Multiobjective Support Vector Machines: Handling Class Imbalance With Pareto Optimality

Shounak Datta and Swagatam Das

***Abstract*— Support vector machines (SVMs) seek to optimize three distinct objectives: maximization of margin, minimization of regularization from the positive class, and minimization of regularization from the negative class. The right choice of weightage for each of these objectives is critical to the quality of the classifier learned, especially in case of the class imbalanced data sets. Therefore, costly parameter tuning has to be undertaken to find a set of suitable relative weights. In this brief, we propose to train SVMs, on two-class as well as multiclass data sets, in a multiobjective optimization framework called radial boundary intersection to overcome this shortcoming. The experimental results suggest that the radial boundary intersection-based scheme is indeed effective in finding the best tradeoff among the objectives compared with parameter-tuning schemes.**

***Index Terms*— Class imbalance, classification, multiobjective optimization, radial boundary intersection (RBI), support vector machines (SVMs).**

I. Introduction

Support vector machines (SVMs) are a class of popular linear binary classifiers, proposed by Cortes and Vapnik [1] in 1995. SVMs can also be applied to nonlinear classification problems using the kernel trick (projecting the data to a higher dimensional space where the classes are linearly separable). SVMs have high generalization capability owing to the ability to find the separating hyperplane which maximizes the margin, the sum of the minimum distances from the two classes to the hyperplane [1]. Regularizing some of the points in the region of overlap (or the region of transition) between the two classes can result in a higher margin. On the other hand, too much of such regularization will result in a miscalibrated hyperplane. Therefore, SVMs require a parameter-tuning process to strike a good balance between the regularization of points and the maximization of the margin.

Class imbalance refers to the situation in which all the classes present in a data set are not equally represented [2], [3]. In other words, one or more (but not all) of the classes have a low fraction of representation in the data. The degree of imbalance can be measured using the imbalance ratio (IR), which is the ratio of the number of representatives from the largest class to that of the smallest class. This is a common phenomenon, which plagues many real-world applications, such as fraud detection and medical diagnosis. The traditional formulations of many aspects of machine learning, such as classification [3], feature selection [4], and semisupervised learning [5], are sensitive to the presence of class imbalance. Such sensitivity of classifiers to class imbalance generally results in the misclassification of the minority class(es) [often being the more important class(es)]. For example, in a credit-card fraud detection application, information about fraudsters is hard to come by, while data for genuine customers are abundant. Hence, a traditional SVM may classify some fraudsters as genuine, potentially resulting in huge losses to the financial firm.

The sensitivity of traditional SVMs to class imbalance is a result of equal penalty being accorded to regularization of points from both the minority class and the majority class. Consequently, due to the scarcity of representatives from the minority class, a large fraction of its representatives are regularized. This results in a bias in favor of classifying new points into the majority class. The common remedies used to make SVMs immune to class imbalance are listed in the following along with the principal disadvantages of each approach.

1) *Oversampling the Minority Class [6], [7]:* The proper extent of oversampling is unknown; oversampling results in higher training complexity due to the increase in the number of minority points.
2) *Undersampling the Majority Class [8]–[10]:* The proper extent of undersampling is unknown; undersampling may result in the loss of critical majority class points.
3) *Cost-Sensitive Learning by Assigning a Lower Penalty for the Regularization of Points From the Majority Class [11], [12]:* The proper set of relative costs is unknown.
4) *Shifting the Separating Hyperplane to Compensate for the Imbalance [13], [14]:* The proper extent of compensation is unknown.
5) *Perturbing the Radial-Basis Function (RBF) Kernels Using Conformal Transformation to Increase Resolution Around Minority Points [15]–[17]:* The proper extent of perturbation required for each class is unknown; can only be applied to RBF kernel-based SVMs.

Furthermore, hybrid methods that combine two or more of the above-mentioned approaches have also been proposed. On one hand, works, such as [18] and [19], combine oversampling with undersampling. On the other hand, works, such as [20] and [21], combine oversampling and cost-sensitive learning. The reader may refer to the survey by Batuwita and Palade [22] for a more detailed discussion on the SVM variants used for imbalanced classification.

It is discernible from the above-mentioned discussion that most imbalance resilient variants of SVM rely on costly parameter tuning to ascertain the proper extent of compensation required to undo the bias due to the abundance of the majority class. The details of tunable parameters for each of the categories are listed in Table I. Moreover, as mentioned earlier, SVMs also require parameter tuning to balance the regularization of datapoints and the maximization of the margin. This results in a complex parameter-tuning regime.

One way to forgo these drawbacks is to simultaneously optimize each of the objectives independently of each other, instead of optimizing a weighted combination thereof. Such optimization problems are known as multiobjective optimization (MOO) problems. When the objectives to be optimized in an MOO problem are in conflict, there does not exist a single solution that can optimize all the objectives together. In such a situation, an MOO solver finds a set of mutually

Manuscript received December 13, 2017; revised May 28, 2018; accepted August 31, 2018. *(Corresponding author: Swagatam Das.)*

The authors are with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata 700108, India (e-mail: shounak.jaduniv@gmail.com; swagatam.das@isical.ac.in).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2869298

TABLE I
LIST OF TUNABLE PARAMETERS FOR CLASSIFIERS RESILIENT TO CLASS IMBALANCE

Method(s)	Tunable parameters for handling class imbalance
Oversampling methods:	
Chawla *et al.* [6], Cervantes *et al.* [7]	S_+ - extent of oversampling of the +ve (minority) class
Undersampling methods:	
Kubat & Matwin [8], Japkowicz [9]	U_- - extent of undersampling of the -ve (majority) class
Cost-sensitive methods:	
Veropoulos *et al.* [11], Masnadi-Shirazi & Vasconcelos [12]	C_+ & C_- - misclassification costs of the +ve and -ve classes, respectively
Hyperplane shifting methods:	
Imam *et al.* [13]	z - relative importance of the minority support vectors
Datta & Das [14]	$P_{eff,+}$ & $P_{eff,-}$ - effective probabilities of the +ve and -ve classes, respectively
Kernel perturbation methods:	
Wu & Chang [15], Maratea *et al.* [16]	one or more parameters for each class determining the extent of kernel perturbation for that particular class
Hybrid methods:	
Wang [18], Peng *et al.* [19]	S_+, U_-, or both
Akbani *et al.* [20], Wang *et al.* [21]	C_+, C_-, S_+, or some combination thereof
Multi-objective method:	
Aşkan & Sayın [23]	$\sum \xi_i^+$ & $\sum \xi_i^-$ - class-wise sum of slack variables (indirectly tuned based on the parameters $Ngrids$ and δ)

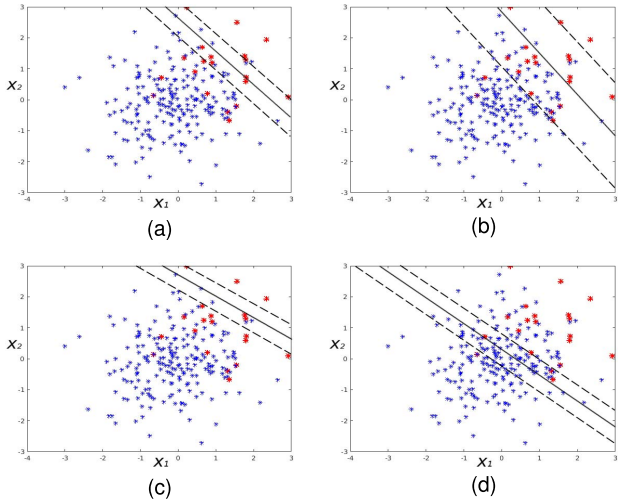

Fig. 1. Illustration showing contradictory relationship between the SVM objectives for a 2-D imbalanced data set. Solid line: decision boundary. Dotted lines: regularized region determining the margin. (a) Small margin and low regularization. (b) Large margin and high regularization. (c) Low regularization from the majority class and high regularization from the minority class. (d) Low regularization from the minority class and high regularization from the majority class. (Best viewed in color.)

nondominated (see Section II) solutions offering the best possible tradeoffs among the objectives.

Three distinct objectives must be optimized for training an SVM on a binary class imbalanced data set, viz., margin maximization, minimization of regularization from the majority class, and minimization of regularization from the minority class. Margin maximization is contradictory to the other objectives, as points from both classes may have to be regularized to maximize the margin [compare Fig. 1(a) and (b)]. Furthermore, improving the performance of the classifier on one of the classes may result in increased regularization from the other class. Hence, the two latter objectives also contradict each other [see Fig. 1(c) and (d)]. Consequently, training such a classifier would lend itself well to the MOO approach. As an additional advantage, the disassociation of the two latter objectives will not allow the abundance of the majority class to unduly affect the learning on the minority class. Moreover, the MOO framework can be used to directly train SVMs on multiclass data sets by incorporating additional objectives for each additional class.

Due to the potential advantages, Bi [24] and Tatsumi *et al.* [25] have, respectively, explored ways to train two-class and multiclass SVMs using MOO. The only MOO-based SVM variant for handling class imbalance is due to Aşkan and Sayın [23], which trains linear SVMs using MOO. However, they solve the MOO problem by transforming it into a set of single-objective problems. This is achieved by converting the latter two objectives into constraints which limit the regularization from each class. Therefore, a grid search must be employed over the set of possible constraints, introducing additional parameters (see Table I) and also impeding direct application to multiclass problems. Therefore, we are motivated to propose a more scalable method for training SVMs in an MOO framework.

In this brief, we propose to train linear programming-based SVMs (LP-SVM) [26] for imbalanced classification by using a recent MOO method called radial boundary intersection (RBI) [27]. The resulting linear subproblems can be efficiently solved using existing LP solvers. This brief adds to the existing body of work in the following ways.

1) We show how LP-SVMs can be trained using an RBI-based decomposition to explore the tradeoffs between the objectives, without having to resort to parameter tuning. The proposed formulations are general and can also be applied to general classification tasks not suffering from class imbalance.
2) Since the RBI framework does-away with the need for parameter tuning and also does not require constraints to limit the regularization from the various classes, we are able to explore the extension of the proposal to multiclass data sets, both directly as well as using the one-versus-all (OVA) strategy.
3) Unlike the methods of [23] and [25], the proposed formulations can make use of the kernel trick, making them more suitable for nonlinear classification problems.

II. MULTIOBJECTIVE OPTIMIZATION AND RADIAL BOUNDARY INTERSECTION

Mathematically, an MOO problem can be expressed as

$$\min_{\mathbf{x}\in\Omega} \mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}, \quad m \geq 2 \tag{1}$$

where $\Omega = \{\mathbf{x} : \mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}\}$ is the feasible region, $\mathbf{F}$ is the vector of objective functions, and $\mathbf{g}$ and $\mathbf{h}$ are the vectors of inequality and equality constraints, respectively, while $\mathbf{lb} \in (\mathbb{R} \cup \{-\infty\})^n$ and $\mathbf{ub} \in (\mathbb{R} \cup \{\infty\})^n$ specify the lower and upper limits of the decision variables (n is the number of decision variables, while m is the number of objectives). Since the objectives are often mutually conflicting, there does not usually exist a single $\mathbf{x}^* \in \Omega$ that minimizes all the objectives together. Instead, one can

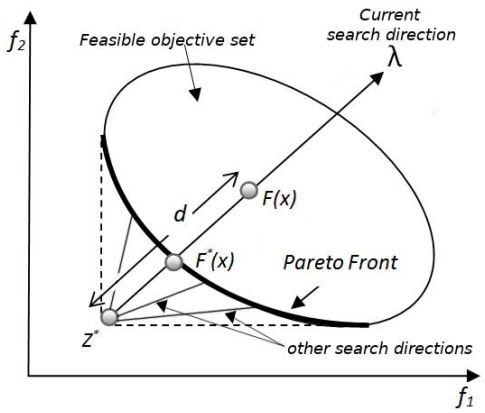

Fig. 2. RBI attempts to find $\mathbf{F}^*(\mathbf{x})$ around the direction λ so as to minimize the distance d from the ideal point $\mathbf{z}^*$.

find optimal tradeoffs among the objectives based on the concepts of Pareto dominance and Pareto optimality. An objective vector $\mathbf{F}(\mathbf{x}_1)$ is said to Pareto dominate another objective vector $\mathbf{F}(\mathbf{x}_2)$, denoted as $\mathbf{F}(\mathbf{x}_1) \succ \mathbf{F}(\mathbf{x}_2)$, if and only if $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)\ \forall i = \{1, 2, \cdots, m\}$ and $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ for at least one $j \in \{1, 2, \cdots, m\}$. A point $\mathbf{x}^* \in \Omega$ is said to be locally Pareto optimal if there exists an open neighborhood $C(\mathbf{x}^*)$ of $\mathbf{x}^*$, such that $\nexists \mathbf{x} \in C(\mathbf{x}^*) \bigcap \Omega$ which dominates $\mathbf{F}(\mathbf{x}^*)$. Furthermore, it is said to be globally Pareto optimal if $\mathbf{F}(\mathbf{x}^*)$ is globally nondominated. In the remainder of this brief, the phrase "Pareto optimality" is used in the sense of local Pareto optimality, unless otherwise stated. Since there can be an infinite number of tradeoffs among the multiple objectives, there exist an infinite number of Pareto optimal solutions. However, an MOO solver finds a finite subset of such solutions called the *Pareto set* (PS). The Pareto front (PF) is the continuous curve formed by the images of all the Pareto optimal solutions in the objective space. Therefore, the images of the PS in the objective space correspond to a finite sampling from the PF.

MOO problems are commonly solved using stochastic/evolutionary optimization techniques, such as SPEA-2 [28], NSGA-II [29], MOEA/D [30], and NICA [31].[1] However, such methods are often computationally expensive. Recently, Datta *et al.* [27] proposed the RBI scheme that can generate a set of diverse solutions having a good coverage of the PF (even along the periphery). The principal idea behind RBI is to select a set of diverse directions λ emanating radially outward from the ideal point $\mathbf{z}^*$. The MOO problem can then be *decomposed* into different subproblems, each of which strives to find a feasible solution as close as possible to the ideal point, in and around a particular direction. Each of these subproblems represents a particular tradeoff among the objectives, and their solutions comprise the PS. The concept of RBI is shown in Fig. 2. Mathematically, this results in single-objective subproblems of the form

$$\min_{\mathbf{x}} g(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*) = d \tag{2a}$$

$$\text{s.t. } \mathbf{z}^* - \mathbf{F}(\mathbf{x}) + d\boldsymbol{\lambda} \leq 0 \tag{2b}$$

$$\mathbf{h}(\mathbf{x}) = 0 \tag{2c}$$

and

$$\mathbf{g}(\mathbf{x}) \leq 0 \tag{2d}$$

[1]Strength Pareto Evolutionary Algorithm-2, Nondominated Sorting Genetic Algorithm-II, multiobjective evolutionary algorithm based on decomposition, and novel immune clonal algorithm.

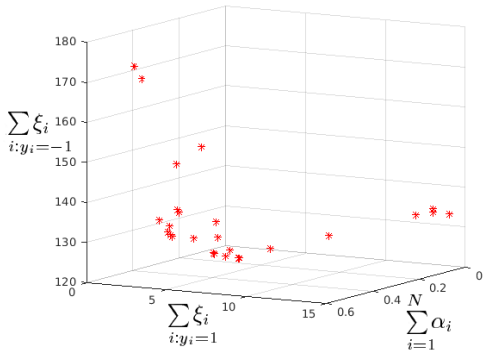

Fig. 3. Samples from the PF obtained by RBI-based LP-SVM on the 2-D imbalanced data set of Fig. 1.

where $d = \|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^*)^T \boldsymbol{\lambda}\| / \|\boldsymbol{\lambda}\|$, and constraint (2b) extends the search to a neighborhood of the selected direction λ to ensure local Pareto optimality [32]. The RBI scheme also has the advantage that efficient search directions can be chosen based on domain knowledge (see Section IV). Due to the high computational expense of the popular stochastic/evolutionary MOO solvers, we propose to decompose LP-SVMs into RBI-based LP subproblems that can be efficiently solved by using the existing LP solvers. The samples from the PF obtained by training LP-SVM using the RBI framework (see Section III) on the 2-D data set of Fig. 1 is shown in Fig. 3.

III. Multiobjective Maximum Margin Machines

A. Two-Class RBI-LP-SVM

The single-objective optimization problem posed by the LP-SVM [26] is as follows:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\xi}, b} \frac{1}{2} \sum_{i=1}^{N} \alpha_i + C \sum_{i=1}^{N} \xi_i \tag{3a}$$

$$\text{s.t. } y_i \left(\sum_{j=1}^{N} \alpha_j y_j K_{ij} - b \right) \geq 1 - \xi_i \quad \forall i \tag{3b}$$

and

$$\xi_i \geq 0 \quad \forall i \tag{3c}$$

where C is the tunable parameter determining the common cost of regularization for points from both classes, $y_i \in \{-1, 1\}$ is the label corresponding to the point $\mathbf{x}_i$, N is the number of training points, K_{ij} is the kernel measuring the inner product between the points $\mathbf{x}_i$ and $\mathbf{x}_j$, ξ_i is the slack variable that measures the extent of regularization of the point $\mathbf{x}_i$, b is the bias term determining the distance of the separating hyperplane from the origin, and $\boldsymbol{\alpha}$ is the vector of coefficients such that

$$\alpha_i \in \begin{cases} \{0\} & \text{if } y_i \left(\sum_{j=1}^{N} \alpha_j y_j K_{ij} - b \right) > 1 \\ (0, C) & \text{if } y_i \left(\sum_{j=1}^{N} \alpha_j y_j K_{ij} - b \right) = 1 \\ \{C\} & \text{if } y_i \left(\sum_{j=1}^{N} \alpha_j y_j K_{ij} - b \right) < 1. \end{cases} \tag{4}$$

Therefore, we can obtain a multiobjective formulation of LP-SVM by separately considering each of the terms in the objective function

of (3), obtaining the MOO

$$\min_{\boldsymbol{\alpha},\boldsymbol{\xi},b} \mathbf{F}_1(\boldsymbol{\alpha},\boldsymbol{\xi},b) = \begin{bmatrix} \sum_{i=1}^{N} \alpha_i \\ \frac{1}{n_+} \sum_{i:y_i=1} \xi_i \\ \frac{1}{n_-} \sum_{i:y_i=-1} \xi_i \end{bmatrix} \tag{5a}$$

$$\text{s.t. } y_i \left(\sum_{j=1}^{N} \alpha_j y_j K_{ij} - b \right) \geq 1 - \xi_i \quad \forall i \tag{5b}$$

and

$$\xi_i \geq 0 \quad \forall i \tag{5c}$$

where n_+ and n_- are the number of minority and majority class points, respectively. The RBI-based decomposition of the MOO problem of (5) yields RBI-LP-SVM subproblems of the form

$$\min_{\boldsymbol{\alpha},\boldsymbol{\xi},b} g^{LP}(\boldsymbol{\alpha},\boldsymbol{\xi},b|\boldsymbol{\lambda},\mathbf{z}^*) = d_{LP} \tag{6a}$$

$$\text{s.t. } \mathbf{z}^* - \mathbf{F}_1(\boldsymbol{\alpha},\boldsymbol{\xi},b) + d\boldsymbol{\lambda} \leq 0 \tag{6b}$$

$$y_i \left(\sum_{j=1}^{N} \alpha_j y_j K_{ij} - b \right) \geq 1 - \xi_i \quad \forall i \tag{6c}$$

and

$$\xi_i \geq 0 \quad \forall i \tag{6d}$$

where $d_{LP} = \|(\mathbf{F}_1(\boldsymbol{\alpha},\boldsymbol{\xi},b) - \mathbf{z}^*)^T \boldsymbol{\lambda}\| / \|\boldsymbol{\lambda}\|$.

B. Multiclass Extension

Apart from the fact that RBI-LP-SVM can be applied to multiclass problems using OVA or one-versus-one strategies, the MOO framework of RBI can also be used to directly train multiclass SVMs. To directly apply the RBI framework to the multiclass case, we consider a single-objective problem of the form

$$\min_{\boldsymbol{\Gamma},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\beta}} \frac{1}{2} \sum_{c=1}^{\mathcal{C}} \sum_{i=1}^{N} \gamma_{ci} + C \sum_{i:y_i=1} (\xi_i + \eta_i) + C \sum_{i:y_i=2} (\xi_i + \eta_i) + \cdots + C \sum_{i:y_i=\mathcal{C}} (\xi_i + \eta_i) \tag{7a}$$

$$\text{s.t. } y_{ci} \left(\sum_{j=1}^{N} \gamma_{cj} y_{cj} K_{ij} - \beta_c \right) \geq 1 - \xi_i \quad \forall c \ \forall i : y_i = c \tag{7b}$$

$$y_{ci} \left(\sum_{j=1}^{N} \gamma_{cj} y_{cj} K_{ij} - \beta_c \right) \leq -1 + \eta_i \quad \forall c \ \forall i : y_i \neq c \tag{7c}$$

and

$$\xi_i \geq 0, \eta_i \geq 0 \quad \forall i \tag{7d}$$

where points of the cth class are distinguished from those of the other $\mathcal{C} - 1$ classes by the hyperplane determined by the coefficients γ_{ci} (the subscript i denotes the correspondence with the point $\mathbf{x}_i$), β_c is the corresponding bias term, ξ_i (and η_i) is the slack variable for assigning (not assigning) the point $\mathbf{x}_i$ to the cth class, and y_{ci} is the binary label for the same point corresponding to the class c derived from the original class label y_i so that

$$y_{ci} = \begin{cases} 1, & \text{if } y_i = c \\ -1, & \text{otherwise.} \end{cases} \tag{8}$$

The subproblems that result from decomposing the problem (7) using RBI are of the form

$$\min_{\boldsymbol{\Gamma},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\beta}} g^{mLP}(\boldsymbol{\Gamma},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{\lambda},\mathbf{z}^*) = d_{mLP} \tag{9a}$$

$$\text{s.t. } \mathbf{z}^* - \mathbf{F}_2(\boldsymbol{\Gamma},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\beta}) + d\boldsymbol{\lambda} \leq 0 \tag{9b}$$

$$y_{ci} \left(\sum_{j=1}^{N} \gamma_{cj} y_{cj} K_{ij} - \beta_c \right) \geq 1 - \xi_i \quad \forall c \ \forall i : y_i = c \tag{9c}$$

$$y_{ci} \left(\sum_{j=1}^{N} \gamma_{cj} y_{cj} K_{ij} - \beta_c \right) \leq -1 + \eta_i \quad \forall c \ \forall i : y_i \neq c \tag{9d}$$

and

$$\xi_i \geq 0, \quad \eta_i \geq 0 \ \forall i \tag{9e}$$

where $d_{mLP} = \|(\mathbf{F}_2(\boldsymbol{\Gamma},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\beta}) - \mathbf{z}^*)^T \boldsymbol{\lambda}\| / \|\boldsymbol{\lambda}\|$, and

$$\mathbf{F}_2(\boldsymbol{\Gamma},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\beta}) = \begin{bmatrix} \sum_{c=1}^{\mathcal{C}} \sum_{i=1}^{N} \gamma_{ci} \\ \frac{1}{n_1} \sum_{i:y_i=1} (\xi_i + \eta_i) \\ \frac{1}{n_2} \sum_{i:y_i=2} (\xi_i + \eta_i) \\ \vdots \\ \frac{1}{n_{\mathcal{C}}} \sum_{i:y_i=\mathcal{C}} (\xi_i + \eta_i) \end{bmatrix} \tag{10}$$

n_j is the number of points in the jth class. We refer to this formulation as RBI-LP-mSVM, while the direct OVA extension of the formulation in (6) is referred as RBI-LP-OVA-SVM.

IV. Experiments

We demonstrate the effectiveness of the proposed methods on 12 small-scale two-class data sets having IR ranging from 1 to 50, 5 large-scale two-class data sets having IR ranging from 1 to 50, 7 small-scale multiclass imbalanced data sets having between 3 and 8 classes and IR ranging from 1 to 50, and 4 large-scale multiclass data sets having between 3 and 7 classes and IR ranging from 1 to 50. The data sets are obtained from the KEEL repository [33], the University of California at Irvine repository [34], the IDA benchmark repository [35], and the Agnostic Learning versus Prior Knowledge Challenge Database [36]. Details about the data sets can be found in the Supplementary Material. Each feature for every data set is standardized so as to have zero mean and unit standard deviation.

We use both SVM and LP-SVM as baselines for the comparison, along with the respective cost-sensitive versions [11] (OVA variants being used for multiclass data sets). Two-class and OVA-based multiclass versions of the recent uNBSVM[2] [14] are also used as a baseline. In addition, we compare against the MOO-based Grid$^-$ method of [23] for two-class data sets and the MOO-based ϵSMOA2[3] algorithm from [25] for the multiclass data sets. The popular SDC[4] method of [20] (which combines oversampling with cost-sensitive learning) is only used for small-scale two-class data sets because of its high computational cost. RBI-LP-SVM is used for the experiments on the small-scale as well as large-scale two-class data sets, while RBI-LP-mSVM is only used for the experiments on

[2]Near Bayesian SVM with unequal costs.
[3]ϵ constrained soft-margin multiobjective one-vs-all 2.
[4]SMOTE with different costs.

TABLE II
PARAMETER SETTINGS FOR CONTENDING ALGORITHMS

Algorithm	Parameter settings
Baseline methods :	
SVM/OVA-SVM	$C \in \mathbb{C}$; Kernels: Linear and Radial Basis Function (RBF); $\sigma \in \mathbb{S}$
Cost-SVM/OVA-Cost-SVM	$C \in \mathbb{C}$; $C_+ \in \mathbb{K}$; $C_- = 1$; Kernels: Linear and RBF; $\sigma \in \mathbb{S}$
LP-SVM/OVA-LP-SVM	$C \in \mathbb{C}$; Kernels: Linear and RBF; $\sigma \in \mathbb{S}$
Cost-LP-SVM/OVA-Cost-LP-SVM	$C \in \mathbb{C}$; $C_+ \in \mathbb{K}$; $C_- = 1$; Kernels: Linear and RBF; $\sigma \in \mathbb{S}$
SDC	$C \in \mathbb{C}$; $C_+ = n_+/n_-$; $C_- = 1$; $S_+ \in \mathbb{S}_1$ if $IR \leq 50$, else $S_+ \in \mathbb{S}_2$; Kernels: Linear and RBF; $\sigma \in \mathbb{S}$
uNBSVM/OVA-uNBSVM	$C \in \{1, 10, 100\}$; $C_+ = n_+/n_-$; $C_- = 1$; $(P_{eff,+}, P_{eff,-}) \in \mathbb{P}$; Kernels: Linear and RBF; $\sigma \in \mathbb{S}$
Grid-	Kernel: Linear; $Ngrids = 5$; $\delta = n_+/Ngrids$
ϵSMOA2	$C \in \mathbb{C}$ for the initial SVM; Kernel: Linear; Final training with the smallest and largest classes respectively as the +ve and -ve classes; Solved using the `fmincon` function in the 2016b release of MATLAB
Proposed methods :	
RBI-LP-SVM	$\mathbf{z}^* = [0, 0, 0]^T$; No. of subproblems = 7; $\sigma \in \mathbb{S}$
RBI-LP-mSVM	$\mathbf{z}^* = [0, 0, \cdots, 0]^T$; No. of subproblems = 7; $\sigma \in \mathbb{S}$
RBI-LP-OVA-SVM	$\mathbf{z}^* = [0, 0, 0]^T$; No. of subproblems = $7\mathcal{C}$ (7 subproblems for each target class in the OVA framework); $\sigma \in \mathbb{S}$

[1] $\mathbb{C} = \{10, 100, 1000\}$.
[2] σ is the kernel width for the RBF kernel. $\mathbb{S} = \{0.1, 0.5, 1, 5, 10, 50, 100\}$.
[3] C_+ and C_- are the relative costs of the +ve (minority) and the -ve (majority) classes, respectively.
[4] $\mathbb{K} = \{\frac{n_+}{4n_-}, \frac{n_+}{3n_-}, \frac{n_+}{2n_-}, \frac{n_+}{n_-}, \frac{2n_+}{n_-}, \frac{3n_+}{n_-}, \frac{4n_+}{n_-}\}$, where n_+ and n_- are the no. of +ve and -ve points, respectively.
[5] S_+ is the extent of oversampling of the +ve class. $\mathbb{S}_1 = \{100\%, 200\%, \cdots, 500\%\}$. $\mathbb{S}_2 = \{200\%, 400\%, \cdots, 1000\%\}$.
[6] $P_{eff,+}$ and $P_{eff,-}$ are respectively the effective probabilities of the +ve and -ve classes.
[7] $\mathbb{P} = \{(\frac{P_-}{1+P_-}, \frac{1}{1+P_-}), (\frac{2P_-}{1+2P_-}, \frac{1}{1+2P_-}), (\frac{5P_-}{1+5P_-}, \frac{1}{1+5P_-})\}$, where $P_- = \frac{n_-}{(n_+ + n_-)}$ is the fraction of representation from the -ve class.
[8] $\mathcal{C}$ is the number of classes in the dataset.

TABLE III
RESULTS FOR TWO-CLASS DATA SETS

Algorithm	12 small-scale datasets					5 large-scale datasets				
	GM		AUC		Total Training	GM		AUC		Total Training
	Avg. Rank	W-T-L	Avg. Rank	W-T-L	Time (sec.)	Avg. Rank	W-T-L	Avg. Rank	W-T-L	Time (sec.)
SVM	4.54	0-5-7	4.88	0-5-7	3.05E+03	3.30	1-2-2	3.00	1-2-2	6.25E+04
Cost-SVM	4.25	0-4-8	3.88	0-7-5	2.31E+04	3.00	1-2-2	3.30	1-2-2	5.52E+05
LP-SVM	6.29	0-3-9	6.13	0-5-7	2.95E+04	5.70	1-0-4	5.70	1-0-4	4.69E+05
Cost-LP-SVM	4.08	0-6-6	3.88	0-8-4	1.86E+05	3.60	1-3-1	3.60	1-2-2	2.91E+06
SDC	7.17	0-0-12	7.13	0-1-11	6.15E+05	-	-	-	-	-
uNBSVM	4.13	0-6-6	4.54	0-6-6	2.11E+03	4.00	1-2-2	4.00	1-1-3	1.90E+04
Grid-	4.38	0-5-7	4.50	0-5-7	2.48E+04	6.20	0-1-4	6.10	0-1-4	9.50E+04
RBI-LP-SVM	**1.17**	-	**1.08**	-	5.79E+04	**2.20**	-	**2.30**	-	1.48E+06

Best results shown in **boldface**.

the small-scale multiclass data sets. RBI-LP-OVA-SVM is used for the large-scale multiclass data sets. This is because the number of constraints in RBI-LP-mSVM (which scales with both the number of points as well as the number of classes) becomes quite large for large-scale multiclass data sets. The parameter settings used for each of the contending methods are summarized in Table II. The directions λ_i are generated for RBI-LP-SVM, RBI-LP-mSVM, and RBI-LP-OVA-SVM according to the formula

$$\lambda_i = \left[\frac{(1-l_i)}{\sqrt{l_i^2 + (1-l_i)^2}}, \frac{(l_i/\sqrt{\mathcal{C}})}{\sqrt{l_i^2 + (1-l_i)^2}}, \cdots, \frac{(l_i/\sqrt{\mathcal{C}})}{\sqrt{l_i^2 + (1-l_i)^2}} \right]$$

where $l_i = (i-1)/(n_\lambda - 1)$ and n_λ is the number of subproblems. Essentially, we vary the coordinate corresponding to margin maximization from 1 to 0 while maintaining identical values for the coordinates corresponding to the different classes, to ensure a good trade-off among all the classes. The codes for the proposed methods can be found at: https://github.com/Shounak-D/Multi-Objective-SVM.

A. Two-Class Data Sets

We compare the tenfold cross-validation results for the proposed RBI-LP-SVM method with those of various states-of-the-art in Table III. We report the average ranks and the Win (W), Tie (T), and Loss (L) counts for the Wilcoxon's ranksum test [37], [38] (with RBI-LP-SVM as the control at 95% confidence level) corresponding to the best average *geometric mean* (GM) [8] and the best average *area under the curve* (AUC) [39] values (the best achieved by any subproblem being reported for the MOO methods). The total training times required for tuning of all parameters and for solving all the subproblems for the MOO methods are reported as well.[5]

The results in Table III show that RBI-LP-SVM generally has achieved the best performance among the contending methods, in terms of both GM and AUC, on both types of data sets. The low average ranking for RBI-LP-SVM attests to this fact. The W-T-L counts suggest that the performance improvement achieved by RBI-LP-SVM over the other contenders is statistically significant.

Apart from the proposed method, Cost-LP-SVM, Cost-SVM, and uNBSVM (all of which rely on cost tuning) are observed to achieve generally good results. Interestingly, the quadratic programming (QP)-based classifiers, such as SVM, Cost-SVM, and uNBSVM, are observed to require lower training times. However, the LP-based Cost-LP-SVM has generally achieved a better performance than Cost-SVM (the best performing QP-based method).

[5] Detailed results can be found in the Supplementary Material.

TABLE IV
RESULTS FOR MULTICLASS DATA SETS

Algorithm	7 small-scale datasets					4 large-scale datasets				
	GM		Avg-AUC		Total Training	GM		Avg-AUC		Total Training
	Avg. Rank	W-T-L	Avg. Rank	W-T-L	Time (sec.)	Avg. Rank	W-T-L	Avg. Rank	W-T-L	Time (sec.)
OVA-SVM	3.64	0-3-4	3.93	0-3-4	2.92E+03	3.63	0-2-2	3.25	0-3-1	6.32E+05
OVA-Cost-SVM	3.50	0-4-3	3.43	0-3-4	2.07E+04	4.25	0-1-3	3.63	0-2-2	4.23E+06
OVA-LP-SVM	4.00	0-4-3	3.93	0-3-4	1.73E+04	4.63	0-2-2	5.25	0-2-2	6.06E+06
OVA-Cost-LP-SVM	2.86	1-5-1	2.57	1-3-3	1.17E+05	4.00	0-1-3	3.00	1-1-2	3.72E+07
OVA-uNBSVM	5.50	0-0-7	5.57	0-0-7	5.26E+02	2.50	1-1-2	4.00	0-2-2	3.98E+02
ϵSMOA2	6.93	0-0-7	6.86	0-0-7	6.78E+03	6.75	0-0-4	6.63	0-0-4	1.62E+08
RBI-LP-mSVM	**1.57**	-	**1.71**	-	3.60E+04	-	-	-	-	-
RBI-LP-OVA-SVM	-	-	-	-	-	**2.25**	-	**2.25**	-	4.86E+06

Best results shown in **boldface**.

The effectiveness of the RBI-based scheme over exhaustive cost tuning is evident from the much lower total training time required by RBI-LP-SVM compared with Cost-LP-SVM. Consequently, the total training time required by the proposed method is greater than but comparable with that of Cost-SVM on small-scale as well as large-scale data sets. Moreover, unlike the MOO scheme of Grid$^-$, the proposed RBI-based scheme is able to maintain a good performance on the large-scale data sets.

B. Multiclass Data Sets

We report the tenfold cross-validation results for multiclass data sets in Table IV. We report the average ranks and the W-T-L counts for the Wilcoxon's ranksum test (with RBI-LP-mSVM and RBI-LP-OVA-SVM as the control, respectively, for the small-scale and large-scale data sets) corresponding to the best average GM and best average Avg-AUC [40] values (the best achieved by any subproblem are reported for the MOO methods) along with total training times.[6]

The results in Table IV mirror those of Table III in that both the proposed RBI-LP-mSVM method (for the small-scale data sets) and RBI-LP-OVA-SVM method (for the large-scale data sets) have achieved the best performance in terms of average rank on GM as well as Avg-AUC. The performance improvement seems to be statistically significant based on the W-T-L counts.

Like in case of the two-class data sets, the QP-based classifiers have required lower training times compared with their LP-based counterparts, with OVA-Cost-SVM being the best in terms of classification performance. Interestingly, despite having $(\mathcal{C}+2)N+\mathcal{C}+1$ constraints, the proposed RBI-LP-mSVM method has scaled well to the small-scale data sets. However, for the large-scale data sets, the RBI-LP-OVA-SVM method had to be employed. RBI-LP-OVA-SVM has achieved the best performance on the large-scale data sets while having total training time comparable with that of OVA-Cost-SVM. This bears testimony to the effectiveness of RBI-LP-SVM in conjunction with the OVA strategy for large-scale multiclass data sets. Therefore, we recommend that RBI-LP-OVA-SVM can be used instead of the direct RBI-LP-mSVM method, for data sets posing more than 3500 constraints for the latter method.

V. CONCLUSION

We introduce a new MOO-based framework that trains LP-SVMs without resorting to costly parameter tuning, by simultaneously minimizing the regularization from each of the classes alongside maximizing the margin. This method, known as RBI-LP-SVM, can be especially useful for class imbalanced learning due to the dissociation between the penalties for regularization from the two classes (association between these two objectives being the principal reason behind the sensitivity of SVMs to class imbalance). We also propose two different extensions of the proposed method to multiclass problems. The first extension, known as RBI-LP-mSVM, enables direct training on multiclass data sets. The second extension, known as RBI-LP-OVA-SVM, uses the OVA strategy to extend RBI-LP-SVM to the multiclass setting. The experimental results show that the proposed methods, which are characterized by a fixed number of convex subproblems, generally perform competitively against the states of the art for two-class as well as multiclass problems. This indicates that the proposed RBI-based framework is able to find better compromises among the objectives compared with parameter-tuning schemes as well as the existing MOO-based schemes.

However, a drawback of the proposed RBI-LP-mSVM variant is that the number of constraints grows with the number of classes as well as the number of data points. This impedes its application to large-scale multiclass data sets. Furthermore, the proposed methods are observed to require greater training times compared with some of the existing methods. Hence, an interesting future avenue of research may be to devise tailor-made solvers that can efficiently solve the subproblems while being able to handle a large number of constraints.

[6]Detailed results can be found in the Supplementary Material.

ACKNOWLEDGMENT

The authors would like to thank Parikshit Shekhawat, prefinal year student pursuing B.Tech. degree in electronics and electrical engineering with IIT Guwahati, Guwahati, India, for helping with the computer implementations of some of the techniques used in our experiments.

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[2] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[3] S. Das, S. Datta, and B. B. Chaudhuri, "Handling data irregularities in classification: Foundations, trends, and future challenges," *Pattern Recognit.*, vol. 81, pp. 674–693, Sep. 2018.
[4] R. Shang, W. Wang, R. Stolkin, and L. Jiao, "Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 793–806, Feb. 2018.
[5] L. C. Jiao, F. Shang, F. Wang, and Y. Liu, "Fast semi-supervised clustering with enhanced spectral embedding," *Pattern Recognit.*, vol. 45, no. 12, pp. 4358–4369, 2012.
[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
[7] J. Cervantes, X. Li, and W. Yu, "Imbalanced data classification via support vector machines and genetic algorithms," *Connection Sci.*, vol. 26, no. 4, pp. 335–348, 2014.
[8] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 14th Int. Conf. Mach. Learn.* Nashville, TN, USA: Morgan Kaufmann, 1997, pp. 179–186.
[9] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Int. Conf. Artif. Intell. (ICAI)*, 2000, pp. 111–117.

[10] J. M. Choi, "A selective sampling method for imbalanced data learning on support vector machines," Ph.D. dissertation, Iowa State Univ., Ames, IA, USA, 2010.
[11] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proc. Int. Joint Conf. AI (IJCAI)*, 1999, pp. 55–60.
[12] H. Masnadi-Shirazi and N. Vasconcelos, "Risk minimization, probability elicitation, and cost-sensitive SVMs," in *Proc. 27th Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2010, pp. 759–766.
[13] T. Imam, K. M. Ting, and J. Kamruzzaman, "z-SVM: An SVM for improved classification of imbalanced data," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 4304. Berlin, Germany: Springer, 2006, pp. 264–273.
[14] S. Datta and S. Das, "Near-Bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs," *Neural Netw.*, vol. 70, pp. 39–52, Oct. 2015.
[15] G. Wu and E. Y. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 786–795, Jun. 2005.
[16] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted F-measure and kernel scaling for imbalanced data learning," *Inf. Sci.*, vol. 257, pp. 331–341, Feb. 2014.
[17] Y. Zhang, P. Fu, W. Liu, and G. Chen, "Imbalanced data classification based on scaling kernel-based support vector machine," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 927–935, 2014.
[18] Q. Wang, "A hybrid sampling svm approach to imbalanced data classification," *Abstr. Appl. Anal.*, vol. 2014, Jun. 2014, Art. no. 972786.
[19] L. Peng, Y. Xiao-Yang, B. Ting-Ting, and H. Jiu-Ling, "Imbalanced data SVM classification method based on cluster boundary sampling and DT-KNN pruning," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 7, no. 2, pp. 61–68, 2014.
[20] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Machine Learning: ECML* (Lecture Notes in Computer Science), vol. 3201. Berlin, Germany: Springer, 2004, pp. 39–50.
[21] S. Wang, Z. Li, W. Chao, and Q. Cao, "Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2012, pp. 1–8.
[22] R. Batuwita and V. Palade, "Class imbalance learning methods for support vector machines," in *Imbalanced Learning Imbalanced Learning: Foundations, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2013, pp. 83–99.
[23] A. Aşkan and S. Sayın, "SVM classification for imbalanced data sets using a multiobjective optimization framework," *Ann. Oper. Res.*, vol. 216, no. 1, pp. 191–203, 2014.
[24] J. Bi, "Multi-objective programming in SVMs," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 35–42.
[25] K. Tatsumi, M. Tai, and T. Tanino, "Multiobjective multiclass support vector machine based on the one-against-all method," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2010, pp. 1–7.
[26] V. N. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.
[27] S. Datta, A. Ghosh, K. Sanyal, and S. Das, "A radial boundary intersection aided interior point method for multi-objective optimization," *Inf. Sci.*, vol. 377, pp. 1–16, Jan. 2017.
[28] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Swiss Federal Inst. Technol. (ETH), Zürich, Switzerland, TIK-Rep. 103, 2001.
[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[30] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[31] R. Shang, L. Jiao, F. Liu, and W. Ma, "A novel immune clonal algorithm for MO problems," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 35–50, Feb. 2012.
[32] P. K. Shukla, "On the normal boundary intersection method for generation of efficient front," in *Proc. 7th Int. Conf. Comput. Sci.* in Lecture Notes in Computer Science, vol. 4487, Y. Shi, G. D. van Albada, J. Dongarra, and P. M. A. Sloot, Eds. Berlin, Germany: Springer, 2007, pp. 310–317.
[33] I. Triguero *et al.*, "KEEL 3.0: An open source software for multi-stage analysis in data mining," *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 1238–1249, 2017.
[34] D. Dheeru and E. K. Taniskidou. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[35] G. Ratsch. (2001). *IDA Benchmark Repository*. [Online]. Available: http://ida.first.fhg.de/projects/bench/benchmarks.htm
[36] I. Guyon. (2006). *Datasets for the Agnostic Learning vs. Prior Knowledge Competition*. [Online]. Available: http://www.agnostic.inf.ethz.ch/datasets.php
[37] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
[38] S. S. Mullick, S. Datta, and S. Das, "Adaptive learning-based k-nearest neighbor classifiers with resilience to class imbalance," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2018.2812279.
[39] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *Proc. ICML Workshop Learn. Imbalanced Data Sets II*, vol. 2, 2003, pp. 1–2.
[40] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, 2001.