46 ± 0.03	0.58 ± 0.04	0.40 ± 0.04
κ-means+WLSE ($\rho=3$)	0.49 ± 0.03	0.10 ± 0.01	0.39 ± 0.02	0.67 ± 0.02	0.44 ± 0.01
κ-means+WLSE ($\rho=4$)	0.56 ± 0.04	0.14 ± 0.01	0.34 ± 0.01	$\mathbf{0.69}\pm0.02$	0.42 ± 0.01
SMOTE+PSO-OFS ($\beta\%=0\%$)	0.23 ± 0.04	0.07 ± 0.06	0.57 ± 0.01	0.44 ± 0.05	0.31 ± 0.05
SMOTE+PSO-OFS ($\beta\%=100\%$)	0.44 ± 0.09	0.15 ± 0.06	0.52 ± 0.09	0.61 ± 0.07	0.48 ± 0.09
SMOTE+PSO-OFS ($\beta\%=200\%$)	0.63 ± 0.06	0.23 ± 0.06	0.50 ± 0.07	$\mathbf{0.69}\pm0.08$	$\mathbf{0.55}\pm0.09$
SMOTE+PSO-OFS ($\beta\%=300\%$)	0.80 ± 0.09	0.58 ± 0.07	0.34 ± 0.05	0.57 ± 0.09	0.47 ± 0.05
SMOTE+PSO-OFS ($\beta\%=400\%$)	$\mathbf{0.84}\pm0.08$	0.69 ± 0.08	0.31 ± 0.04	0.51 ± 0.08	0.45 ± 0.05

normalised to within the range [0, 1] using the operation (29). The SMOTE adopted the 5-nearest neighbour scheme to generate synthetic training data. The over-sampling rate $\beta\%$ was set to 0%, 100%, 300%, 500%, 800%, 1000%, 1500% and 2000%, respectively. The swarm size and the number of movements were set to $S=10$ and $L=20$ for the PSO. The 8-fold cross validation results obtained by the various classifiers are shown in Table 5. For the SMOTE+PSO-OFS, the best overall test performance was achieved at

Table 5
Eight-fold cross validation classification performance and standard deviations for ADI data set.

Method	TP%	FP%	Pr	G-mean	F-meas
SMOTE+1-NN ($\beta\%=0\%$)	0.32 ± 0.07	0.08 ± 0.01	0.13 ± 0.03	0.54 ± 0.06	0.19 ± 0.04
SMOTE+1-NN ($\beta\%=100\%$)	0.44 ± 0.10	0.12 ± 0.02	0.11 ± 0.03	0.62 ± 0.07	0.18 ± 0.04
SMOTE+1-NN ($\beta\%=300\%$)	0.51 ± 0.13	0.19 ± 0.02	0.09 ± 0.02	0.64 ± 0.09	0.15 ± 0.04
SMOTE+1-NN ($\beta\%=500\%$)	0.58 ± 0.14	0.22 ± 0.02	0.09 ± 0.02	0.67 ± 0.08	0.15 ± 0.04
SMOTE+1-NN ($\beta\%=800\%$)	0.62 ± 0.12	0.26 ± 0.02	0.08 ± 0.02	0.67 ± 0.07	0.14 ± 0.03
SMOTE+1-NN ($\beta\%=1000\%$)	0.66 ± 0.13	0.27 ± 0.02	0.08 ± 0.02	0.69 ± 0.07	0.14 ± 0.03
SMOTE+1-NN ($\beta\%=1500\%$)	0.69 ± 0.14	0.31 ± 0.02	0.07 ± 0.02	0.69 ± 0.07	0.13 ± 0.03
SMOTE+1-NN ($\beta\%=2000\%$)	0.73 ± 0.14	0.34 ± 0.02	0.07 ± 0.01	0.69 ± 0.07	0.13 ± 0.02
SMOTE+3-NN ($\beta\%=0\%$)	0.23 ± 0.08	0.04 ± 0.01	0.18 ± 0.06	0.46 ± 0.08	0.20 ± 0.07
SMOTE+3-NN ($\beta\%=100\%$)	0.39 ± 0.14	0.11 ± 0.01	0.12 ± 0.03	0.59 ± 0.10	0.18 ± 0.06
SMOTE+3-NN ($\beta\%=300\%$)	0.57 ± 0.02	0.17 ± 0.01	0.11 ± 0.03	0.68 ± 0.10	0.18 ± 0.05
SMOTE+3-NN ($\beta\%=500\%$)	0.65 ± 0.15	0.22 ± 0.02	0.10 ± 0.03	0.71 ± 0.08	0.17 ± 0.04
SMOTE+3-NN ($\beta\%=800\%$)	0.69 ± 0.16	0.28 ± 0.02	0.08 ± 0.02	0.70 ± 0.09	0.15 ± 0.04
SMOTE+3-NN ($\beta\%=1000\%$)	0.73 ± 0.17	0.30 ± 0.02	0.08 ± 0.02	0.71 ± 0.09	0.15 ± 0.03
SMOTE+3-NN ($\beta\%=1500\%$)	0.76 ± 0.16	0.34 ± 0.02	0.07 ± 0.01	0.70 ± 0.08	0.14 ± 0.03
SMOTE+3-NN ($\beta\%=2000\%$)	0.77 ± 0.13	0.38 ± 0.03	0.07 ± 0.01	0.69 ± 0.06	0.13 ± 0.02
LOO-AUC+OFS ($\rho=1$)	0.21 ± 0.03	$\mathbf{0.01}\pm0.01$	0.67 ± 0.08	0.46 ± 0.03	0.32 ± 0.04
LOO-AUC+OFS ($\rho=5$)	0.55 ± 0.09	0.14 ± 0.02	0.33 ± 0.02	0.68 ± 0.05	0.41 ± 0.04
LOO-AUC+OFS ($\rho=10$)	0.71 ± 0.05	0.22 ± 0.03	0.30 ± 0.01	0.74 ± 0.02	0.42 ± 0.01
LOO-AUC+OFS ($\rho=15$)	0.77 ± 0.02	0.25 ± 0.02	0.28 ± 0.01	0.76 ± 0.01	0.41 ± 0.02
LOO-AUC+OFS ($\rho=20$)	0.88 ± 0.03	0.36 ± 0.04	0.24 ± 0.02	0.75 ± 0.02	0.37 ± 0.02
κ-means+WLSE ($\rho=1$)	0.19 ± 0.03	0.02 ± 0.00	0.61 ± 0.05	0.43 ± 0.03	0.29 ± 0.03
κ-means+WLSE ($\rho=5$)	0.62 ± 0.03	0.17 ± 0.02	0.32 ± 0.02	0.72 ± 0.01	0.42 ± 0.02
κ-means+WLSE ($\rho=10$)	0.80 ± 0.03	0.27 ± 0.02	0.28 ± 0.02	$\mathbf{0.77}\pm0.02$	0.42 ± 0.02
κ-means+WLSE ($\rho=15$)	0.87 ± 0.02	0.34 ± 0.03	0.25 ± 0.02	0.75 ± 0.02	0.38 ± 0.02
κ-means+WLSE ($\rho=20$)	$\mathbf{0.91}\pm0.02$	0.44 ± 0.03	0.21 ± 0.01	0.71 ± 0.02	0.34 ± 0.02
SMOTE+PSO-OFS ($\beta\%=0\%$)	0.20 ± 0.04	$\mathbf{0.01}\pm0.01$	$\mathbf{0.70}\pm0.09$	0.44 ± 0.04	0.30 ± 0.03
SMOTE+PSO-OFS ($\beta\%=100\%$)	0.30 ± 0.07	0.04 ± 0.02	0.53 ± 0.09	0.55 ± 0.05	0.39 ± 0.03
SMOTE+PSO-OFS ($\beta\%=300\%$)	0.51 ± 0.07	0.11 ± 0.03	0.38 ± 0.04	0.67 ± 0.02	$\mathbf{0.43}\pm0.02$
SMOTE+PSO-OFS ($\beta\%=500\%$)	0.72 ± 0.09	0.23 ± 0.06	0.29 ± 0.03	0.74 ± 0.02	0.41 ± 0.03
SMOTE+PSO-OFS ($\beta\%=800\%$)	0.77 ± 0.07	0.28 ± 0.08	0.27 ± 0.03	0.74 ± 0.02	0.40 ± 0.03
SMOTE+PSO-OFS ($\beta\%=1000\%$)	0.82 ± 0.04	0.29 ± 0.04	0.27 ± 0.02	0.76 ± 0.01	0.41 ± 0.01
SMOTE+PSO-OFS ($\beta\%=1500\%$)	0.89 ± 0.04	0.35 ± 0.04	0.25 ± 0.02	0.76 ± 0.02	0.39 ± 0.02
SMOTE+PSO-OFS ($\beta\%=2000\%$)	0.88 ± 0.02	0.35 ± 0.03	0.24 ± 0.02	0.75 ± 0.02	0.38 ± 0.02

the over-sampling rate around $\beta\% = 1000\%$ to 1500%, and the SMOTE+PSO-OFS showed a competitive performance to the other methods.

6. Conclusions

The RBF classifier performs well on balanced or slightly imbalanced data sets, and our previous work has provided an efficient and tunable RBF classifier optimised by the PSO based on the OFS procedure. For highly imbalanced data sets, however, the performance of the tunable RBF classifier may no longer be satisfactory. In order to combat challenging imbalanced classification problems, many approaches have been proposed, which aim to reduce the influence from the underlying imbalanced distribution. In particular, the SMOTE is effective to increase the significance of the positive class in the decision region. In this contribution, we have proposed a powerful and efficient algorithm for solving two-class imbalanced problems, referred to as the SMOTE+PSO-RBF, by combining the SMOTE and the PSO optimised RBF classifier. The experimental results presented in this study have demonstrated that the proposed SMOTE+PSO-RBF offers a very competitive solution to other existing state-of-the-arts methods for combating imbalanced classification problems.

Acknowledgements

This work was supported by UK EPSRC. The authors would like to thank Dr. P.A.S. Reed and Dr. K.K. Lee for their help with ADI data set.

References

[1] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.
[2] G.M. Weiss, Mining with rarity: a unifying framework, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 7–19.
[3] N. Japkowicz, Concept-learning in the presence of between-class and within-class imbalances, in: E. Stroulia, S. Matwin (Eds.), Advances in Artificial Intelligence, vol. 2056, Springer-Verlag, Berlin, 2001, pp. 67–77.
[4] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, A. Geissbuhler, Learning from imbalanced data in surveillance of nosocomial infection, Artif. Intell. Med. 37 (2006) 7–18.
[5] R.B. Rao, S. Krishnan, R.S. Niculescu, Data mining for improved cardiac care, ACM SIGKDD Explor. Newsl. 8 (1) (2006) 3–10.
[6] C. Yu, L. Chou, D. Chang, Predicting protein–protein interactions in unbalanced data using the primary structure of proteins, BMC Bioinformatics 11 (1) (2010) 167–177.
[7] F. Provost, T. Fawcett, R. Kohavi, The case against accuracy estimation for comparing induction algorithms, in: Proceedings of the 15th International Conference on Machine Learning, Madison, USA, July 24–27, 1998, pp. 445–453.
[8] T. Fawcett, F. Provost, Adaptive fraud detection, Data Min. Knowl. Discovery 1 (1997) 291–316.
[9] D.A. Cieslak, N.V. Chawla, A. Striegel, Combating imbalance in network intrusion datasets, in: Proceedings of the 2006 IEEE International Conference on Granular Computing, Atlanta, USA, May 10–12, 2006, pp. 732–737.
[10] G.M. Weiss, H. Hirsh, Learning to predict rare events in event sequences, in: Proceedings of the Fourth IEEE International Conference on Knowledge Discovery and Data Mining, New York, USA, August 27–31, 1998, pp. 359–363.
[11] F. Provost, Machine Learning from Imbalanced Data Sets 101, AAAI Workshop on Learning from Imbalanced Data Sets, 2000.
[12] H. He, A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[13] R. Barandela, J.S. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (2003) 849–851.
[14] V. García, J.S. Sánchez, R.A. Mollineda, R. Alejo, J.M. Sotoca, The Class Imbalance Problem in Pattern Classification and Learning, in: II Congreso Español de Informática, 2007, pp. 283–291.
[15] N.V. Chawla, D.A. Cieslak, L.O. Hall, A. Joshi, Automatically countering imbalance and its empirical relationship to cost, Data Min. Knowl. Discovery 17 (2) (2008) 225–252.
[16] G.M. Weiss, K. McCarthy, B. Zabar, Cost-sensitive learning vs. sampling: which is best for handling unbalanced classes with unequal error costs?, in: Proceedings of the 2007 IEEE International Conference on Data Mining, 2007, pp. 35–41.
[17] G.M. Weiss, F. Provost, Learning when training data are costly: the effect of class distribution on tree induction, Artif. Intell. Res. 19 (2003) 315–354.
[18] A. Estabrooks, T. Jo, N. Japkowicz, A multiple resampling method for learning from imbalanced data sets, J. Chem. Inf. Modeling 20 (1) (2004) 18–36.
[19] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explor. Newsl. 6 (1) (2004) 20–29.
[20] C. Drummond, R.C. Holte, C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling, in: 2003 International Conference on Machine Learning—Workshop Learning from Imbalanced Datasets II, Washington DC, USA, August 21, 2003, pp. 1–8.
[21] N.V. Chawla, K.W. Bowyer, L.O. Hall, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[22] C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence, Seattle, USA, August 4–10, 2001, pp. 973–978.
[23] K.M. Ting, An instance-weighting method to induce cost-sensitive trees, IEEE Trans. Knowl. Data Eng. 14 (3) (2002) 659–665.
[24] M.A. Maloof, Learning when data sets are imbalanced and when costs are unequal and unknown, in: 2003 International Conference on Machine Learning—Workshop Learning from Imbalanced Datasets II, Washington DC, USA, August 21, 2003, pp. 1–8.
[25] K. McCarthy, B. Zabar, G. Weiss, Does cost-sensitive learning beat sampling for classifying rare classes? in: Proceedings of the First International Workshop on Utility-Based Data Mining, Chicago, USA, August 21, 2005, pp. 69–77.
[26] X. Hong, S. Chen, C.J. Harris, A kernel-based two-class classifier for imbalanced data sets, IEEE Trans. Neural Networks 18 (1) (2007) 28–41.
[27] J. Moody, C.J. Darken, Fast learning in networks of locally-tuned processing units, Neural Comput. 1 (1989) 281–294.
[28] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall, Upper Saddle River, NJ, 1999.
[29] X. Hong, S. Chen, C.J. Harris, A fast linear-in-the-parameters classifier construction algorithm using orthogonal forward selection to minimize leave-one-out misclassification rate, Int. J. Syst. Sci. 39 (2) (2008) 119–125.
[30] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 (2) (1991) 302–309.
[31] X. Hong, P.M. Sharkey, K. Warwick, Automatic nonlinear predictive model construction using forward regression and the press statistic, IEE Proc. Control Theory Appl. 150 (3) (2003) 245–254.
[32] X. Hong, P.M. Sharkey, K. Warwick, A robust nonlinear identification algorithm using press statistic and forward regression, IEEE Trans. Neural Networks 14 (2) (2003) 454–458.
[33] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modelling using orthogonal forward regression with PRESS statistic and regularization, IEEE Trans. Syst. Man Cybern. 34 (2) (2004) 898–911.
[34] S. Chen, X.X. Wang, X. Hong, C.J. Harris, Kernel classifier construction using orthogonal forward selection and boosting with Fisher ratio class separability measure, IEEE Trans. Neural Networks 17 (6) (2006) 1652–1656.
[35] S. Chen, X. Hong, C.J. Harris, Construction of RBF classifiers with tunable units using orthogonal forward selection based on leave-one-out misclassification rate, in: Proceedings of the 2006 International Joint Conference on Neural Networks, Vancouver, Canada, July 16–21, 2006, pp. 6390–6394.
[36] S. Chen, X. Hong, B.L. Luk, C.J. Harris, Construction of tunable radial basis function networks using orthogonal forward selection, IEEE Trans. Syst. Man, Cybern. Part B 39 (2) (2009) 457–466.
[37] S. Chen, X.X. Wang, C.J. Harris, Experiments with repeating weighted boosting search for optimization in signal processing applications, IEEE Trans. Syst. Man Cybern. Part B 35 (4) (2005) 682–693.
[38] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, 2001.
[39] S. Chen, X. Hong, C.J. Harris, Radial basis function classifier construction using particle swarm optimisation aided orthogonal forward regression, in: Proceedings of the 2010 International Joint Conference on Neural Networks, Barcelona, Spain, July 18–23, 2010, pp. 3418–3423.
[40] S. Chen, X. Hong, C.J. Harris, Particle swarm optimization aided orthogonal forward regression for unified data modelling, IEEE Trans. Evol. Comput. 14 (4) (2010) 477–499.
[41] A. Ratnaweera, S.K. Halgamuge, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.
[42] W.-F. Leong, G.G. Yen, PSO-based multiobjective optimization with dynamic population size and adaptive local archives, IEEE Trans. Syst. Man Cybern. Part B 38 (5) (2008) 1270–1293.
[43] 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.
[44] M. Ramezani, M.-R. Haghifam, C. Singh, H. Seifi, M.P. Moghaddam, Determination of capacity benefit margin in multiarea power systems using particle swarm optimization, IEEE Trans. Power Syst. 24 (2) (2009) 631–641.
[45] S. Chen, W. Yao, H.R. Palally, L. Hanzo, Particle swarm optimisation aided MIMO transceiver designs, in: Y. Tenne, C.-K. Goh (Eds.), Computational Intelligence in Expensive Optimization Problems, Springer-Verlag, Berlin, 2010, pp. 487–511.

[46] H.L. Wei, S.A. Billings, Y. Zhao, L.Z. Gu, Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio-temporal system identification, IEEE Trans. Neural Networks 20 (1) (2009) 181–185.
[47] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: Proceedings of the 15th European Conference on Machine Learning, Pisa, Italy, September 20–24, 2004, pp. 39–50.
[48] P. Kang, S. Cho, Eus svms: ensemble of under-sampled svms for data imbalance problems, in: Proceedings of the 13th International Conference on Neural Information Processing, Hong Kong, China, October 3–6, 2006, pp. 837–846.
[49] B.X. Wang, N. Japkowicz, Boosting support vector machines for imbalanced data sets, in: Proceedings of the 17th International Conference on Foundations of Intelligent Systems, Toronto, Canada, May 20–23, 2008, pp. 38–47.
[50] Y. Tang, Y.-Q. Zhang, Z. Huang, X. Hu, Granular svm-rfe gene selection algorithm for reliable prostate cancer classification on microarray expression data, in: Proceedings of the 5th IEEE Symposium on Bioinformatics and Bioengineering, Minneapolis, USA, October 19–21, 2005, pp. 290–293.
[51] Y. Tang, Y.-Q. Zhang, Granular svm with repetitive undersampling for highly imbalanced protein homology prediction, in: Proceedings of the 2006 IEEE International Conference on Granular Computing, Atlanta, USA, May 10–12, 2006, pp. 457–460.
[52] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[53] R.H. Myers, Classical and Modern Regression with Applications, second ed., PWS-KENT, Boston, 1990.
[54] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, Department of Computer Science, University of California, Irvine, CA, 1998 ⟨http://archive.ics.uci.edu/ml/datasets.html⟩.
[55] K.K. Lee, C.J. Harris, S.R. Gunn, P.A.S. Reed, Classification of imbalanced data with transparent kernel, in: Proceedings of the 2001 International Joint Conference on Neural Networks, Washington DC, USA, July 15–19, 2001, pp. 2410–2415.
[56] A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (1997) 1145–1159.
[57] C. van Rijsbergen, Information Retrieval, Butterworths, London, UK, 1979.

Ming Gao received the MEng degree from Northwestern Polytechnical University (NPU), Shaan'xi, P.R. China, in 2006, and the MEng degree from the Beihang University, Beijing, P.R. China, in 2009. He is now working towards the PhD degree in the Systems Engineering School, the University of Reading (UoR), Reading, UK. His research interests are machine learning, pattern recognition, and their applications in imbalanced problems.

Xia Hong received her university education at National University of Defense Technology, P.R. China (BSc, 1984; MSc, 1987), and University of Sheffield, UK (PhD, 1998), all in automatic control. She worked as a Research Assistant in Beijing Institute of Systems Engineering, Beijing, China from 1987 to 1993. She worked as a Research Fellow in the Department of Electronics and Computer Science at University of Southampton from 1997 to 2001. She is currently a 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.

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 conducted 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 conducted 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 coauthored 12 books and over 400 research papers, and he was the associate editor of numerous international journals including Automatica, Engineering Applications of AI, International Journal of General Systems Engineering, International Journal of System Science and the International Journal 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.