

## Dynamic class imbalance learning for incremental LPSVM

Shaoning Pang<sup>a,\*</sup>, Lei Zhu<sup>a</sup>, Gang Chen<sup>a</sup>, Abdolhossein Sarrafzadeh<sup>a</sup>, Tao Ban<sup>b</sup>, Daisuke Inoue<sup>b</sup>

<sup>a</sup> Department of Computing, Unitec Institute of Technology, Private Bag 92025, Auckland 1025, New Zealand

<sup>b</sup> National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan

### ARTICLE INFO

#### Article history:

Received 7 May 2012

Received in revised form 27 February 2013

Accepted 28 February 2013

#### Keywords:

Linear Proximal Support Vector Machines (LPSVMs)

Incremental learning

Data streams

Dynamic class imbalance learning (DCIL)

Weighted LPSVM

### ABSTRACT

Linear Proximal Support Vector Machines (LPSVMs), like decision trees, classic SVM, etc. are originally not equipped to handle drifting data streams that exhibit high and varying degrees of class imbalance. For online classification of data streams with imbalanced class distribution, we propose a dynamic class imbalance learning (DCIL) approach to incremental LPSVM (IncLPSVM) modeling. In doing so, we simplify a computationally non-renewable weighted LPSVM to several core matrices multiplying two simple weight coefficients. When data addition and/or retirement occurs, the proposed DCIL-IncLPSVM<sup>1</sup> accommodates newly presented class imbalance by a simple matrix and coefficient updating, meanwhile ensures no discriminative information lost throughout the learning process. Experiments on benchmark datasets indicate that the proposed DCIL-IncLPSVM outperforms classic IncSVM and IncLPSVM in terms of *F*-measure and *G*-mean metrics. Moreover, our application to online face membership authentication shows that the proposed DCIL-IncLPSVM remains effective in the presence of highly dynamic class imbalance, which usually poses serious problems to previous approaches.

© 2013 Elsevier Ltd. All rights reserved.

### 1. Introduction

Class imbalance occurs frequently in datasets from many real-world applications, such as intrusion detection, medical diagnosis and web mining, etc. (He & Chen, 2008). A large number of studies (Hong, Chen, & Harris, 2007; Huang, Yang, King, & Lyu, 2006; Wu & Chang, 2005; Zhou & Liu, 2006) have been conducted for investigating the impact of class imbalance on supervised machine learning, on the basis of training a classifier directly from a static set of training data (called batch learning hereafter). However in the context of data stream mining where data samples are available at different times, class imbalance learning involves not just biased class data distribution, but also concept drift, target variable changing over time in unforeseen ways. It remains as an interesting and demanding research topic for many supervised learning approaches. This paper addresses the challenge on classic Linear Proximal Support Vector Machine (LPSVM), and proposes a novel incremental LPSVM for class imbalance robust data stream mining.

Batch LPSVM (Fung & Mangasarian, 2001b), similar to classic SVM, has a default assumption that training samples are distributed evenly among two classes. In reality, class imbalance is a common phenomenon in training data. One class known as the

majority class may have many more samples than the other class, called the minority class in the paper. In the presence of imbalanced class distribution, the separating plane of LPSVM biases easily to the minority class. As a consequence the classification accuracy of the majority class overwhelms that of the minority class. A serious class imbalance may even lead to the classifier completely lose its classification capability, classifying data samples all into the majority class. wLPSVM (Fung & Mangasarian, 2005; Zhuang et al., 2005) has been proposed to solve the class imbalance problem for batch LPSVM learning, in which the contribution of two classes is balanced by weights calculated on the basis of the class distribution ratio.

In the data stream environment as shown in Fig. 1, dynamic class imbalance occurs in that new data becomes available and old data becomes unwanted continuously over an indefinite (possibly infinite) lifetime, and the class distribution ratio (i.e., number of samples in class 1/number of samples in class 2) might become high and vary at different time points. Learning from such data stream, incremental learning algorithms (Pang, Ban, Kadobayashi, & Kasabov, 2012; Pang, Ozawa, & Kasabov, 2005) are desirable to pose a capability for dynamic class imbalance learning (DCIL), i.e. learning from data to adjust itself adaptively to accommodate varied class imbalances.

To upgrade the wLPSVM for DCIL, it is required to instantly update the LPSVM and its weights whenever data addition/retirement occurred. Updating weights for classic wLPSVM however is a challenge, because weights re-calculation for wLPSVM involves updating several matrix multiplications as detailed in Section 4. This paper proposes a new incremental learning of wLPSVM

\* Corresponding author. Tel.: +64 9 8154321.

E-mail address: [ppang@unitec.ac.nz](mailto:ppang@unitec.ac.nz) (S. Pang).

<sup>1</sup> Matlab source code is available at

<http://www.dmlri.info/index.php/incremental-learning.html>.

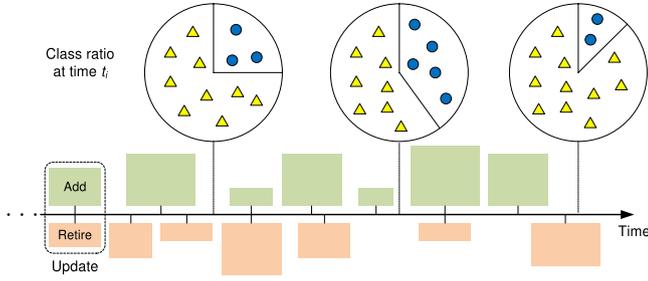


Fig. 1. Demonstration of class imbalance variation in data stream.

Table 1

Notations.

Notation	Descriptions
$\mathbf{X}$	Instance matrix, $\mathbf{X} \in \mathbb{R}^{n \times d}$
$\mathbf{Y}$	Class label vector, $\mathbf{Y} \in \mathbb{R}^{n \times 1}$
$d$	Dimension of instance
$n$	Number of training samples
$\mathbf{x}_i$	Column vector of $i$ th instance, $\mathbf{x}_i^T$ is the $i$ th row of $\mathbf{X}$
$\mathbf{w}, b$	Parameters in LPSVM class separating plane
$\xi$	Column vector of training errors
$\mathbf{O}$	Normal direction of LPSVM class separating plane, $\mathbf{O} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \in \mathbb{R}^{(d+1) \times 1}$
$\mathbf{D}$	Diagonal matrix of class labels, $\mathbf{D} = \text{diag}(\mathbf{Y})$
$\mathbf{E}$	Expanded instance matrix, $\mathbf{E} = [\mathbf{X} \quad -\mathbf{e}]$
$\mathbf{N}$	Diagonal weighting matrix
$\sigma_+/\sigma_-$	Class-wise weight for positive/negative class
$l_+/l_-$	Number of samples in positive/negative class

for DCIL, where non-stationary imbalanced stream data mining problem is explicitly formalized as learning from data chunks of imbalanced class ratio, which become available in an incremental manner. In the proposed DCIL-IncLPSVM, we derive a new expression of wLPSVM in which the weights are presented as two simple coefficients and LPSVM is represented as four core matrices. This has made the updating of weights straightforward and simple.

In general, the proposed DCIL-IncLPSVM updates its weights and LPSVM simultaneously whenever a chunk of data is presented/removed. The theoretical guarantee is that the updated wLPSVM is deterministically identical to the wLPSVM obtained from batch learning. Because the DCIL-IncLPSVM adapts its weights dynamically with the current class imbalance ratio, the proposed DCIL-IncLPSVM copes effectively any dynamic class imbalance throughout the incremental learning circle. For the convenience of method derivation and clarity of presentation, we summarize most notations used in the paper in Table 1.

The rest of this paper is organized as follows: Section 2 reviews the class imbalance impacts on SVMs and existing solutions. Section 3 introduces Linear Proximal Support Vector Machines (LPSVMs) and weighted LPSVM (wLPSVM) which are the fundamental of our work. Section 3.2 presents the proposed DCIL-IncLPSVM including proofs and derivations. Experiments and discussion are given in Section 5. Finally, we conclude this paper in Section 6.

## 2. Class imbalance impacts on SVMs and solutions

In literature, it has been well studied that SVMs can be degraded by high incidences of false negatives when the training samples of the target class are heavily out-numbered by the training samples of a nontarget class. The decision boundary of SVM can be skewed towards the minority (positive) class, which causes the generation of a high number of false-negative predictions, and lowers the performance on the positive class (Akbari, Kwek, & Japkowicz, 2004;

Wu & Chang, 2005). Existing class imbalance learning methods for reducing the effect of data imbalance can be summarized into three categories: sampling methods, weighting, and SVM ensemble.

### 2.1. Sampling method

The use of sampling methods for class imbalance learning considers the modification of an imbalanced data set by some mechanisms in order to provide a balanced distribution (He & Garcia, 2009). A variety of sampling methods have been developed so far. Undersampling, oversampling, re-sampling has the majority/minority class examples randomly removed/duplicated respectively until a particular class distribution ratio is met (Liu, Wu, & Zhou, 2006, 2009; Weiss, 2004). The synthetic minority oversampling technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) is a powerful method that has shown a great deal of success in SVM related applications (Gouripeddi et al., 2009; Maciejewski & Stefanowski, 2011; Wang, 2008). In general, sampling methods are applicable to various type of SVM algorithms being used (Batuwita & Palade, 2010) for data modeling, thus they are potential choices for SVM class imbalance learning, but they are not able to completely solve the class imbalance problem.

### 2.2. SVM ensemble method

Ensemble is seen as a promising method with the ability to improve the generalization performance of classification algorithms. For solving class imbalance, ensembles of classifiers consist of a set of individually trained classifiers whose predictions are combined to classify unknown samples (Zhai, Yang, Ma, & Ruan, 2010). Assume the class distribution ratio is  $1 : r$  in the data set. The majority class is then divided into  $r$  disjunct partitions. With each of these  $r$  partitions, the complete minority class is combined to create  $r$  balanced training sets.  $r$  SVMs are trained based on these training set. The classification outputs of these SVMs are combined by majority voting. In literature, partitioning ensembles (Dong & Han, 2005; Yan, Liu, Jin, & Hauptmann, 2003) including EasyEnsemble (Liu et al., 2006, 2009) and SVM trees (Pang, Kim, & Bang, 2003; Pang, Kim, & Sung-yang, 2005) all present as a modification of under-sampling method. Learn++ (Ditzler, Muhlbaier, & Polikar, 2010; Ditzler & Polikar, 2010; Muhlbaier, Topalis, & Polikar, 2004), builds a fundamental ensemble mechanism for adaptive class imbalance learning.

### 2.3. Weighting method

Instead of creating balanced data distributions through different sampling strategies, weighting methods tackle the imbalanced learning problem by using different weights to balance the contribution of the minority and majority classes. The weight can be considered as a numerical cost of classifying examples from one class to another. Typically, there is no cost for correct classification of either class and the cost of misclassifying minority examples is higher than the contrary case.

The weighting method can be embedded into the SVM objective function. A modified objective function assigns weights as two misclassification cost values  $\sigma_+$  and  $\sigma_-$  to each class respectively as

$$\min \frac{\|\mathbf{w}\|^2}{2} + C \left( \sigma_+ \sum_{i|y_i=+1} \xi_i + \sigma_- \sum_{i|y_i=-1} \xi_i \right) \quad (1)$$

$$\text{s.t. } y_i(\mathbf{x}_i^T \mathbf{w} + b) + \xi_i \geq 1 \quad \xi_i \geq 0 \quad \forall i \in \{1, \dots, n\},$$

has  $\sigma_+$  and  $\sigma_-$  corresponding to the class distribution ratio. By assigning a higher misclassification cost for the positive (minority) class examples than the negative (majority) class examples,

the effect of class imbalance could be reduced. Example methods include Different Error Costs (DEC) (Veropoulos, Campbell, & Cristianini, 1999) and fuzzy SVM for Class Imbalance Learning (FSVM-CIL) (Batuwita & Palade, 2010). An alternative weighting method can be applied directly to the SVM decision function as,

$$f(x) = \text{sign} \left( \sigma_+ \sum_{i|y_i=+1}^n \alpha_i y_i K(x_i, x) + \sigma_- \sum_{j|y_j=-1}^n \alpha_j y_j K(x_j, x) \right) \quad (2)$$

where  $\sigma_+$  and  $\sigma_-$  are weights on support vectors. To adjust the decision boundary's bias towards the minority (positive), higher weights are utilized for the positive support vectors (SVs) (Imam, Ting, & Kamruzzaman, 2006).

### 3. Linear proximal SVM

#### 3.1. Batch LPSVM

Let  $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  be a given training dataset,  $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$  and  $\mathbf{Y} = [y_1 \ y_2 \ \dots \ y_n]^T \in \{+1, -1\}^{n \times 1}$  as its instance matrix and corresponding label vector, respectively. A classic SVM (Vapnik, 1995) learns a class separating plane

$$\mathbf{x}^T \mathbf{w} + b = 0, \quad (3)$$

which lies midway between two parallel bounding planes formulated as,

$$\begin{aligned} \mathbf{x}^T \mathbf{w} + b &= +1 \\ \mathbf{x}^T \mathbf{w} + b &= -1. \end{aligned} \quad (4)$$

In practice, bounding planes (4) bound two classes often with some non-negative errors  $\xi_i$

$$\begin{aligned} \mathbf{x}_i^T \mathbf{w} + b + \xi_i &\geq +1 \quad \text{for } y_i = +1 \\ \mathbf{x}_i^T \mathbf{w} + b - \xi_i &\leq -1 \quad \text{for } y_i = -1. \end{aligned} \quad (5)$$

Here, the distance between these two planes equals to  $\frac{2}{\|\mathbf{w}\|}$ , which is called the “margin” in literature. The  $\mathbf{w}$  and  $b$  in (3) and (4) are obtained by solving an optimization problem

$$\min \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^n \xi_i \quad (6)$$

$$\text{s.t. } y_i(\mathbf{x}_i^T \mathbf{w} + b) + \xi_i \geq 1 \quad \xi_i \geq 0 \quad \forall i \in \{1, \dots, n\},$$

where  $\frac{\|\mathbf{w}\|^2}{2}$  is for maximizing the margin, and  $\sum_{i=1}^n \xi_i$  for minimizing the total training error. The regularization parameter  $C$  balances the importance of error and margin (Yamasaki & Ikeda, 2005). In practice, the dual problem of (6) is solved to obtain a SVM classifier (Cauwenberghs & Poggio, 2000; Karasuyama & Takeuchi, 2010).

LPSVM (Fung & Mangasarian, 2001b) models a binary classification as a regularized least square problem, which simplifies the above SVM optimization and results in an extremely efficient training algorithm. The optimization of LPSVM is given as,

$$\min \frac{C}{2} \|\xi\|^2 + \frac{1}{2} (\mathbf{w}^T \mathbf{w} + b^2) \quad (7)$$

$$\text{s.t. } \mathbf{D}(\mathbf{X}\mathbf{w} - \mathbf{e}b) + \xi = \mathbf{e},$$

where  $\xi \in \mathbb{R}^{n \times 1}$  refers to the vector of training errors,  $\mathbf{D} = \text{diag}(\mathbf{Y}) \in \mathbb{R}^{n \times n}$  denotes a diagonal matrix of class labels, and  $\mathbf{e} = [1, \dots, 1]^T \in \mathbb{R}^{n \times 1}$ . By (7), classic LPSVM seeks a class separating plane

$$\mathbf{x}^T \mathbf{w} - b = 0 \quad (8)$$

which lies midway between two parallel proximal planes

$$\begin{aligned} \mathbf{x}^T \mathbf{w} - b &= +1 \\ \mathbf{x}^T \mathbf{w} - b &= -1. \end{aligned} \quad (9)$$

In contrast to classic SVM, LPSVM in (7) replaces the inequality constraint of (6) with an equality condition. As a result, the planes in (9) are not bounding planes anymore, but can be seen as “proximal” planes around which the instances of each class are clustered, as we have

$$\begin{aligned} \mathbf{x}_i^T \mathbf{w} - b + \xi_i &= +1 \quad \text{for } y_i = +1 \\ \mathbf{x}_i^T \mathbf{w} - b - \xi_i &= -1 \quad \text{for } y_i = -1, \end{aligned} \quad (10)$$

and the error variable  $\xi_i$  is not necessary to be non-negative. These planes (9) are pushed as far apart as possible by the term of  $(\mathbf{w}^T \mathbf{w} + b^2)$  in the LPSVM optimization (7), and the total training error are minimized by the term of  $\|\xi\|^2$ . To summarize from a geometrical perspective, both SVM and LPSVM learn a separating plane that lies in the midway of two parallel planes which are pushed as far apart as possible. Parallel planes in classic SVM case bound two classes, and in LPSVM these planes perform as “proximal” planes around which the instances of each class are clustered.

To solve (7), the equality constraint is substituted to the objective function, thus (7) is transformed to an unconstrained optimization problem,

$$\min G = \frac{C}{2} \|\mathbf{D}(\mathbf{X}\mathbf{w} - \mathbf{e}b) - \mathbf{e}\|^2 + \frac{1}{2} (\mathbf{w}^T \mathbf{w} + b^2). \quad (11)$$

Set the partial derivatives of  $G$  to 0, we have

$$\begin{aligned} \frac{\partial G}{\partial \mathbf{w}} &= \mathbf{C}\mathbf{X}^T \mathbf{D}(\mathbf{D}(\mathbf{X}\mathbf{w} - \mathbf{e}b) - \mathbf{e}) + \mathbf{w} \\ &= \mathbf{C}\mathbf{X}^T \mathbf{X}\mathbf{w} + \mathbf{w} - \mathbf{C}\mathbf{X}^T \mathbf{e}b - \mathbf{C}\mathbf{X}^T \mathbf{D}\mathbf{e} \\ &= 0 \\ \frac{\partial G}{\partial b} &= -\mathbf{C}\mathbf{e}^T \mathbf{D}(\mathbf{D}(\mathbf{X}\mathbf{w} - \mathbf{e}b) - \mathbf{e}) + b \\ &= -\mathbf{C}\mathbf{e}^T \mathbf{X}\mathbf{w} + \mathbf{C}\mathbf{e}^T \mathbf{e}b + b + \mathbf{C}\mathbf{e}^T \mathbf{D}\mathbf{e} \\ &= 0, \end{aligned} \quad (12)$$

where  $\mathbf{D}\mathbf{D} = \mathbf{I}$  and  $\mathbf{e}^T \mathbf{e} = \|\mathbf{e}\|^2 = n$ .

Solving the linear system (12), the solution of LPSVM optimization (i.e., (7)) is obtained,

$$\begin{aligned} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} &= \begin{bmatrix} \mathbf{X}^T \mathbf{X} + \frac{\mathbf{I}}{C} & -\mathbf{X}^T \mathbf{e} \\ -\mathbf{e}^T \mathbf{X} & n + \frac{1}{C} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}^T \mathbf{D}\mathbf{e} \\ -\mathbf{e}^T \mathbf{D}\mathbf{e} \end{bmatrix} \\ &= \left[ \frac{\mathbf{I}}{C} + \begin{bmatrix} \mathbf{X}^T \\ -\mathbf{e}^T \end{bmatrix} \begin{bmatrix} \mathbf{X} & -\mathbf{e} \end{bmatrix} \right]^{-1} \begin{bmatrix} \mathbf{X}^T \mathbf{D}\mathbf{e} \\ -\mathbf{e}^T \mathbf{D}\mathbf{e} \end{bmatrix}. \end{aligned} \quad (13)$$

Let  $\mathbf{E} = [\mathbf{X} \quad -\mathbf{e}]$  and  $\mathbf{O} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$ , (13) can be formulated as

$$\mathbf{O} = \left( \frac{\mathbf{I}}{C} + \mathbf{E}^T \mathbf{E} \right)^{-1} \mathbf{E}^T \mathbf{D}\mathbf{e}. \quad (14)$$

To simplify (14), we use  $\mathbf{M}$  and  $\mathbf{v}$  to denote the matrix term  $\frac{\mathbf{I}}{C} + \mathbf{E}^T \mathbf{E}$  and vector term  $\mathbf{E}^T \mathbf{D}\mathbf{e}$  in (14) respectively. Then, (14) can be rewritten as

$$\mathbf{O} = \mathbf{M}^{-1} \mathbf{v}, \quad (15)$$

and the LPSVM decision function is obtained

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w} - b = [\mathbf{x}^T \quad -1] \mathbf{O} \begin{cases} > 0 & \Rightarrow y = +1 \\ < 0 & \Rightarrow y = -1. \end{cases} \quad (16)$$

In summary, the algorithm of batch LPSVM can be stated as Algorithm 1.

**Algorithm 1** Batch LPSVM Algorithm (Fung and Fung, 2001)

---

**Input:** Instance matrix  $\mathbf{X} \in \mathbb{R}^{n \times d}$ ; Class label vector  $\mathbf{Y} \in \mathbb{R}^{n \times 1}$ ; Regularization parameter  $C$ .  
**Output:**  $\mathbf{O} \in \mathbb{R}^{(d+1) \times 1}$ .  
1: Form identity matrix  $\mathbf{I} \in \mathbb{R}^{(d+1) \times (d+1)}$  and column vector  $\mathbf{e} = [1, \dots, 1]^T \in \mathbb{R}^{n \times 1}$ ;  
2: Expand the instance matrix  $\mathbf{X}$  into  $\mathbf{E}$ ,  $\mathbf{E} = \begin{bmatrix} \mathbf{X} & -\mathbf{e} \end{bmatrix}$ ;  
3: Transform the class label vector  $\mathbf{Y}$  into diagonal matrix  $\mathbf{D}$ ,  $\mathbf{D} = \text{diag}(\mathbf{Y})$ ;  
4: Compute  $\mathbf{M} = \frac{\mathbf{I}}{C} + \mathbf{E}^T \mathbf{E}$  and  $\mathbf{v} = \mathbf{E}^T \mathbf{D} \mathbf{e}$ ;  
5: Compute  $\mathbf{O} = \mathbf{M}^{-1} \mathbf{v}$ .

---

## 3.2. wLPSVM for class imbalance learning

The class imbalance problem arises when samples of the class of interest are relatively rare as compared with the other class. The total training error  $\|\xi\|^2$  in (7) comes from two classes, the error thus can be represented as  $\|\xi\|^2 = \|\xi_+\|^2 + \|\xi_-\|^2$ . In the presence of class imbalance, the LPSVM optimization (7) has  $\|\xi_+\|^2 \ll \|\xi_-\|^2$ . As a result, LPSVM shifts its positive class proximal plane away from the separating plane, which enlarges the margin at the price of an augmented  $\|\xi_+\|^2$ . Consequently, the separating plane biases to the positive class, and results in the worse recognition of the positive class.

For LPSVM class imbalance learning, Fung et al. proposed a weighted LPSVM (wLPSVM) (Fung & Mangasarian, 2005; Zhuang et al., 2005), in which the classic LPSVM optimization (7) is revised to

$$\min \frac{C}{2} \xi^T \mathbf{N} \xi + \frac{1}{2} (\mathbf{w}^T \mathbf{w} + b^2) \quad (17)$$

$$\text{s.t. } \mathbf{D}(\mathbf{X}\mathbf{w} - \mathbf{e}b) + \xi = \mathbf{e},$$

where  $\mathbf{N}$  is a diagonal weighting matrix with,

$$\mathbf{N}_{ii} = \begin{cases} \sigma_+ & \text{if } y_i = +1 \\ \sigma_- & \text{if } y_i = -1, \end{cases} \quad (18)$$

in which the class-wise weight  $\sigma$  is used,  $\sigma_+$  for the positive class and  $\sigma_-$  for the negative class, to balance the impacts of two classes to the LPSVM separating plane. In practice,  $\sigma$  is determined by the number of samples for each class. For example in Tao and Ji (2007),  $\sigma_+$  and  $\sigma_-$  are calculated as the ratio of the contrary class size (i.e.,  $l_-$  or  $l_+$ ) to the size of the whole dataset,

$$\begin{aligned} \sigma_+ &= l_- / (l_+ + l_-) \\ \sigma_- &= l_+ / (l_+ + l_-). \end{aligned} \quad (19)$$

By a similar approach as (11)–(14), (17) can be solved and the wLPSVM solution is obtained as,

$$\mathbf{O} = \left( \frac{\mathbf{I}}{C} + \mathbf{E}^T \mathbf{N} \mathbf{E} \right)^{-1} \mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e}. \quad (20)$$

Algorithm 2 presents the batch wLPSVM algorithm that implements (20).

**Algorithm 2** Batch wLPSVM Algorithm (Fung and Mangasarian, 2005)

---

**Input:** Instance matrix  $\mathbf{X} \in \mathbb{R}^{n \times d}$ ; Class label vector  $\mathbf{Y} \in \mathbb{R}^{n \times 1}$ ; Regularization parameter  $C$ .  
**Output:**  $\mathbf{O} \in \mathbb{R}^{(d+1) \times 1}$ .  
1: Calculate weights  $\sigma_+$  and  $\sigma_-$  following weighting function such as (19);  
2: Form weight matrix  $\mathbf{N}$  following (18);  
3: Expand the instance matrix  $\mathbf{X}$  into  $\mathbf{E}$ ,  $\mathbf{E} = \begin{bmatrix} \mathbf{X} & -\mathbf{e} \end{bmatrix}$ ;  
4: Transform the class label vector  $\mathbf{Y}$  into diagonal matrix  $\mathbf{D}$ ,  $\mathbf{D} = \text{diag}(\mathbf{Y})$ ;  
5: Compute  $\mathbf{M} = \frac{\mathbf{I}}{C} + \mathbf{E}^T \mathbf{N} \mathbf{E}$  and  $\mathbf{v} = \mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e}$ ;  
6: Compute  $\mathbf{O} = \mathbf{M}^{-1} \mathbf{v}$ .

---

The incremental learning of LPSVM is to update existing LPSVM model by retiring old data and/or adding new data simultaneously. Assume that the current LPSVM model on  $\mathcal{S}$  is obtained from (14),  $\mathcal{S}_r$  is an “old” subset of  $\mathcal{S}$  that needs to be retired, and  $\mathcal{S}_a$  a “new” set of data that needs to be added.

Fung and Mangasarian (2001a) proposed an incremental LPSVM model (InclPSVM) (21) to handle the retirement of  $\mathcal{S}_r$  and the addition of  $\mathcal{S}_a$ ,

$$\begin{aligned} \mathbf{O}' &= \left( \frac{\mathbf{I}}{C} + \mathbf{E}^T \mathbf{E} - \mathbf{E}_r^T \mathbf{E}_r + \mathbf{E}_a^T \mathbf{E}_a \right)^{-1} (\mathbf{E}^T \mathbf{D} \mathbf{e} - \mathbf{E}_r^T \mathbf{D}_r \mathbf{e} + \mathbf{E}_a^T \mathbf{D}_a \mathbf{e}) \\ &= (\mathbf{M} - \mathbf{E}_r^T \mathbf{E}_r + \mathbf{E}_a^T \mathbf{E}_a)^{-1} (\mathbf{v} - \mathbf{E}_r^T \mathbf{D}_r \mathbf{e} + \mathbf{E}_a^T \mathbf{D}_a \mathbf{e}), \end{aligned} \quad (21)$$

where  $\mathbf{E}_r \in \mathbb{R}^{n_r \times (d+1)}$ ,  $\mathbf{D}_r \in \mathbb{R}^{n_r \times n_r}$ ,  $\mathbf{E}_a \in \mathbb{R}^{n_a \times (d+1)}$  and  $\mathbf{D}_a \in \mathbb{R}^{n_a \times n_a}$ . This method can update accurately a batch LPSVM, however it considers no dynamic class imbalance of data streams.

## 4. Proposed incremental LPSVM

For dynamic class imbalance learning (DCIL), the idea of the proposed DCIL-InclPSVM is to update the batch wLPSVM described above over newly presented data, by performing simultaneously the incremental learning of its LPSVM and weights.

For the convenience of method derivations, we introduce first two lemmas on basic matrix decomposition:

**Lemma 1.** Let  $\begin{bmatrix} \mathbf{X} & \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_a & \mathbf{y}_a \\ \mathbf{x}_b & \mathbf{y}_b \end{bmatrix}$ ,  $\mathbf{E} = \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix} = \begin{bmatrix} \mathbf{x}_a & -\mathbf{e} \\ \mathbf{x}_b & -\mathbf{e} \end{bmatrix}$ ,  $\mathbf{D} = \begin{bmatrix} \mathbf{D}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_b \end{bmatrix}$  and  $\mathbf{N} = \begin{bmatrix} \mathbf{N}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_b \end{bmatrix}$ . Then,

$$\begin{aligned} \mathbf{E}^T \mathbf{N} \mathbf{E} &= \mathbf{E}_a^T \mathbf{N}_a \mathbf{E}_a + \mathbf{E}_b^T \mathbf{N}_b \mathbf{E}_b \\ \mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e} &= \mathbf{E}_a^T \mathbf{D}_a \mathbf{N}_a \mathbf{e} + \mathbf{E}_b^T \mathbf{D}_b \mathbf{N}_b \mathbf{e} \\ \mathbf{E}^T \mathbf{E} &= \mathbf{E}_a^T \mathbf{E}_a + \mathbf{E}_b^T \mathbf{E}_b \\ \mathbf{E}^T \mathbf{e} &= \mathbf{E}_a^T \mathbf{e} + \mathbf{E}_b^T \mathbf{e}. \end{aligned} \quad (22)$$

**Lemma 2.** Let  $\mathcal{S}$  be a dataset with  $n$  samples, i.e.,  $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ .  $\eta = [1 \ 2 \ \dots \ i \ \dots \ j \ \dots \ n]$  identifies a sequence of data samples. Applying  $\eta$  to the dataset, we form instance matrix  $\mathbf{X}^\eta = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_i \ \dots \ \mathbf{x}_j \ \dots \ \mathbf{x}_n]^T$  and label vector  $\mathbf{Y}^\eta = [y_1 \ y_2 \ \dots \ y_i \ \dots \ y_j \ \dots \ y_n]^T$ , and corresponding matrices  $\mathbf{E}^\eta$ ,  $\mathbf{D}^\eta$  and  $\mathbf{N}^\eta$ . Given another sample sequence  $\eta' = [1 \ 2 \ \dots \ j \ \dots \ i \ \dots \ n]$  and corresponding  $\mathbf{E}^{\eta'}$ ,  $\mathbf{D}^{\eta'}$  and  $\mathbf{N}^{\eta'}$ . Then,

$$\begin{aligned} \mathbf{E}^{\eta T} \mathbf{N}^\eta \mathbf{E}^\eta &= \mathbf{E}^{\eta' T} \mathbf{N}^{\eta'} \mathbf{E}^{\eta'} \\ \mathbf{E}^{\eta T} \mathbf{D}^\eta \mathbf{N}^\eta \mathbf{e} &= \mathbf{E}^{\eta' T} \mathbf{D}^{\eta'} \mathbf{N}^{\eta'} \mathbf{e} \\ \mathbf{E}^{\eta T} \mathbf{E}^\eta &= \mathbf{E}^{\eta' T} \mathbf{E}^{\eta'} \\ \mathbf{E}^{\eta T} \mathbf{e} &= \mathbf{E}^{\eta' T} \mathbf{e} \end{aligned} \quad (23)$$

Lemma 2 implies that the equations in (23) hold with respect to arbitrary sample sequences.

## 4.1. DCIL derivation

Given an initial wLPSVM model (20) on dataset  $\mathcal{S}$ . If  $\mathcal{S}_l$  denotes the set of data that remains after retiring  $\mathcal{S}_r$ , and  $\mathcal{S}'$  denotes the

updated dataset after the addition and retirement, then we have

$$\begin{aligned}\mathcal{S} &= \mathcal{S}_l \cup \mathcal{S}_r \\ \mathcal{S}' &= \mathcal{S}_l \cup \mathcal{S}_a.\end{aligned}\quad (24)$$

Then, the incremental learning of wLPSVM is to compute a updated wLPSVM model  $\mathcal{O}'$  based on  $\mathcal{S}_a$ ,  $\mathcal{S}_r$ , and a trained wLPSVM model (i.e., (20)) on the current dataset  $\mathcal{S}$ .

By (20), a batch wLPSVM on the updated dataset  $\mathcal{S}'$  can be written as,

$$\mathcal{O}' = \left( \frac{\mathbf{I}}{C} + \mathbf{E}'^T \mathbf{N}' \mathbf{E}' \right)^{-1} \mathbf{E}'^T \mathbf{D}' \mathbf{N}' \mathbf{e}.\quad (25)$$

In (25), the weighting matrix  $\mathbf{N}$  can be simply updated by,

$$\mathbf{N}'_{ii} = \begin{cases} \sigma'_+ = l'_- / (l'_+ + l'_-) & \text{if } y'_i = +1 \\ \sigma'_- = l'_+ / (l'_+ + l'_-) & \text{if } y'_i = -1, \end{cases}\quad (26)$$

in which  $l'_- = l_- - l_{r-} + l_{a-}$  and  $l'_+ = l_+ - l_{r+} + l_{a+}$ . However, its problematic to update the matrix multiplications  $\mathbf{E}^T \mathbf{N} \mathbf{E}$  and  $\mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e}$ . To overcome this difficulty, we propose to simplify (25) by Proposition 1, transforming the weighting matrix  $\mathbf{N}$  to two simple weight coefficients  $\sigma_+$  and  $\sigma_-$ .

**Proposition 1.** Given a wLPSVM model (20) over dataset  $\mathcal{S}$ , and let  $\mathbf{M}_+ = \mathbf{E}_+^T \mathbf{E}_+$ ,  $\mathbf{M}_- = \mathbf{E}_-^T \mathbf{E}_-$ ,  $\mathbf{v}_+ = \mathbf{E}_+^T \mathbf{e}$  and  $\mathbf{v}_- = \mathbf{E}_-^T \mathbf{e}$ , then the wLPSVM model (20) can be reformulated as,

$$\mathcal{O} = \left( \frac{\mathbf{I}}{C} + \sigma_+ \mathbf{M}_+ + \sigma_- \mathbf{M}_- \right)^{-1} (\sigma_+ \mathbf{v}_+ - \sigma_- \mathbf{v}_-).\quad (27)$$

**Proof.** As  $\mathcal{S}$  is a 2-class dataset decomposable into  $\mathcal{S}_+$  and  $\mathcal{S}_-$ , we apply Lemmas 2 and 1 to the term  $\mathbf{E}^T \mathbf{N} \mathbf{E}$  in (20), and have

$$\mathbf{E}^T \mathbf{N} \mathbf{E} = \mathbf{E}_+^T \mathbf{N}_+ \mathbf{E}_+ + \mathbf{E}_-^T \mathbf{N}_- \mathbf{E}_-.\quad (28)$$

As  $\mathbf{N}_+ = \sigma_+ \mathbf{I}$  and  $\mathbf{N}_- = \sigma_- \mathbf{I}$ , term  $\mathbf{E}^T \mathbf{N} \mathbf{E}$  can be further written as,

$$\mathbf{E}^T \mathbf{N} \mathbf{E} = \sigma_+ \mathbf{E}_+^T \mathbf{E}_+ + \sigma_- \mathbf{E}_-^T \mathbf{E}_-.\quad (29)$$

Similarly, applying Lemmas 2 and 1 to the term  $\mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e}$  in (20), we have

$$\mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e} = \mathbf{E}_+^T \mathbf{D}_+ \mathbf{N}_+ \mathbf{e} + \mathbf{E}_-^T \mathbf{D}_- \mathbf{N}_- \mathbf{e}.\quad (30)$$

As  $\mathbf{D}_+ = \mathbf{I}$  and  $\mathbf{D}_- = -\mathbf{I}$ , also  $\mathbf{N}_+ = \sigma_+ \mathbf{I}$  and  $\mathbf{N}_- = \sigma_- \mathbf{I}$ , we have

$$\mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e} = \sigma_+ \mathbf{E}_+^T \mathbf{e} - \sigma_- \mathbf{E}_-^T \mathbf{e}.\quad (31)$$

Substituting (29) and (31) into (20), and replacing  $\mathbf{E}_+^T \mathbf{E}_+$ ,  $\mathbf{E}_-^T \mathbf{E}_-$ ,  $\mathbf{E}_+^T \mathbf{e}$  and  $\mathbf{E}_-^T \mathbf{e}$  with  $\mathbf{M}_+$ ,  $\mathbf{M}_-$ ,  $\mathbf{v}_+$  and  $\mathbf{v}_-$  respectively, we obtain

$$\mathcal{O} = \left( \frac{\mathbf{I}}{C} + \sigma_+ \mathbf{M}_+ + \sigma_- \mathbf{M}_- \right)^{-1} (\sigma_+ \mathbf{v}_+ - \sigma_- \mathbf{v}_-).\quad \square\quad (32)$$

Applying Proposition 1 to (25), we rewrite the updated wLPSVM model (25) as,

$$\mathcal{O}' = \left( \frac{\mathbf{I}}{C} + \sigma'_+ \mathbf{M}'_+ + \sigma'_- \mathbf{M}'_- \right)^{-1} (\sigma'_+ \mathbf{v}'_+ - \sigma'_- \mathbf{v}'_-).\quad (33)$$

When data addition and/or retirement is incurred,  $\sigma'$  in (33) can be re-calculated by (26). So, we consider only the updating of  $\mathbf{M}_+$ ,  $\mathbf{M}_-$ ,  $\mathbf{v}_+$  and  $\mathbf{v}_-$ .

As  $\mathcal{S}_+$  by (24) is decomposable into  $\mathcal{S}_{l+}$  and  $\mathcal{S}_{r+}$ , and so for  $\mathcal{S}'_+$  decomposable into  $\mathcal{S}_{l+}$  and  $\mathcal{S}_{a+}$ , we can apply Lemmas 2 and 1 to

decompose  $\mathbf{M}_+$  and  $\mathbf{M}'_+$ , and have

$$\mathbf{M}_+ = \mathbf{E}_+^T \mathbf{E}_+ = \mathbf{E}_{l+}^T \mathbf{E}_{l+} + \mathbf{E}_{r+}^T \mathbf{E}_{r+}\quad (34)$$

$$\mathbf{M}'_+ = \mathbf{E}'_+^T \mathbf{E}'_+ = \mathbf{E}_{l+}^T \mathbf{E}_{l+} + \mathbf{E}_{a+}^T \mathbf{E}_{a+}\quad (35)$$

(35) minus (34), we obtain the updating of  $\mathbf{M}_+$  as,

$$\mathbf{M}'_+ = \mathbf{M}_+ - \mathbf{E}_{r+}^T \mathbf{E}_{r+} + \mathbf{E}_{a+}^T \mathbf{E}_{a+}.\quad (36)$$

By an analogous process, we can update  $\mathbf{M}_-$ ,  $\mathbf{v}_+$  and  $\mathbf{v}_-$  respectively as,

$$\mathbf{M}'_- = \mathbf{M}_- - \mathbf{E}_{r-}^T \mathbf{E}_{r-} + \mathbf{E}_{a-}^T \mathbf{E}_{a-}$$

$$\mathbf{v}'_+ = \mathbf{v}_+ - \mathbf{E}_{r+}^T \mathbf{e} + \mathbf{E}_{a+}^T \mathbf{e}\quad (37)$$

$$\mathbf{v}'_- = \mathbf{v}_- - \mathbf{E}_{r-}^T \mathbf{e} + \mathbf{E}_{a-}^T \mathbf{e}.$$

## 4.2. Algorithm script

For DCIL of data streams, the matrix multiplication terms in (20) are divided into a set of class-wise matrices in (32) whose updating for incremental learning is straightforward. Most importantly, current class imbalance level is modeled here as two simple coefficients, and utilized in every step of LPSVM updating. This empowers the incremental LPSVM with the capability for DCIL. The pseudo-code of the proposed DCIL-IncLPSVM is given in Algorithm 3.

---

### Algorithm 3 The proposed DCIL-IncLPSVM Algorithm

---

**Input:** initial DCIL-IncLPSVM model:

$\{\mathcal{O}, \mathbf{M}_+, \mathbf{M}_-, \mathbf{v}_+, \mathbf{v}_-, l_+, l_-, C\}$ ;  $\mathbf{X}_r$  and  $\mathbf{Y}_r$  to retire;

$\mathbf{X}_a$  and  $\mathbf{Y}_a$  to add; and regularization parameter  $C$ .

**Output:** updated DCIL-IncLPSVM model:

$\{\mathcal{O}', \mathbf{M}'_+, \mathbf{M}'_-, \mathbf{v}'_+, \mathbf{v}'_-, l'_+, l'_-, C\}$ .

- 1: **if** input model is empty **then**
  - 2:   Form  $\mathbf{E}_{a+}$  and  $\mathbf{E}_{a-}$  according to  $\mathbf{X}_a$  and  $\mathbf{Y}_a$
  - 3:   Compute  $\mathbf{M}_+ = \mathbf{E}_{a+}^T \mathbf{E}_{a+}$ ,  $\mathbf{M}_- = \mathbf{E}_{a-}^T \mathbf{E}_{a-}$ ,  $\mathbf{v}_+ = \mathbf{E}_{a+}^T \mathbf{e}$  and  $\mathbf{v}_- = \mathbf{E}_{a-}^T \mathbf{e}$ ;
  - 4:   Compute  $\sigma_+$  and  $\sigma_-$  according to (19);
  - 5:   Compute  $\mathcal{O} = (\frac{\mathbf{I}}{C} + \sigma_+ \mathbf{M}_+ + \sigma_- \mathbf{M}_-)^{-1} (\sigma_+ \mathbf{v}_+ - \sigma_- \mathbf{v}_-)$  according to (27);
  - 6: **else**
  - 7:   Generate  $\mathbf{E}_{a+}$ ,  $\mathbf{E}_{a-}$ ,  $\mathbf{E}_{r+}$  and  $\mathbf{E}_{r-}$  according to  $\mathbf{X}_a$ ,  $\mathbf{Y}_a$ ,  $\mathbf{X}_r$  and  $\mathbf{Y}_r$ ;
  - 8:   Compute  $\mathbf{M}'_+, \mathbf{M}'_-, \mathbf{v}'_+, \mathbf{v}'_-$  according to (36) and (37)
  - 9:   Compute  $l'_+ = l_+ - l_{r+} + l_{a+}$  and  $l'_- = l_- - l_{r-} + l_{a-}$ ;
  - 10:   Compute  $\sigma'_+$  and  $\sigma'_-$ ; according to (26);
  - 11:   Compute  $\mathcal{O}' = (\frac{\mathbf{I}}{C} + \sigma'_+ \mathbf{M}'_+ + \sigma'_- \mathbf{M}'_-)^{-1} (\sigma'_+ \mathbf{v}'_+ - \sigma'_- \mathbf{v}'_-)$  according to (25).
  - 12: **end if**
- 

## 4.3. Extension to multiple classes

Assume we have  $k$ -class streaming data, the previously studied multi-category incremental LPSVM (McIncLPSVM) (Tveit & Hetl, 2003) learns and updates  $k$  individual binary IncLPSVMs in a one-against-rest manner without balancing training errors from two classes. The classification of a test sample is conducted by comparing the its decision values on each class. However, those binary one-against-rest IncLPSVMs are all built on a 1-to- $(k-1)$  partition data which poses normally a serious class imbalance, therefore each IncLPSVM suffers from the class imbalance problem and the overall accuracy of McIncLPSVM is affected consequently. To mitigate this difficulty, we extend in the following the above proposed DCIL to McDCIL-IncLPSVM for multi-class classification.

Let  $\mathcal{S}$  be a  $k$ -class initial training dataset including  $n$  samples in which the classes are labeled as  $1, 2, \dots, k$ ,  $\mathbf{X} \in \mathbb{R}^{n \times d}$  and  $\mathbf{Y} \in \{1, 2, \dots, k\}^{n \times 1}$  as its instance matrix and corresponding label vector, respectively. To achieve multi-class classification on  $\mathcal{S}$ , for any class  $j \in \{1, 2, \dots, k\}$  we use an individual binary DCIL-IncLPSVM

to separate the class  $j$  from the rest as

$$\mathbf{O}_{cj} = \left( \frac{\mathbf{I}}{C} + \sigma_+ \mathbf{M}_+ + \sigma_- \mathbf{M}_- \right)^{-1} (\sigma_+ \mathbf{v}_+ - \sigma_- \mathbf{v}_-), \quad (38)$$

in which samples from class  $j$  are taken as positive and the rest samples are taken as negative. Thus, we have

$$\begin{aligned} \mathbf{M}_+ &= \mathbf{E}_{cj}^T \mathbf{E}_{cj} = \mathbf{M}_{cj} \\ \mathbf{v}_+ &= \mathbf{E}_{cj}^T \mathbf{e} = \mathbf{v}_{cj} \end{aligned} \quad (39)$$

and

$$\begin{aligned} \sigma_+ &= (l - l_{cj})/l \\ \sigma_- &= l_{cj}/l. \end{aligned} \quad (40)$$

Applying Lemmas 2 and 1 to decompose term  $\mathbf{M}_-$  and  $\mathbf{v}_-$  in (38), we obtain

$$\begin{aligned} \mathbf{M}_- &= \sum_{i=1}^{j-1} \mathbf{E}_{ci}^T \mathbf{E}_{ci} + \sum_{i=j+1}^k \mathbf{E}_{ci}^T \mathbf{E}_{ci} = \mathbf{E}^T \mathbf{E} - \mathbf{E}_{cj}^T \mathbf{E}_{cj} = \mathbf{M} - \mathbf{M}_{cj} \\ \mathbf{v}_- &= \sum_{i=1}^{j-1} \mathbf{E}_{ci}^T \mathbf{e} + \sum_{i=j+1}^k \mathbf{E}_{ci}^T \mathbf{e} = \mathbf{E}^T \mathbf{e} - \mathbf{E}_{cj}^T \mathbf{e} = \mathbf{v} - \mathbf{v}_{cj}. \end{aligned} \quad (41)$$

Substituting (39)–(41) into (38), we have the DCIL-IncLPSVM for class  $j$  as,

$$\mathbf{O}_{cj} = \left( \frac{\mathbf{I}}{C} + \frac{l_{cj}}{l} \mathbf{M} + \frac{l - 2l_{cj}}{l} \mathbf{M}_{cj} \right)^{-1} \left( \mathbf{v}_{cj} - \frac{l_{cj}}{l} \mathbf{v} \right). \quad (42)$$

Given dataset  $\mathcal{S}_r$  to be retried and  $\mathcal{S}_a$  to be added. The above DCIL-IncLPSVM for incremental learning is supposed to be updated for:

(1) any existing class (i.e.,  $\forall j \in \{1, 2, \dots, k\}$ ), we update terms in (42) as

$$\begin{aligned} \mathbf{M}' &= \mathbf{M} - \mathbf{E}_r^T \mathbf{E}_r + \mathbf{E}_a^T \mathbf{E}_a \\ \mathbf{M}'_{cj} &= \mathbf{M}_{cj} - \mathbf{E}_{cjr}^T \mathbf{E}_{cjr} + \mathbf{E}_{cja}^T \mathbf{E}_{cja} \\ \mathbf{v}' &= \mathbf{v} - \mathbf{E}_r^T \mathbf{e} + \mathbf{E}_a^T \mathbf{e} \\ \mathbf{v}'_{cj} &= \mathbf{v}_{cj} - \mathbf{E}_{cjr}^T \mathbf{e} + \mathbf{E}_{cja}^T \mathbf{e} \\ l' &= l - l_r + l_a \\ l'_{cj} &= l_{cj} - l_{cjr} + l_{cja}, \end{aligned} \quad (43)$$

and obtain the updated model as

$$\mathbf{O}'_{cj} = \left( \frac{\mathbf{I}}{C} + \frac{l'_{cj}}{l'} \mathbf{M}' + \frac{l' - 2l'_{cj}}{l'} \mathbf{M}'_{cj} \right)^{-1} \left( \mathbf{v}'_{cj} - \frac{l'_{cj}}{l'} \mathbf{v}' \right), \quad (44)$$

separating the class  $r$  from the rest.

(2) any new class (i.e.,  $\forall j > k$ ), we construct a new model  $\mathbf{O}'_{cj}$  to distinguish the class from the rest as

$$\mathbf{O}'_{cj} = \left( \frac{\mathbf{I}}{C} + \frac{l_{cj}}{l} \mathbf{M}' + \frac{l' - 2l_{cj}}{l'} \mathbf{M}'_{cj} \right)^{-1} \left( \mathbf{v}_{cj} - \frac{l_{cj}}{l'} \mathbf{v}' \right), \quad (45)$$

where  $\mathbf{M}_{cj} = \mathbf{E}_{cja}^T \mathbf{E}_{cja}$  and  $\mathbf{v}_{cj} = \mathbf{E}_{cja}^T \mathbf{e}$ .

In summary, the pseudo-code of the proposed DCIL-IncLPSVM for multi-class classification is given in Algorithm 4.

---

#### Algorithm 4

The proposed multi-class DCIL-IncLPSVM (McDCIL-IncLPSVM)

---

**Input:** multi-class DCIL-IncLPSVM model,  $\mathbf{X}_r$  and  $\mathbf{Y}_r$  to retire,  $\mathbf{X}_a$  and  $\mathbf{Y}_a$  to add and regularization parameter  $C$ .

**Output:** multi-class DCIL-IncLPSVM model  $\{\mathbf{O}_{cj}, \mathbf{M}_{cj}, \mathbf{v}_{cj}, l_{cj}, \mathbf{M}, \mathbf{v}, l, C, \mathbf{I}\}$  or  $\{\mathbf{O}'_{cj}, \mathbf{M}'_{cj}, \mathbf{v}'_{cj}, l'_{cj}, \mathbf{M}', \mathbf{v}', l', C, \mathbf{I}\}$ .

```

1: if input model is empty then
2:   Generate  $\mathbf{I}, \mathbf{E}$  and  $l$  according to  $\mathbf{X}_a$  and  $\mathbf{Y}_a$ ;
3:   Compute  $\mathbf{M} = \mathbf{E}^T \mathbf{E}$  and  $\mathbf{v} = \mathbf{E}^T \mathbf{e}$ ;
4:   for  $j = 1$  to  $k$  do
5:     Generate  $\mathbf{E}_{cj}$  and  $l_{cj}$  according to  $\mathbf{X}_{c_j}$  and  $\mathbf{Y}_{c_j}$ ;
6:     Compute  $\mathbf{M}_{cj}$  and  $\mathbf{v}_{cj}$  according to (39);
7:     Compute  $\mathbf{O}_{cj}$  according to (42);
8:   end for
9: else
10:  Generate  $\mathbf{E}_a, \mathbf{E}_r, l_a$  and  $l_r$  according to  $\mathbf{X}_a, \mathbf{Y}_a, \mathbf{X}_r$  and  $\mathbf{Y}_r$ ;
11:  Update  $\mathbf{M}, \mathbf{v}$  and  $l$  according to (43);
12:  for  $j = 1$  to  $k$  do
13:    Generate  $\mathbf{E}_{cja}, \mathbf{E}_{cjr}, l_{cja}$  and  $l_{cjr}$  according to  $\mathbf{X}_a, \mathbf{Y}_a, \mathbf{X}_r$  and  $\mathbf{Y}_r$ ;
14:    Update  $\mathbf{M}_{cj}, \mathbf{v}_{cj}$  and  $l_{cj}$  according to (43);
15:    Compute  $\mathbf{O}'_{cj}$  according to (44);
16:  end for
17:  for all  $j > k$  do
18:    Generate  $\mathbf{E}_{cja}$  and  $l_{cja}$  according to  $\mathbf{X}_a$  and  $\mathbf{Y}_a$ ;
19:    Compute  $\mathbf{M}_{cj} = \mathbf{E}_{cja}^T \mathbf{E}_{cja}$  and  $\mathbf{v}_{cj} = \mathbf{E}_{cja}^T \mathbf{e}$ ;
20:    Compute  $\mathbf{O}_{cj}$  according to (45);
21:  end for
22: end if

```

---

## 5. Experiments and discussions

In this section, the effectiveness of our algorithm in learning from imbalanced data is testified. We compared the performance of the proposed DCIL-IncLPSVM with other benchmark linear classifiers, which include the classic linear SVM, classic LPSVM, and the SMOTE, a widely adopted class-imbalance countermeasure (Chawla et al., 2002). The comparison is conducted in two scenarios: static datasets with various degrees of class imbalance and data streams with dynamic class imbalances, on a Dell server with Intel Xeon 2.66 GHz CPU, 1T Ram and RedHat Linux OS.

### 5.1. Experimental setup

For the purpose of investigating the impact of different levels of class imbalance on the training of a classifier and verifying the ability of proposed DCIL-IncLPSVM to overcome it, samples from original benchmark datasets are grouped and selected artificially to form binary-class experimental datasets with predefined class ratio.

For static class imbalance test, the whole training dataset is presented to learners in one batch. For incremental learning test, we first construct initial models using 10% of the data, and the remaining 90% training instances are divided equally into 9 subsets and presented to the learners sequentially. As the incremental learning proceeds, the learners are fed with subsets of new data in a sequential manner until all data from the dataset are consumed.

### 5.2. Performance measurements

For evaluating the algorithm performance on imbalanced datasets, the normal metric of the overall accuracy is no longer sufficient (Chen & He, 2009; He & Garcia, 2009). Here several performance metrics based on the confusion matrix is utilized to give a more comprehensive and objective assessment. Specifically, for a binary classifier the confusion matrix is consisted of True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). In our experiments, “positive” refers to the minority class whereas “negative” is taken as the majority class.

**Table 2**  
Dataset description.

Dataset	Class min./maj.	#Var.	Exp.	Imbalance ratio #maj./#min.	Training set		Testing set	
					#Neg.	#Pos.	#Neg.	#Pos.
Breast wisconsin	Abnormal/normal	9	1	1	20	20	40	40
			2	8	160	20		
			3	20	400	20		
Car	Very good/remainder	6	1	1	20	20	40	40
			2	10	200	20		
			3	80	1600	20		
Abalone	Less than 5 rings/remainder	8	1	1	20	20	40	40
			2	10	200	20		
			3	200	4000	20		
Internet advertisements	Ad./not ad.	1558	1	1	100	100	250	250
			2	4	400	100		
			3	16	1600	100		
Cover type	Type 5/remainder	54	1	1	1000	1000	5000	5000
			2	8	8000	1000		
			3	80	80000	1000		
Cardiotocography	Pathologic/remainder	21	1	1	50	50	100	100
			2	8	400	50		
			3	32	1600	50		

In this paper, we evaluate classification performance through five metrics: sensitivity, specificity,  $F$ -measure, relative sensitivity (RS) and  $G$ -mean. Sensitivity and specificity measure the accuracy on positive and negative samples respectively

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (46)$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Normally, high sensitivity and specificity implies high classification performance.  $F$ -measure is defined as

$$F\text{-measure} = \frac{(1 + \beta)^2 \times \text{Recall} \times \text{Precision}}{\beta^2 \times \text{Recall} + \text{Precision}}, \quad (47)$$

where

$$\text{Precision} = \frac{TP}{TP + FP} \quad (48)$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

and  $\beta$  is a coefficient that balances the relative importance of precision and recall. Following common practice, we assign  $\beta$  to 1.  $G$ -mean is another metric widely adopted for evaluating classifiers on imbalanced dataset, which is defined as

$$G\text{-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}. \quad (49)$$

For statistical hypothesis testing, we conduct paired  $T$ -test on  $F$ -measure and  $G$ -mean for each algorithm in comparison to the proposed DCIL-IncLPSVM, and calculate  $p$ -value under the level of 95% confidence.

### 5.3. Static class imbalance robustness tests

In this section, we compare the performance of learners on datasets with varied class imbalance ratio, dimensionality and number of samples.

Six publicly known UCI datasets are utilized to conduct the comparison. Table 2 shows the characteristics of these datasets. For each dataset, 3 experiments (i.e., Exp.1–3) are defined with 3 different degrees of class imbalance. Each experiment consists of 20 rounds of independent tests. For each test, training sets are randomly selected and presented to LPSVM, SVM (Chang & Lin,

2011), SMOTE + LPSVM (SMOTE), DCIL-IncLPSVM (DCIL), weighted LPSVM, weighted SVM (SVM-w) (Chang & Lin, 2011) and partitioning ensemble SVM (SVM-ensemble), respectively; and testing sets in balanced class distribution are used to evaluate the performance of each classifier.

Tables 3 and 4 give the experimental results, where performance on each measurement is shown as 'average value  $\pm$  standard division,' and time costs for training is given in CPU seconds. The  $p$ -value on  $F$ -measure and  $G$ -mean is categorized as  $0.05 > p > 0.01$ ,  $0.01 > p > 0.001$  and  $0.001 > p$ , and presented at the corresponding column of the table as '\*', '\*\*' and '\*\*\*' respectively. Here, the absence of any \* denotes  $p$ -value greater than 0.05, which implies no significant statistical difference for a particular pair-wise comparison.

As we can see, for the first four measurements, the proposed DCIL-IncLPSVM achieves the same performance as the SMOTE, both outperforming SVM and LPSVM in the presence of class imbalance. From Exp1 to Exp3, with the increase of class imbalance, the specificity of LPSVM and SVM grows to 100% and the sensitivity drops significantly. This follows that LPSVM and SVM are vulnerable on the classification of minority class and result in lower  $F$ -measure and  $G$ -mean. In some extreme cases, they may classify all data instances into the majority class (e.g., Exp2 and Exp3 on Car, Exp3 on Abalone, and Exp3 on Cover Type). In contrast, the proposed DCIL-IncLPSVM performs consistently stable despite varied degrees of class imbalance. For training efficiency, the training time for DCIL-IncLPSVM is seen normally slightly greater than that of the LPSVM, but apparently both are at the same level, and greater than that of the SVM. This is also consistent with the theoretical complexity comparison that we conduct in Table 5.

The interesting similarity between DCIL-IncLPSVM and SMOTE can be explained as follows. The nature of SMOTE for class-imbalance problem-solving is over-sampling. SMOTE expands the minority class to the size of the majority class under the condition that the expanded minority class data follows the same distribution of the original minority class. As a result, this increases the total training error from minority class by  $n_{\text{maj}}/n_{\text{min}}$  times when those synthesized samples by SMOTE are added in the LPSVM training. The class-balancing effectiveness of SMOTE equals to that of the proposed DCIL-IncLPSVM  $\|\xi_+\|^2$  weight, (26). However SMOTE for oversampling needs often extra time to synthesize samples, its training time and memory usage are both higher than proposed DCIL-IncLPSVM (e.g., Exp3 on Cover Type), especially when a large number of synthesis samples are required for class-balancing.

**Table 3**  
Results of static class imbalance robustness tests (I).

Dataset	Exp.	Algorithm	Sensitivity	Specificity	F-measure	G-mean	Training time
Breast wisconsin	1	LPSVM	91.13 ± 5.50	97.75 ± 2.08	94.17 ± 3.22	<b>94.33</b> ± 3.05	<b>0.0004</b> ± 0.0001
		SVM	93.25 ± 6.23	95.63 ± 6.84	<b>94.37</b> ± 4.60	94.31 ± 4.95	0.0009 ± 0.0027
		SMOTE	91.13 ± 5.50	97.75 ± 2.08	94.17 ± 3.22	<b>94.33</b> ± 3.05	0.0007 ± 0.0001
		DCIL	91.13 ± 5.50	97.75 ± 2.08	94.17 ± 3.22	<b>94.33</b> ± 3.05	0.0005 ± 0.0001
		wLPSVM	91.13 ± 5.50	97.75 ± 2.08	94.17 ± 3.22	<b>94.33</b> ± 3.05	0.0005 ± 0.0001
		SVM-w	93.25 ± 6.23	95.63 ± 6.84	<b>94.37</b> ± 4.60	94.31 ± 4.95	0.0009 ± 0.0026
	SVM-ensemble	93.25 ± 6.23	95.63 ± 6.84	<b>94.37</b> ± 4.60	94.31 ± 4.95	0.0013 ± 0.0020	
	2	LPSVM	86.00 ± 6.29	98.38 ± 1.63	91.55 ± 3.83**	91.91 ± 3.52**	<b>0.0004</b> ± 0.0001
		SVM	81.13 ± 8.08	98.50 ± 1.84	88.62 ± 5.23***	89.27 ± 4.63***	0.0016 ± 0.0006
		SMOTE	90.25 ± 4.47	98.13 ± 1.56	93.90 ± 2.57	94.07 ± 2.44	0.0122 ± 0.0003
		DCIL	90.25 ± 4.47	98.13 ± 1.56	93.90 ± 2.57	94.07 ± 2.44	0.0005 ± 0.0001
		wLPSVM	90.25 ± 4.47	98.13 ± 1.56	93.90 ± 2.57	94.07 ± 2.44	0.0005 ± 0.0001
		SVM-w	92.75 ± 4.47	97.50 ± 1.94	<b>94.96</b> ± 2.70	<b>95.06</b> ± 2.59	0.0028 ± 0.0022
	SVM-ensemble	92.38 ± 3.83	97.75 ± 1.75	94.89 ± 2.35	95.00 ± 2.26	0.0033 ± 0.0011	
	3	LPSVM	77.88 ± 9.06	98.50 ± 2.00	86.54 ± 5.40***	87.40 ± 4.78***	<b>0.0003</b> ± 0.0002
		SVM	75.38 ± 11.02	98.75 ± 1.85	84.87 ± 7.69***	86.00 ± 6.57***	0.0103 ± 0.0137
		SMOTE	90.63 ± 5.35	97.25 ± 2.08	<b>93.66</b> ± 3.09	<b>93.83</b> ± 2.93	0.0119 ± 0.0051
		DCIL	90.63 ± 5.35	97.25 ± 2.08	<b>93.66</b> ± 3.09	<b>93.83</b> ± 2.93	<b>0.0003</b> ± 0.0002
wLPSVM		90.63 ± 5.35	97.25 ± 2.08	<b>93.66</b> ± 3.09	<b>93.83</b> ± 2.93	<b>0.0003</b> ± 0.0002	
SVM-w		86.75 ± 12.17	97.00 ± 2.18	90.90 ± 8.12	91.42 ± 6.97	0.0523 ± 0.0405	
SVM-ensemble	86.50 ± 11.82	97.63 ± 2.01	91.08 ± 8.02	91.60 ± 6.89	0.0543 ± 0.0615		
Car	1	LPSVM	99.75 ± 1.09	80.88 ± 8.71	91.29 ± 3.44	89.67 ± 4.71	<b>0.0004</b> ± 0.0001
		SVM	95.88 ± 5.49	89.25 ± 6.13	<b>92.80</b> ± 3.42	<b>92.38</b> ± 3.58*	0.0009 ± 0.0009
		SMOTE	99.75 ± 1.09	80.88 ± 8.71	91.29 ± 3.44	89.67 ± 4.71	0.0006 ± 0.0001
		DCIL	99.75 ± 1.09	80.88 ± 8.71	91.29 ± 3.44	89.67 ± 4.71	<b>0.0004</b> ± 0.0001
		wLPSVM	99.75 ± 1.09	80.88 ± 8.71	91.29 ± 3.44	89.67 ± 4.71	<b>0.0004</b> ± 0.0001
		SVM-w	95.88 ± 5.49	89.25 ± 6.13	<b>92.80</b> ± 3.42	<b>92.38</b> ± 3.58*	0.0008 ± 0.0009
	SVM-ensemble	95.88 ± 5.49	89.25 ± 6.13	<b>92.80</b> ± 3.42	<b>92.38</b> ± 3.58*	0.0012 ± 0.0008	
	2	LPSVM	2.75 ± 3.25	100.00 ± 0.00	NaN***	11.41 ± 12.03***	<b>0.0003</b> ± 0.0002
		SVM	71.13 ± 7.22	98.38 ± 3.19	82.16 ± 4.45***	83.50 ± 3.82***	0.0013 ± 0.0009
		SMOTE	100.00 ± 0.00	85.63 ± 4.32	93.33 ± 1.84	92.50 ± 2.37	0.0099 ± 0.0056
		DCIL	100.00 ± 0.00	85.63 ± 4.32	93.33 ± 1.84	92.50 ± 2.37	0.0004 ± 0.0002
		wLPSVM	100.00 ± 0.00	85.63 ± 4.32	93.33 ± 1.84	92.50 ± 2.37	0.0004 ± 0.0002
		SVM-w	99.63 ± 1.63	86.75 ± 4.62	<b>93.64</b> ± 1.93	<b>92.92</b> ± 2.41	0.0038 ± 0.0022
	SVM-ensemble	90.88 ± 6.99	95.00 ± 3.62	92.68 ± 4.18	92.82 ± 3.88	0.0389 ± 0.0462	
	3	LPSVM	0.00 ± 0.00	100.00 ± 0.00	NaN***	0.00 ± 0.00***	<b>0.0003</b> ± 0.0002
		SVM	34.25 ± 5.48	99.75 ± 0.75	50.68 ± 6.06***	58.26 ± 4.68***	0.0226 ± 0.0079
		SMOTE	100.00 ± 0.00	86.38 ± 4.84	93.67 ± 2.13	92.90 ± 2.60	0.0371 ± 0.0150
		DCIL	100.00 ± 0.00	86.38 ± 4.84	93.67 ± 2.13	92.90 ± 2.60	0.0005 ± 0.0002
wLPSVM		100.00 ± 0.00	86.38 ± 4.84	93.67 ± 2.13	92.90 ± 2.60	0.0005 ± 0.0002	
SVM-w		99.25 ± 2.25	89.50 ± 4.78	<b>94.67</b> ± 2.09*	94.20 ± 2.47*	0.0684 ± 0.0253	
SVM-ensemble	93.25 ± 5.37	96.38 ± 2.90	<b>94.67</b> ± 3.02	<b>94.74</b> ± 2.88*	0.1627 ± 0.0477		
Abalone	1	LPSVM	97.88 ± 1.82	93.38 ± 3.65	<b>95.74</b> ± 1.70	<b>95.57</b> ± 1.83	0.0005 ± 0.0001
		SVM	96.88 ± 3.43	93.38 ± 5.02	95.24 ± 2.78	95.05 ± 2.92	0.0010 ± 0.0032
		SMOTE	97.88 ± 1.82	93.38 ± 3.65	<b>95.74</b> ± 1.70	<b>95.57</b> ± 1.83	0.0007 ± 0.0002
		DCIL	97.88 ± 1.82	93.38 ± 3.65	<b>95.74</b> ± 1.70	<b>95.57</b> ± 1.83	0.0005 ± 0.0001
		wLPSVM	97.88 ± 1.82	93.38 ± 3.65	<b>95.74</b> ± 1.70	<b>95.57</b> ± 1.83	0.0005 ± 0.0001
		SVM-w	96.88 ± 3.43	93.38 ± 5.02	95.24 ± 2.78	95.05 ± 2.92	<b>0.0003</b> ± 0.0000
	SVM-ensemble	96.88 ± 3.43	93.38 ± 5.02	95.24 ± 2.78	95.05 ± 2.92	0.0008 ± 0.0001	
	2	LPSVM	75.00 ± 9.68	98.63 ± 2.01	84.70 ± 6.21***	85.80 ± 5.42***	<b>0.0005</b> ± 0.0002
		SVM	86.25 ± 7.35	98.00 ± 2.45	91.48 ± 4.08***	91.83 ± 3.71***	0.0010 ± 0.0003
		SMOTE	97.75 ± 1.56	94.00 ± 4.06	<b>95.98</b> ± 2.03	<b>95.83</b> ± 2.16	0.0143 ± 0.0023
		DCIL	97.75 ± 1.56	94.00 ± 4.06	<b>95.98</b> ± 2.03	<b>95.83</b> ± 2.16	<b>0.0005</b> ± 0.0001
		wLPSVM	97.75 ± 1.56	94.00 ± 4.06	<b>95.98</b> ± 2.03	<b>95.83</b> ± 2.16	<b>0.0005</b> ± 0.0001
		SVM-w	97.38 ± 2.43	91.75 ± 5.65	94.77 ± 2.60*	94.46 ± 2.86*	0.0030 ± 0.0022
	SVM-ensemble	97.13 ± 2.53	94.00 ± 3.25	95.95 ± 1.86	95.50 ± 1.90	0.0035 ± 0.0011	
	3	LPSVM	0.00 ± 0.00	100.00 ± 0.00	NaN***	0.00 ± 0.00***	<b>0.0007</b> ± 0.0003
		SVM	0.00 ± 0.00	100.00 ± 0.00	NaN***	0.00 ± 0.00***	0.0076 ± 0.0025
		SMOTE	97.00 ± 1.70	93.75 ± 3.31	<b>95.47</b> ± 1.69	<b>95.34</b> ± 1.79	0.0807 ± 0.0239
		DCIL	97.00 ± 1.70	93.75 ± 3.31	<b>95.47</b> ± 1.69	<b>95.34</b> ± 1.79	0.0009 ± 0.0004
wLPSVM		97.00 ± 1.70	93.75 ± 3.31	<b>95.47</b> ± 1.69	<b>95.34</b> ± 1.79	0.0009 ± 0.0004	
SVM-w		95.88 ± 4.20	94.50 ± 3.32	95.20 ± 1.70	95.12 ± 1.65	0.2702 ± 0.1879	
SVM-ensemble	95.88 ± 3.19	94.75 ± 3.17	95.12 ± 1.55	95.17 ± 1.57	0.2824 ± 0.0741		

#### 5.4. Dynamic class imbalance robustness tests on face membership authentication

In context of incremental learning, class imbalance may occur in two cases: (1) the class distribution of the whole dataset is imbalanced, and the class distribution of data subsets is also imbalanced;

and (2) the class distribution of the whole dataset is balanced, but subsets are class imbalanced.

To evaluate the algorithm robustness to dynamic class imbalance, we study the face membership authentication (FMA) problem (Pang et al., 2003). The membership authentication problem is to distinguish the membership class from the non-membership

**Table 4**  
Results of static class imbalance robustness tests (II).

Dataset	Exp.	Algorithm	Sensitivity	Specificity	F-measure	G-mean	Training time
Internet advertisements	1	LPSVM	87.92 ± 2.29	96.96 ± 1.24	<b>92.07</b> ± 1.22	<b>92.32</b> ± 1.14	<b>1.0001</b> ± 0.0505
		SVM	89.76 ± 2.42	93.64 ± 1.86	91.53 ± 1.56	91.67 ± 1.50	1.8807 ± 0.9405
		SMOTE	87.92 ± 2.29	96.96 ± 1.24	<b>92.07</b> ± 1.22	<b>92.32</b> ± 1.14	1.0141 ± 0.0235
		DCIL	87.92 ± 2.29	96.96 ± 1.24	<b>92.07</b> ± 1.22	<b>92.32</b> ± 1.14	1.1442 ± 0.0206
		wLPSVM	87.92 ± 2.29	96.96 ± 1.24	<b>92.07</b> ± 1.22	<b>92.32</b> ± 1.14	1.1442 ± 0.0206
		SVM-w	89.76 ± 2.42	93.64 ± 1.86	91.53 ± 1.56	91.67 ± 1.50	1.9199 ± 0.9359
	SVM-ensemble	89.76 ± 2.42	93.64 ± 1.86	91.53 ± 1.56	91.67 ± 1.50	2.0535 ± 1.0351	
	2	LPSVM	79.22 ± 2.58	99.38 ± 0.30	88.08 ± 1.62***	88.72 ± 1.45***	<b>0.9744</b> ± 0.0438
		SVM	85.02 ± 2.10	97.88 ± 1.07	90.85 ± 1.21	91.21 ± 1.12	5.5934 ± 2.0941
		SMOTE	85.84 ± 2.97	98.04 ± 0.81	91.39 ± 1.70	91.72 ± 1.56	1.2726 ± 0.0318
		DCIL	85.84 ± 2.97	98.04 ± 0.81	91.39 ± 1.70	91.72 ± 1.56	1.1383 ± 0.0170
		wLPSVM	85.84 ± 2.97	98.04 ± 0.81	91.39 ± 1.70	91.72 ± 1.56	1.1383 ± 0.0170
		SVM-w	89.76 ± 3.69	95.84 ± 1.82	<b>92.54</b> ± 1.79***	<b>92.72</b> ± 1.64***	3.4988 ± 0.9967
	SVM-ensemble	87.76 ± 3.52	97.82 ± 0.93	92.37 ± 2.11**	92.63 ± 1.95**	3.2514 ± 1.3482	
	3	LPSVM	76.02 ± 2.42	99.76 ± 0.29	86.24 ± 1.54***	87.07 ± 1.36***	<b>1.1241</b> ± 0.0650
		SVM	77.28 ± 3.10	99.72 ± 0.38	87.01 ± 1.93***	87.77 ± 1.72***	14.5757 ± 5.6493
		SMOTE	85.10 ± 2.22	98.60 ± 0.86	91.25 ± 1.32	91.59 ± 1.23	2.2402 ± 0.0824
		DCIL	85.10 ± 2.22	98.60 ± 0.86	91.25 ± 1.32	91.59 ± 1.23	1.4347 ± 0.0443
wLPSVM		85.10 ± 2.22	98.60 ± 0.86	91.25 ± 1.32	91.59 ± 1.23	1.4347 ± 0.0443	
SVM-w		88.78 ± 2.23	96.74 ± 1.25	<b>92.45</b> ± 1.18**	<b>92.66</b> ± 1.11**	19.4058 ± 7.5932	
SVM-ensemble	86.24 ± 1.92	98.38 ± 0.75	91.80 ± 1.13*	92.10 ± 1.054*	4.1036 ± 0.6133		
Cover type	1	LPSVM	90.82 ± 1.41	79.02 ± 0.77	85.75 ± 0.58	84.71 ± 0.47	0.0041 ± 0.0002
		SVM	93.01 ± 1.40	78.94 ± 0.97	<b>86.89</b> ± 0.51***	<b>85.68</b> ± 0.42***	0.5366 ± 0.0501
		SMOTE	90.82 ± 1.41	79.02 ± 0.77	85.75 ± 0.58	84.71 ± 0.47	0.0037 ± 0.0002
		DCIL	90.82 ± 1.41	79.02 ± 0.77	85.75 ± 0.58	84.71 ± 0.47	<b>0.0032</b> ± 0.0002
		wLPSVM	90.82 ± 1.41	79.02 ± 0.77	85.75 ± 0.58	84.71 ± 0.47	<b>0.0032</b> ± 0.0002
		SVM-w	93.01 ± 1.40	78.94 ± 0.97	<b>86.89</b> ± 0.51***	<b>85.68</b> ± 0.424***	0.5535 ± 0.0874
	SVM-ensemble	93.01 ± 1.40	78.95 ± 0.96	<b>86.90</b> ± 0.514***	<b>85.68</b> ± 0.42***	0.5722 ± 0.0434	
	2	LPSVM	15.35 ± 3.26	99.20 ± 0.22	26.28 ± 5.02***	38.75 ± 4.53***	<b>0.0137</b> ± 0.0010
		SVM	6.40 ± 0.48	99.65 ± 0.13	11.99 ± 0.82***	25.25 ± 0.89***	1.9643 ± 0.2200
		SMOTE	90.97 ± 1.06	79.12 ± 0.61	<b>85.88</b> ± 0.62	<b>84.84</b> ± 0.59	4.0198 ± 0.0664
		DCIL	90.97 ± 1.06	79.12 ± 0.61	<b>85.88</b> ± 0.62	<b>84.84</b> ± 0.59	0.0203 ± 0.0011
		wLPSVM	90.97 ± 1.06	79.12 ± 0.61	<b>85.88</b> ± 0.62	<b>84.84</b> ± 0.59	0.0203 ± 0.0011
		SVM-w	99.67 ± 0.33	64.40 ± 1.22	84.73 ± 0.36***	80.11 ± 0.68***	10.5458 ± 0.3992
	SVM-ensemble	91.65 ± 1.28	77.52 ± 0.82	85.13 ± 0.61**	84.24 ± 0.56**	11.3787 ± 0.3721	
	3	LPSVM	0.00 ± 0.00	100.00 ± 0.00	NaN***	0.00 ± 0.00***	<b>0.1466</b> ± 0.0330
		SVM	0.00 ± 0.00	100.00 ± 0.00	NaN***	0.00 ± 0.00***	14.5101 ± 1.5504
		SMOTE	90.44 ± 0.84	79.40 ± 0.66	85.70 ± 0.44	84.73 ± 0.43	21.8657 ± 2.1887
		DCIL	90.44 ± 0.84	79.40 ± 0.66	85.70 ± 0.44	84.73 ± 0.43	0.2795 ± 0.0386
wLPSVM		90.44 ± 0.84	79.40 ± 0.66	85.70 ± 0.44	84.73 ± 0.43	0.2795 ± 0.0386	
SVM-w		99.90 ± 0.10	56.88 ± 1.14	82.21 ± 0.37***	75.37 ± 0.74***	1323.8602 ± 85.2465	
SVM-ensemble	93.10 ± 0.92	79.07 ± 0.54	<b>87.00</b> ± 0.48***	<b>85.80</b> ± 0.45***	2238.2005 ± 215.5607		
Cardiotocography	1	LPSVM	89.95 ± 4.20	91.35 ± 2.97	90.55 ± 2.24	90.60 ± 2.10	<b>0.0005</b> ± 0.0001
		SVM	90.65 ± 3.84	91.00 ± 3.95	<b>90.80</b> ± 2.16	<b>90.77</b> ± 2.15	0.0018 ± 0.0002
		SMOTE	89.95 ± 4.20	91.35 ± 2.97	90.55 ± 2.24	90.60 ± 2.10	0.0008 ± 0.0001
		DCIL	89.95 ± 4.20	91.35 ± 2.97	90.55 ± 2.24	90.60 ± 2.10	0.0009 ± 0.0017
		wLPSVM	89.95 ± 4.20	91.35 ± 2.97	90.55 ± 2.24	90.60 ± 2.10	0.0009 ± 0.0017
		SVM-w	90.65 ± 3.84	91.00 ± 3.95	<b>90.80</b> ± 2.16	<b>90.77</b> ± 2.15	0.0018 ± 0.0002
	SVM-ensemble	90.65 ± 3.84	91.00 ± 3.95	<b>90.80</b> ± 2.16	<b>90.77</b> ± 2.15	0.0023 ± 0.0002	
	2	LPSVM	60.10 ± 7.71	99.35 ± 0.79	74.49 ± 5.77***	77.11 ± 4.81***	<b>0.0004</b> ± 0.0002
		SVM	74.85 ± 5.73	98.05 ± 1.43	84.56 ± 3.94***	85.60 ± 3.43***	0.0096 ± 0.0051
		SMOTE	89.30 ± 3.33	92.30 ± 1.98	<b>90.64</b> ± 1.87	<b>90.76</b> ± 1.74	0.0232 ± 0.0104
		DCIL	89.30 ± 3.33	92.30 ± 1.98	<b>90.64</b> ± 1.87	<b>90.76</b> ± 1.74	0.0005 ± 0.0002
		wLPSVM	89.30 ± 3.33	92.30 ± 1.98	<b>90.64</b> ± 1.87	<b>90.76</b> ± 1.74	0.0005 ± 0.0002
		SVM-w	91.00 ± 5.13	89.85 ± 3.34	90.43 ± 2.03	90.34 ± 1.79	0.0199 ± 0.0098
	SVM-ensemble	86.95 ± 5.34	93.75 ± 2.32	89.94 ± 2.79	90.22 ± 2.53	0.0219 ± 0.0098	
	3	LPSVM	28.70 ± 4.36	99.85 ± 0.36	44.37 ± 5.314***	53.37 ± 4.13***	<b>0.0007</b> ± 0.0003
		SVM	55.85 ± 7.37	99.70 ± 0.56	71.25 ± 6.00***	74.45 ± 4.88***	0.0316 ± 0.0093
		SMOTE	89.30 ± 3.65	93.30 ± 1.73	<b>91.09</b> ± 2.13	<b>91.25</b> ± 2.00	0.0472 ± 0.0129
		DCIL	89.30 ± 3.65	93.30 ± 1.73	<b>91.09</b> ± 2.13	<b>91.25</b> ± 2.00	0.0011 ± 0.0004
wLPSVM		89.30 ± 3.65	93.30 ± 1.73	<b>91.09</b> ± 2.13	<b>91.25</b> ± 2.00	0.0011 ± 0.0004	
SVM-w		92.90 ± 3.27	89.10 ± 2.34	91.07 ± 1.87	91.11 ± 1.81	0.1494 ± 0.0547	
SVM-ensemble	87.65 ± 2.95	94.80 ± 1.91	90.88 ± 1.78	91.14 ± 1.68	0.1665 ± 0.0164		

class based on the personal facial images stored in a database. The FMA problem presents here a comprehensive dynamic class imbalance case, because the membership group is generally much smaller than the non-membership group, and the size of membership is adjustable. We use the Mpeg-7 dataset (Kim, Kim, & Lee, 2003) that includes face images collected from 271 individuals (5 images for each). Each image has the size of  $23 \times 28$  pixels. In the

experiment, we use directly pixel values as the features of each face image.

For online FMA experiment, we select in turn 4 of 5 images for training and the left one for testing, and conduct a 5-fold cross validation. For each round cross-validation, we set up our experiments the same as before by dividing training data equally into 10 subsets and training each learner by 10 stages incremental learning. At

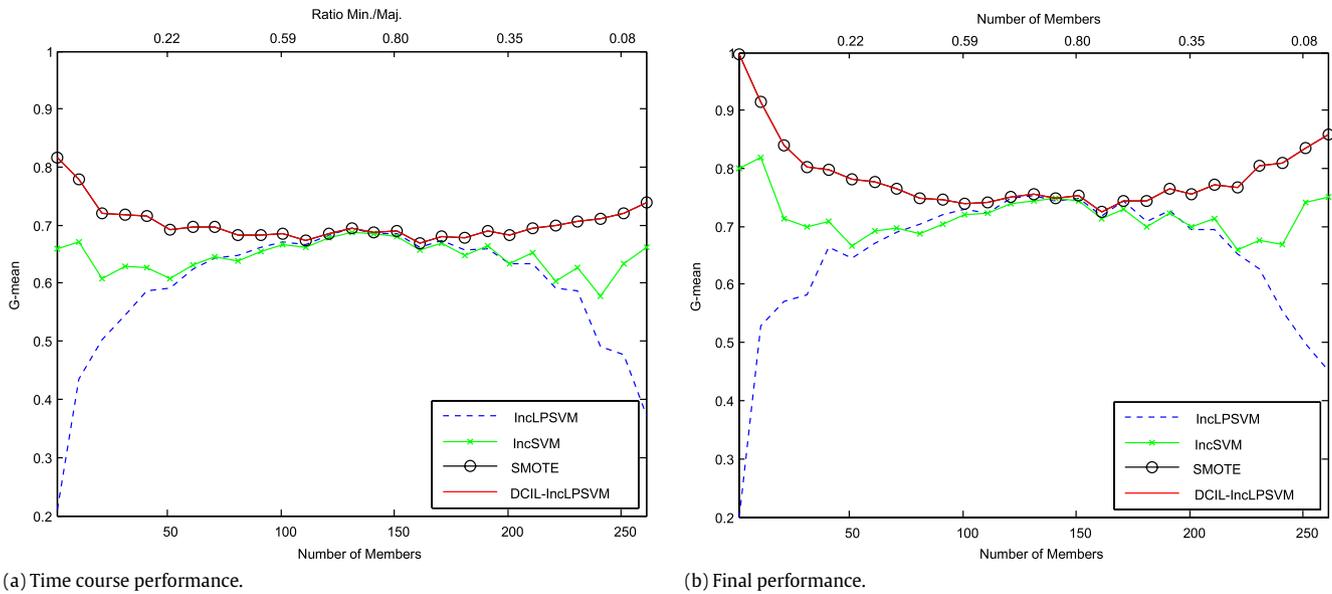


Fig. 2. Robustness tests on online face membership authentication under the condition that the whole dataset has diverse class imbalances.

Table 5

Theoretical complexity comparison, where  $n$  denotes the number of training samples,  $d$  the number of attributes, and  $n_{\text{new}}$  the number of synthesized samples.

Algorithm	Time complexity	Space complexity
LPSVM	Less than $O(nd^2 + d^3)$	$O(d^2)$
SVM	Between $O(n)$ and $O(n^3)$ (Platt, 1999)	$O(n^2)$ (Tsang, Kwok, Cheung, & Cristianini, 2005)
SMOTE	Around $O(4n_{\text{new}}d)$	$O(n_{\text{new}}d)$
DCIL	Less than $O(nd^2 + d^3)$	$O(d^2)$
SVM-w	Between $O(n)$ and $O(n^3)$	$O(n^2)$
SVM-ensemble	Between $O(2n)$ and $O(n^3)$	$O(n^2)$

each incremental learning stage, one subset is used for incremental learning, and the learning effectiveness in terms of  $G$ -mean (i.e., (49)) is evaluated on the testing data. Fig. 2 presents the comparison results of four incremental learning algorithms: Incremental LPSVM (IncLPSVM), Incremental SVM (IncSVM), SMOTE, and the proposed DCIL-IncLPSVM. Here, the performance of each algorithm is measured by the average  $G$ -mean value of 5-fold cross validation. The final incremental stage  $G$ -mean is recorded as the ‘final performance’, and the average  $G$ -mean of all 10 stages incremental learning as the ‘time course performance’.

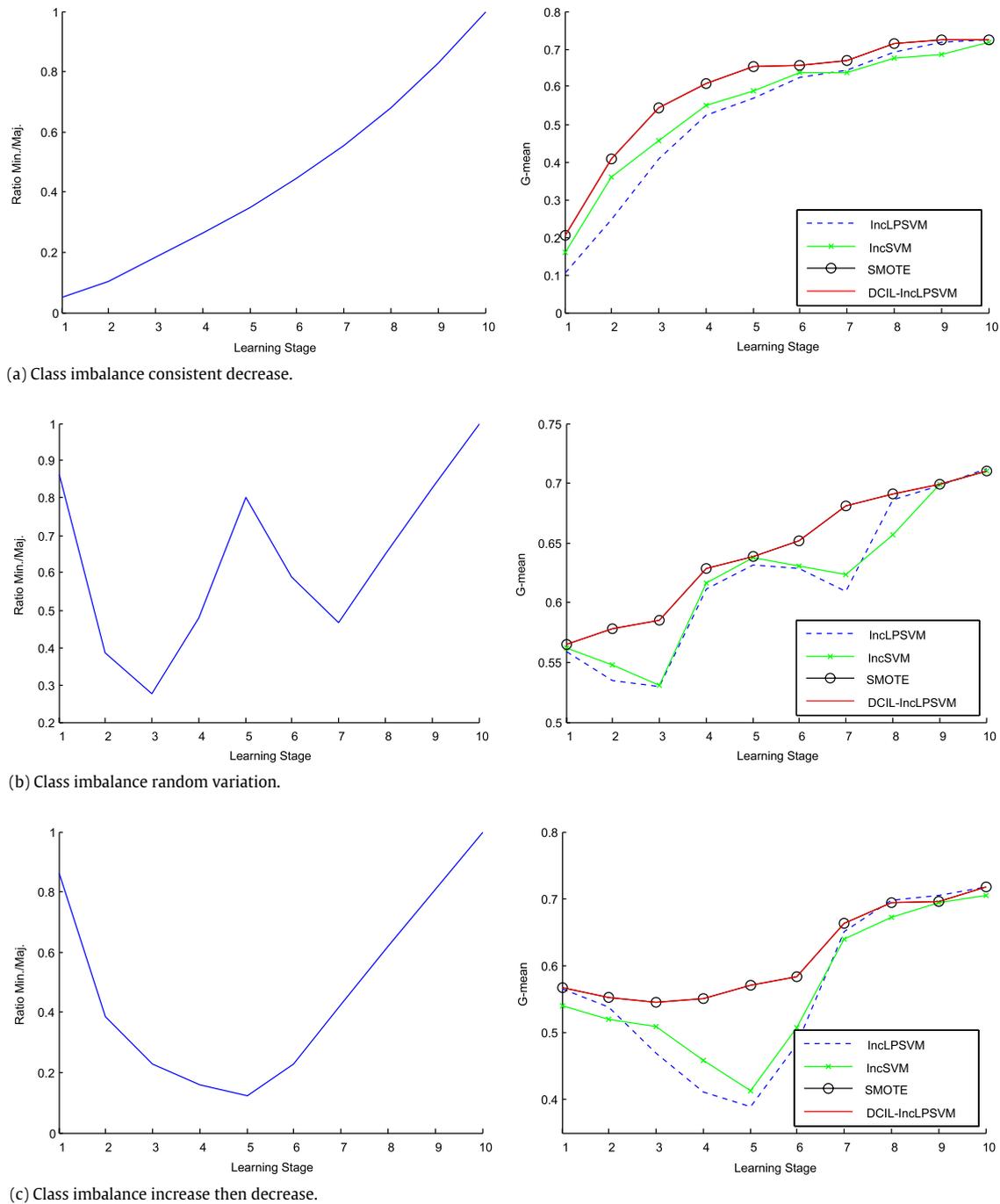
As shown in Fig. 2, we first compare the performance of each classifier under the condition that the whole dataset has diverse class imbalances. In doing so, the membership group size is set to range from 1 to 270 person with a step size of 10 persons. This makes the class imbalance ratio of the whole dataset increase first from 1/270 to 135/136 in which the membership group is the minority class, and then decrease from 135/136 back to 1/270 where the non-membership group becomes the minority class. The obtained results are shown in Fig. 2(a) and (b) respectively. As we can see for both comparisons, all algorithms perform similarly if the membership sizes are between 100 and 150, and the whole dataset is almost class balanced (i.e., the ratio is close to 1). When the data has noticeable class imbalance, the proposed DCIL-IncLPSVM and SMOTE significantly outperform IncLPSVM and IncSVM. In the case of extreme class imbalance when membership size is below 30 or over 240 (i.e., the ratio is close to  $1/270 = 0.003$ ), IncLPSVM  $G$ -mean drops down to under 0.5, but the proposed DCIL-IncLPSVM

and SMOTE maintains the same superiority to IncSVM with its  $G$ -mean growing to above 0.8.

The second robustness test is under the condition that the whole dataset is class balanced, and the subsets at different stages of incremental learning are imbalanced. We are particularly interested in assessing the algorithms’ responses to varied class imbalance caused by data additions. In the experiment, we use the overall balanced dataset with 135 members and 136 non-members. For online FMA, we change the level of class imbalance according to schemes ‘consistent decrease’, ‘random variation’ and ‘increase then decrease’ respectively. Fig. 3 gives the experimental results, where left figures plot the real time variation of class imbalance, and right ones summarize the obtained  $G$ -means. As we can see, for all three experiments, the performance of the proposed DCIL-IncLPSVM, similar to SMOTE, increases consistently through the whole learning lifetime despite of the diverse class imbalance variations. However, for the IncSVM and IncLPSVM, the rising of class imbalance has caused apparent performance degradation. As seen from stages 1–3 and 5–7 in Fig. 3(b) and stages 1–5 in Fig. 3(c), the performance of both IncSVM and IncLPSVM drops significantly. However, at the final stage when the class distribution is back to balance, all algorithms give very similar performance. This indicates that for online incremental learning, the effect of class imbalance may overwhelm the contribution of newly added data.

### 5.5. Class imbalance on multi-class face recognition

For multi-class classification, traditional multi-class (e.g.,  $k$ -class,  $k > 2$ ) IncLPSVM (McIncLPSVM) is a combination of  $k(k - 1)$  binary IncLPSVMs, in which class imbalance is incurred for each individual binary IncLPSVM training. The overall accuracy of multi-class classification is affected by the performance of each binary IncLPSVM. The same class-imbalance influence also adheres to typical multi-class incremental SVM (McIncSVM) (Cauwenberghs & Poggio, 2000). The proposed multi-class DCIL-IncLPSVM (McDCIL-IncLPSVM) suppresses such impacts by performing a real-time DCIL on each binary IncLPSVM. Therefore we can safely assume that the proposed algorithm will achieve bigger improvement as compared to the McIncLPSVM and McIncSVM, especially when the number of classes for classification is large.



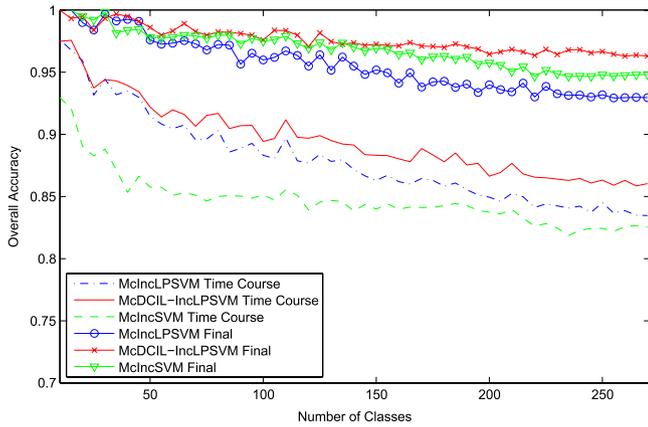
**Fig. 3.** Robustness tests on online face membership authentication under the condition that the whole dataset is class balanced and subsets for stage incremental learning are imbalanced.

To testify this assumption, we initially compare the above three multi-class SVM algorithms for the classification of six benchmark UCI (Frank & Asuncion, 2010) multi-class datasets. Table 6 gives the dataset description and the obtained testing results. As we can see for six datasets, the proposed McDCIL-IncLPSVM has all significantly higher accuracy than that of the McInclPSVM at a cost of some reasonable extra training time. As compared to the McInclSVM, the McDCIL-IncLPSVM outperforms the McInclSVM on four datasets, in which *Leaf-Margin* and *Leaf-Texture* both have as big as 100 classes; *Glass* and *Yeast* have less than 10 classes, but their class distributions are found highly imbalanced. Next, we further study the face recognition (FR) task using the Mpeg-7 face

dataset introduced above. Face recognition is a multi-class classification problem in which each person represents one class, and the objective is to correctly classify unlabeled facial images. The same as above, we use pixel values as the features of each face image and a 5-fold cross validation approach for the experiment. We compare two multi-class classifiers for online face recognition with different numbers of classes. In doing so, we set the online FR system working on 10–270 random selected persons with 5 persons as the interval. Fig. 4 presents the final and time course average accuracy of the two algorithms. As we can see, all accuracies decrease with the increase of the class number, which indicates that larger number of classes increases the classification difficulty for both algorithms.

**Table 6**  
Results of multi-class classification tests.

Dataset	#Class	#Var.	#Train	#Test	Algorithm	Accuracy	Training time
Glass	6	9	190	24	McLPSVM	41.04 ± 4.93***	0.0006 ± 0.0003
					McDCIL-IncLPSVM	66.46 ± 9.31	0.0016 ± 0.0014
					McSVM	57.71 ± 8.69***	0.0054 ± 0.0011
Yeast	9	8	1389	90	McLPSVM	36.04 ± 1.91***	0.0013 ± 0.0006
					McDCIL-IncLPSVM	52.11 ± 1.88	0.0021 ± 0.0010
					McSVM	50.63 ± 1.94***	0.1034 ± 0.0466
CANE-9	9	856	900	180	McLPSVM	91.03 ± 2.02**	0.1623 ± 0.0092
					McDCIL-IncLPSVM	93.06 ± 1.79	1.2771 ± 0.0904
					McSVM	90.58 ± 2.82 *	0.0573 ± 0.0058
USPS	10	256	7298	2000	McLPSVM	88.39 ± 0.56***	0.1225 ± 0.0234
					McDCIL-IncLPSVM	92.89 ± 0.46	0.2641 ± 0.0583
					McSVM	94.09 ± 0.48***	5.3489 ± 0.1363
Leaf-margin	100	64	1000	600	McLPSVM	54.89 ± 1.20***	0.0121 ± 0.0029
					McDCIL-IncLPSVM	76.53 ± 1.63	0.0853 ± 0.0235
					McSVM	75.40 ± 1.32**	0.7897 ± 0.1597
Leaf-texture	100	64	1000	600	McLPSVM	58.12 ± 1.48***	0.0088 ± 0.0022
					McDCIL-IncLPSVM	63.78 ± 1.63	0.0588 ± 0.0165
					McSVM	63.59 ± 1.60	0.7194 ± 0.2875

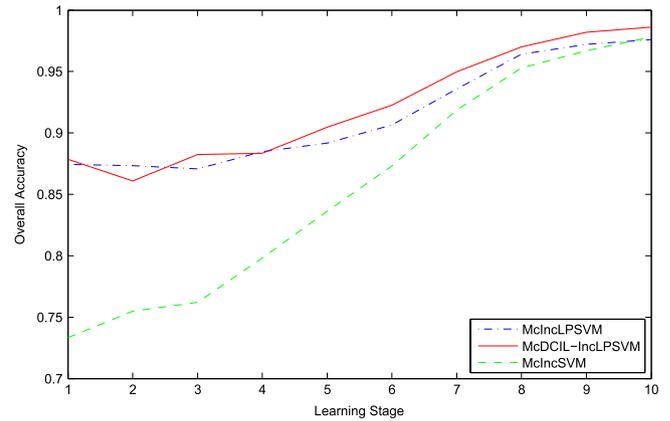


**Fig. 4.** Multi-class classification tests on online face recognition in different numbers of classes.

The superiority of the proposed DCIL-IncLPSVM in terms of its final accuracy emerges when the class number is around 30, and it grows steadily with the increase of the class number. When the number of classes rises to 270, such superiority rises to 3.5%. Although the superiority on time course average accuracy is not as much as that on the final accuracy, it is apparent that the consistent increase trend is the same for both evaluations. This indicates that the proposed DCIL-IncLPSVM performs better online face recognition that involves more than 30 persons, and consistently outperforms traditional IncLPSVM throughout the whole incremental learning process. Fig. 5 gives an example of online face recognition for 50 persons, where the overall accuracy of McIncLPSVM, McIncSVM, and McDCIL-IncLPSVM are plotted at 10 incremental stages.

## 6. Conclusions and future work

Aimed at solving the class imbalance problem of LPSVM data stream learning, we successfully developed in this paper a DCIL solution that enables learning incrementally LPSVM and its weights in real time according to the class imbalance ratio of the training data. Our analysis guarantees that the updated LPSVM obtained from proposed DCIL-IncLPSVM (i.e., the model after



**Fig. 5.** Multi-class classification tests on online face recognition in 50 classes.

incremental learning on newly added/retired data) is equivalent to a batch LPSVM model over the updated dataset (i.e., the dataset after data addition/retirement).

For LPSVM batch learning, wLPSVM (Fung & Mangasarian, 2005) models successfully the class imbalance trade-off by a weighting matrix  $\mathbf{N}$ . For online learning in the difficulty of dynamic class imbalance, a straightforward solution is to update  $\mathbf{N}$  and LPSVM for incremental learning. However it is problematic to update  $\mathbf{N}$ , because it involves the updating of two matrix multiplications. For the DCIL of LPSVM, we derive, without loss of information, a new expression of wLPSVM. In this expression, matrix multiplication  $\mathbf{E}^T \mathbf{N} \mathbf{E}$  is replaced with  $\sigma_+ \mathbf{M}_+ + \sigma_- \mathbf{M}_-$  and  $\mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e}$  with  $\sigma_+ \mathbf{v}_+ - \sigma_- \mathbf{v}_-$ . As a result, the weighting matrix  $\mathbf{N} \in \mathbb{R}^{n \times n}$  is transformed to two  $\mathbb{R}^1$  coefficients  $\sigma_+$  and  $\sigma_-$ , which can be easily updated upon any data addition/retirement. For both static and dynamic class imbalance learning, the developed DCIL-IncLPSVM demonstrated high robustness that we expect in practice.

For class imbalance learning, wLPSVM measures the bias of class distribution just by the number of samples, but does not consider the state of sample distribution. However it is noticed that for LPSVM or SVM training, even if the number of samples is even for both classes, bias on decision boundary may still happen due to biased data scattering. Hence, taking sample distribution into

account to weight LPSVM is an interesting direction for our future research.

## Appendix

### A.1. Lemma 1 proof

$$\begin{aligned}
 \mathbf{E}^T \mathbf{N} \mathbf{E} &= \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix}^T \begin{bmatrix} \mathbf{N}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_b \end{bmatrix} \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{E}_a^T \mathbf{E}_b^T \\ \mathbf{E}_b^T \end{bmatrix} \begin{bmatrix} \mathbf{N}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_b \end{bmatrix} \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix} \\
 &= \mathbf{E}_a^T \mathbf{N}_a \mathbf{E}_a + \mathbf{E}_b^T \mathbf{N}_b \mathbf{E}_b \\
 \mathbf{E}^T \mathbf{D} \mathbf{N} \mathbf{e} &= \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix}^T \begin{bmatrix} \mathbf{D}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_b \end{bmatrix} \begin{bmatrix} \mathbf{N}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_b \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{E}_a^T & \mathbf{E}_b^T \end{bmatrix} \begin{bmatrix} \mathbf{D}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_b \end{bmatrix} \begin{bmatrix} \mathbf{N}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_b \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix} \\
 &= \mathbf{E}_a^T \mathbf{D}_a \mathbf{N}_a \mathbf{e} + \mathbf{E}_b^T \mathbf{D}_b \mathbf{N}_b \mathbf{e} \\
 \mathbf{E}^T \mathbf{E} &= \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix}^T \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix} = \begin{bmatrix} \mathbf{E}_a^T & \mathbf{E}_b^T \end{bmatrix} \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix} = \mathbf{E}_a^T \mathbf{E}_a + \mathbf{E}_b^T \mathbf{E}_b \\
 \mathbf{E}^T \mathbf{e} &= \begin{bmatrix} \mathbf{E}_a \\ \mathbf{E}_b \end{bmatrix}^T \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{E}_a^T & \mathbf{E}_b^T \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix} = \mathbf{E}_a^T \mathbf{e} + \mathbf{E}_b^T \mathbf{e}.
 \end{aligned} \tag{50}$$

### A.2. Lemma 2 proof

Let  $\mathbf{E}_i = [\mathbf{x}_i^T \quad -1]$ ,  $\mathbf{N}_i = \begin{cases} \sigma_+ & \text{if } y_i = +1 \\ \sigma_- & \text{if } y_i = -1 \end{cases}$  and  $\mathbf{D}_i = y_i$  be the terms corresponding to the  $i$ -th sample, apply Lemma 1 repeatedly, then

$$\begin{aligned}
 \mathbf{E}^{\eta T} \mathbf{N}^{\eta} \mathbf{E}^{\eta} &= \sum_{i=1}^n \mathbf{E}_i^T \mathbf{N}_i \mathbf{E}_i = \mathbf{E}^{\eta T} \mathbf{N}^{\eta} \mathbf{E}^{\eta} \\
 \mathbf{E}^{\eta T} \mathbf{D}^{\eta} \mathbf{N}^{\eta} \mathbf{e} &= \sum_{i=1}^n \mathbf{E}_i^T \mathbf{D}_i \mathbf{N}_i = \mathbf{E}^{\eta T} \mathbf{D}^{\eta} \mathbf{N}^{\eta} \mathbf{e} \\
 \mathbf{E}^{\eta T} \mathbf{E}^{\eta} &= \sum_{i=1}^n \mathbf{E}_i^T \mathbf{E}_i = \mathbf{E}^{\eta T} \mathbf{E}^{\eta} \\
 \mathbf{E}^{\eta T} \mathbf{e} &= \sum_{i=1}^n \mathbf{E}_i^T \mathbf{e} = \mathbf{E}^{\eta T} \mathbf{e}.
 \end{aligned} \tag{51}$$

## References

- Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European conference on machine learning*, ECML (pp. 39–50).
- Batuwita, R., & Palade, V. (2010). FSVM-CIL: fuzzy support vector machines for class imbalance learning. *IEEE Transactions on Fuzzy Systems*, 18(3), 558–571.
- Cauwenberghs, G., & Poggio, T. (2000). Incremental and decremental support vector machine learning. In *Advances in neural information processing systems NIPS 2000* (pp. 409–415).
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27:1–27:27. [Online]. Available: <http://doi.acm.org/10.1145/1961189.1961199>.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- Chen, S., & He, H. (2009). Sera: selectively recursive approach towards nonstationary imbalanced stream data mining. In *Proceedings of 2009 international joint conference on neural networks* (pp. 522–529).

- Ditzler, G., Muhlbaier, M. D., & Polikar, R. (2010). Incremental learning of new classes in unbalanced datasets: Learn.UDNC. In *Multiple classifier systems* (pp. 33–42).
- Ditzler, G., & Polikar, R. (2010). An ensemble based incremental learning framework for concept drift and class imbalance. In *IJCNN'10* (pp. 1–8).
- Dong, Y.-S., & Han, K.-S. (2005). Text classification based on data partitioning and parameter varying ensembles. In *Proceedings of the 2005 ACM symposium on applied computing, SAC'05*. (pp. 1044–1048). New York, NY, USA: ACM.
- Frank, A., & Asuncion, A. (2010). UCI machine learning repository. [Online]. Available: <http://archive.ics.uci.edu/ml>.
- Fung, G. M., & Mangasarian, O. L. (2001a). Incremental support vector machine classification. In *Proceedings of 7th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 77–86).
- Fung, G., & Mangasarian, O. L. (2001b). Proximal support vector machine classifiers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 77–86). San Francisco, California.
- Fung, G. M., & Mangasarian, O. L. (2005). Multicategory proximal support vector machine classifiers. *Machine Learning*, 59, 77–97.
- Gouripeddi, R., Balasubramanian, V., Panchanathan, S., Harris, J., Bhaskaran, A., & Siegel, R. (2009). Predicting risk of complications following a drug eluting stent procedure: a SVM approach for imbalanced data. In *Proceedings of 22nd IEEE international symposium on computer-based medical systems, CBMS*. (pp. 1–7).
- He, H., & Chen, S. (2008). Imorl: incremental multiple-object recognition and localization. *IEEE Transactions on Neural Networks*, 19(10), 1727–1738.
- He, H., & Garcia, E. (2009). Learning from imbalanced data. *The IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.
- Hong, X., Chen, S., & Harris, C. (2007). A kernel-based two-class classifier for imbalanced data sets. *IEEE Transactions on Neural Networks*, 18(1), 28–41.
- Huang, K., Yang, H., King, I., & Lyu, M. (2006). Imbalanced learning with a biased minimax probability machine. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(4), 913–923.
- Imam, T., Ting, K., & Kamruzzaman, J. (2006). z-SVM: an SVM for improved classification of imbalanced data. In A. Sattar, & B.-H. Kang (Eds.), *Lecture notes in computer science: vol. 4304. AI 2006: advances in artificial intelligence* (pp. 264–273). Berlin, Heidelberg: Springer.
- Karasuyama, M., & Takeuchi, I. (2010). Multiple incremental decremental learning of support vector machines. *IEEE Transactions on Neural Networks*, 21(7), 1048–1059.
- Kim, M.-S., Kim, D., & Lee, S.-Y. (2003). Face recognition using the embedded hmm with second-order block-specific observations. *Pattern Recognition*, 36(11), 2723–2735.
- Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2006). Exploratory under-sampling for class-imbalance learning. In *Proceedings of sixth international conference on data mining, ICDM*. (pp. 965–969). December.
- Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2), 539–550.
- Maciejewski, T., & Stefanowski, J. (2011). Local neighbourhood extension of smote for mining imbalanced data. In *Proceedings of 2011 IEEE symposium on computational intelligence and data mining, CIDM*. (pp. 104–111).
- Muhlbaier, M., Topalis, A., & Polikar, R. (2004). Incremental learning from unbalanced data. In *Neural networks, 2004. Proceedings. 2004 IEEE international joint conference on, Vol. 2* (pp. 1057–1062). July.
- Pang, S., Ban, T., Kadobayashi, Y., & Kasabov, N. (2012). Lda merging and splitting with applications to multiagent cooperative learning and system alteration. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2), 552–564.
- Pang, S., Kim, D., & Bang, S.-Y. (2003). Membership authentication in the dynamic group by face classification using SVM ensemble. *Pattern Recognition Letters*, 24(1–3), 215–225.
- Pang, S., Kim, D., & Bang, S.-Y. (2005). Face membership authentication using SVM classification tree generated by membership-based lle data partition. *IEEE Transactions on Neural Network*, 16(2), 436–446.
- Pang, S., Ozawa, S., & Kasabov, N. (2005). Incremental linear discriminant analysis for classification of data streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(5), 905–914.
- Platt, J. C. (1999). Chapter fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods* (pp. 185–208). Cambridge, MA, USA: MIT Press, [Online]. Available: <http://dl.acm.org/citation.cfm?id=299094.299105>.
- Tao, X., & Ji, H. (2007). A modified PSVM and its application to unbalanced data classification. In *Proceedings of third international conference on natural computation. Vol. 1* (pp. 488–490).
- Tsang, I. W., Kwok, J. T., Cheung, P. ming, & Cristianini, N. (2005). Core vector machines: fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6, 363–392.
- Tveit, A., & Hettl, M. L. (2003). Multicategory incremental proximal support vector classifiers. In *Lecture notes in artificial intelligence (LNAI): vol. 2773. Proceedings of the 7th international conference on knowledge-based information & engineering systems, KES2003*, (pp. 386–392). Springer-Verlag.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin, Germany: Springer-Verlag.
- Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI* (pp. 55–60).
- Wang, H.-Y. (2008). Combination approach of smote and biased-SVM for imbalanced datasets. In *Proceedings of IEEE international joint conference on neural networks* (pp. 228–231).

- Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter – Special issue on learning from imbalanced datasets*, 6, 7–19.
- Wu, G., & Chang, E. (2005). KBA: kernel boundary alignment considering imbalanced data distribution. *The IEEE Transactions on Knowledge and Data Engineering*, 17(6), 786–795.
- Yamasaki, T., & Ikeda, K. (2005). Incremental SVMs and their geometrical analyses. In *Proceedings of international conference on neural networks and brain 05*. Vol. 3 (pp. 1734–1738).
- Yan, R., Liu, Y., Jin, R., & Hauptmann, A. (2003). On predicting rare classes with SVM ensembles in scene classification. In *Proceedings of 2003 IEEE international conference on acoustics, speech, and signal processing (ICASSP'03)*, Vol. 3(4) (pp. 21–29).
- Zhai, Y., Yang, B., Ma, N., & Ruan, D. (2010). New construction of ensemble classifiers for imbalanced datasets. In *Proceedings of 2010 international conference on intelligent systems and knowledge engineering, ISKE*. (pp. 228–233).
- Zhou, Z.-H., & Liu, X.-Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *The IEEE Transactions on Knowledge and Data Engineering*, 18(1), 63–77.
- Zhuang, D., Zhang, B., Yang, Q., Yan, J., Chen, Z., & Chen, Y. (2005). Efficient text classification by weighted proximal SVM. In *Proceedings of fifth IEEE international conference on data mining*.