Tackle Balancing Constraint for Incremental Semi-Supervised Support Vector Learning

Shuyang Yu*
Northeastern University
20164430@stu.neu.edu.cn

Bin Gu*
JD Finance America Corporation
jsgubin@gmail.com

Kunpeng Ning
Nanjing University of Aeronautics and Astronautics
ningkp@nuaa.edu.cn

Haiyan Chen
Nanjing University of Aeronautics and Astronautics
chenhaiyan@nuaa.edu.cn

Jian Pei
Simon Fraser University
JD.com
jpei@cs.sfu.ca

Heng Huang†
University of Pittsburgh
JD Finance America Corporation
heng.huang@pitt.edu

ABSTRACT

Semi-Supervised Support Vector Machine (S^{3}VM) is one of the most popular methods for semi-supervised learning. To avoid the trivial solution of classifying all the unlabeled examples to a same class, balancing constraint is often used with S^{3}VM (denoted as BCS^{3}VM). Recently, a novel incremental learning algorithm (IL-S^{3}VM) based on the path following technique was proposed to significantly scale up S^{3}VM. However, the dynamic relationship of balancing constraint with previous labeled and unlabeled samples impede their incremental method for handling BCS^{3}VM. To fill this gap, in this paper, we propose a new incremental S^{3}VM algorithm (IL-BCS^{3}VM) based on IL-S^{3}VM which can effectively handle the balancing constraint and directly update the solution of BCS^{3}VM. Specifically, to handle the dynamic relationship of balancing constraint with previous labeled and unlabeled samples, we design two unique procedures which can respectively eliminate and add the balancing constraint into S^{3}VM. More importantly, we provide the finite convergence analysis for our IL-BCS^{3}VM algorithm. Experimental results on a variety of benchmark datasets not only confirm the finite convergence of IL-BCS^{3}VM, but also show a huge reduction of computational time compared with existing batch and incremental learning algorithms, while retaining the similar generalization performance.

CCS CONCEPTS

• **Computing methodologies → Machine learning approaches**; **Classification and regression trees**.

KEYWORDS

Semi-Supervised Support Vector Machine; balancing constraint; incremental learning; path following algorithm

ACM Reference Format:
Shuyang Yu, Bin Gu, Kunpeng Ning, Haiyan Chen, Jian Pei, and Heng Huang. 2019. Tackle Balancing Constraint for Incremental Semi-Supervised Support Vector Learning. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330962

1 INTRODUCTION

In many real-world emerging applications, such as image retrieval [20], gene profiling [19] and cancer classification [23], it is usually quite difficult to obtain labeled samples, as the labelling processes are tedious and expensive. Thus labeled samples only account for a small percentage in most datasets, which make it difficult for supervised learning methods to achieve satisfied performance. Therefore semi-supervised support vector machine (S^{3}VM) [13] was proposed as a powerful model to improve the generalization accuracy of SVMs using plentiful unlabeled data. Given the training dataset consisted with a labeled dataset $L = \{(x_1, y_1), \cdots, (x_l, y_l)\}$ and an unlabeled set $U = \{x_{l+1}, \cdots, x_{l+u}\}$, where $x_i \in R^n$, $y_i \in \{+1, -1\}$, $i = 1, \cdots, l$, l is the number of labeled data and u is the number of unlabeled data. Considering the decision function of SVMs is $f(x) = \langle w, \phi(x) \rangle + b$[1], S^{3}VM aims to learn a maximum margin over labeled and unlabeled samples as follows.

$$\min_{w,b} \frac{1}{2}\langle w, w \rangle + C \sum_{i=1}^{l} h_1(y_i f(x_i)) + C^* \sum_{i=l+1}^{l+u} h_1(|f(x_i)|) \quad (1)$$

where $h_t(\cdot) = \max(0, t - \cdot)$ is the hinge loss, $h_t(|\cdot|)$ is the symmetric hinge loss, C and C^* are predefined parameters[2]. S^{3}VM is one of the most popular methods for semi-supervised learning.

The real-world tasks of S^{3}VM often lie in high dimensions with few labeled samples [1]. It is highly possible to generate an imbalanced prediction (*i.e.*, the number of samples classified to one class is significantly higher than those classified to another class) which can greatly bring down the generalization performance of S^{3}VM.

*Both authors contributed equally to the paper

†To whom all correspondence should be addressed. H.H. was partially supported by U.S. NSF IIS 1836945, IIS 1836938, DBI 1836866, IIS 1845666, IIS 1852606, IIS 1838627, IIS 1837956.

[1] w and b are the parameters of model function, and $\phi(\cdot)$ is a transformation function from an input space to a high-dimensional reproducing kernel Hilbert space.

[2] If $C^* = 0$, the Eq. (1) degenerates to the standard SVM optimization problem. For $C^* > 0$, we use the symmetric hinge loss to penalize the unlabeled data inside the margin.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330962

To avoid the trivial solution as discussed above, multiple methods have been proposed to cure the imbalanced classification problem, such as resampling technique and the use of balancing constraint. Specifically, Mitra et al. [15], Chen et al. [16], Valentini [17] and Li et al. [18] tried to deal with the issue of imbalanced classification by either adopting under-sampling to select a subset of negative training examples, over-sampling to generate more positive examples, or a combination of these two methods. Joachims [10], Chapelle and Zien [9], Sindhwani and Keerthi [11], Tian et al. [12] and Chapelle et al. [13] used various balancing constraints, which is a more popular method for handling the issue of imbalanced classification compared with other techniques. Specifically, to make the S^{3}VM models more suitable for imbalanced datasets, Chapelle and Zien [9] modified these models by introducing a slightly relaxed balancing constraint (denoted as BCS^{3}VM), *i.e.* $\frac{1}{u}\sum_{i=l+1}^{l+u} f(x_i) = \frac{1}{l}\sum_{i=1}^{l} y_i$, which aims to ensure that the fraction of positive and negatives assigned to the unlabeled data should be the same fraction as found in the labeled data. This paper focuses on the BCS^{3}VM because of its effectiveness and popularity.

Although BCS^{3}VM algorithm is an improved version of S^{3}VM which can effectively avoid the trivial solution of classifying all the unlabeled examples into a same class, its objective formulation is still a non-convex optimization problem like S^{3}VM. Fung and Mangasarian [2], Collobert et al. [1], and Wang et al. [3] applied the concave-convex procedure (CCCP) to solve the non-convex problem of S^{3}VM, which also can be an effective method to solve BCS^{3}VM. However, those methods usually have rather high computational cost which can hinder the application of S^{3}VM models in large scale datasets. As pointed by Chapelle and Zien [9], scaling up BCS^{3}VM is still an open question.

Incremental learning is an important method for processing large amounts of data using comparatively smaller computing resources [22]. Several incremental learning algorithms have been proposed for SVMs, such as [4–6]. Specifically, these incremental algorithms used the path following technique [24, 25] to update the solutions by maintaing the KKT conditions [26]. Compared with other techniques for incremental learning, path following technique is more often adopted due to its efficiency and convergence guarantee. Recently, targeted at the non-convex problem of S^{3}VM, a novel incremental learning algorithm (IL-S^{3}VM) based on the path following technique in the framework of CCCP [7] was proposed, which can not only solve the non-convex problem but also significantly reduce the computational complexity of S^{3}VM. Thus inspired by IL-S^{3}VM, incremental learning also could be an effective method to scale up BCS^{3}VM. However, the dynamic relationship of balancing constraint with previous labeled and unlabeled samples impede their incremental method for handling BCS^{3}VM.

To fill this gap, in this paper, we propose a new incremental S^{3}VM algorithm with balancing constraints (IL-BCS^{3}VM) based on IL-S^{3}VM, which can effectively handle the balancing constraint and directly update the solution of BCS^{3}VM. Specifically, in order to handle the dynamic relationship of balancing constraint with previous labeled and unlabeled samples, we design two unique procedures in our IL-BCS^{3}VM algorithm, which can respectively eliminate and add the balancing constraint into S^{3}VM, so that the efficient incremental learning for BCS^{3}VM can be achieved. What's more, we provide the finite convergence analysis for IL-BCS^{3}VM. Experimental results on a variety of benchmark datasets not only confirm the finite convergence of IL-BCS^{3}VM, but also show a huge reduction of computational time compared with existing batch and incremental learning algorithms, while retaining the similar generalization performance.

Contributions. The main contributions of this paper are summarized as follows.

(1) We propose a new incremental S^{3}VM learning algorithm with balancing constraint (IL-BCS^{3}VM) targeted at more realistic classification problems that often arise in S^{3}VM. To the best of our knowledge, IL-BCS^{3}VM is the first path following algorithm by overcoming the challenge of the dynamic relationship of balancing constraint with previous labeled and unlabeled samples.

(2) Though our IL-BCS^{3}VM can handle a more complicated problem compared with the one of IL-S^{3}VM, we prove that it converges to a local minimal, and the computational complexity is still in the same scale with IL-S^{3}VM which is significantly cheaper than the ones of most existing BCS^{3}VM algorithms.

2 REVISIT OF IL-S^{3}VM

In this section, we briefly revisit the CCCP formulation of IL-S^{3}VM. After that we introduce the main idea of IL-S^{3}VM algorithm.

2.1 CCCP Formulation of IL-S^{3}VM

According to the theory of CCCP [14], to solve the objective function (1), we need to reformulate the objective into a summation of a convex function $J_{vex}(\theta)$ and a concave function $J_{cav}(\theta)$, where θ is the parameters of the model. Because every unlabeled sample has the possibility of being positive or negative label, the unlabeled dataset U should be doubled and a new artificial labeled dataset $\widetilde{U} = \{(x_{l+1}, +1), \cdots, (x_{l+u}, +1), (x_{l+u+1}, -1), \cdots, (x_{l+2u}, -1)\}$ is created. Thus, the original formulation Eq. (1) can be transformed as follows:

$$\min_{w,b} \underbrace{\frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{l} h_1(y_i f(x_i)) + C^*\sum_{i=l+1}^{l+2u} h_1(y_i f(x_i))}_{J_{vex}(\theta)} \underbrace{-C^*\sum_{i=l+1}^{l+2u} h_0(y_i f(x_i))}_{J_{cav}(\theta)} \qquad (2)$$

During the process of solving Eq. (2), we use μ_i which is defined in Eq. (3) to simplify the calculation procedure of the CCCP.

$$\mu_i = y_i \frac{\partial J_{cav}(\theta)}{\partial f(x_i)} = \begin{cases} C^* & \text{if} \quad y_i f(x_i) < 0, \ i \geq l+1 \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Then we can obtain the primal convex inner loop (denoted as CIL) problem for Eq. (2) based on the CCCP which is skipped here. We directly show the corresponding dual CIL problem as follows [1]:

$$\begin{aligned} \min_{\widetilde{\alpha}} \quad & \frac{1}{2}\widetilde{\alpha}^T H \widetilde{\alpha} - y^T \widetilde{\alpha} \\ s.t. \quad & \sum_{i=1}^{l+2u} \widetilde{\alpha}_i = 0 \quad \underline{C}_i \le \widetilde{\alpha}_i \le \overline{C}_i, \ i = 1, \cdots, l+2u. \end{aligned} \tag{4}$$

where H is a positive semidefinite matrix with $H_{ij} = K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ for all $1 \le i, j \le l+2u$, $K(x_i, x_j)$ is the kernel function. $\widetilde{\alpha}_i = y_i(\alpha_i - \mu_i)$, $\underline{C}_i$ and $\overline{C}_i$ are defined as follows:

$$\underline{C}_i = \begin{cases} -\mu_i & \text{if} \quad y_i = +1 \\ \mu_i - C & \text{if} \quad y_i = -1, 1 \le i \le l \\ \mu_i - C^* & \text{if} \quad y_i = -1, i \ge l+1 \end{cases} \tag{5}$$

$$\overline{C}_i = \begin{cases} C - \mu_i & \text{if} \quad y_i = +1, 1 \le i \le l \\ -\mu_i & \text{if} \quad y_i = +1, i \ge l+1 \\ \mu_i & \text{if} \quad y_i = -1 \end{cases} \tag{6}$$

2.2 IL-S^{3}VM

According to the convex optimization theory [8], by introducing Lagrangian multiplier b' corresponding to the constraint in Eq. (4), the dual CIL problem (*i.e.*, Eq. (4)) can be transformed as follows.

$$W = \min_{\underline{C} \le \widetilde{\alpha} \le \overline{C}} \max_{b'} \frac{1}{2}\widetilde{\alpha}^T H \widetilde{\alpha} - y^T \widetilde{\alpha} + b' \left(\sum_{i=1}^{l+2u} \widetilde{\alpha}_i \right) \tag{7}$$

The first-order partial derivative of W leads to the following KKT conditions.

$$\frac{\partial W}{\partial b'} = \sum_{i=1}^{l+2u} \widetilde{\alpha}_i = 0 \tag{8}$$

$$g_i \overset{\text{def}}{=} \frac{\partial W}{\partial \widetilde{\alpha}_i} = \sum_{j=1}^{l+2u} \widetilde{\alpha}_j H_{ij} + b' - y_i, \quad \forall i \ge 1$$

$$\begin{cases} > 0 & \text{then} \quad \widetilde{\alpha}_i = \underline{C}_i & O \\ = 0 & \text{then} \quad \underline{C}_i \le \widetilde{\alpha}_i \le \overline{C}_i & M \\ < 0 & \text{then} \quad \widetilde{\alpha}_i = \overline{C}_i & E \end{cases} \tag{9}$$

Correspondingly, the extended training dataset $L \cup \widetilde{U}$ can be partitioned into three categories as $S = \{M, E, O\}$, which is shown in Eq. (9).

We define the additions in L and $\widetilde{U}$ as L_N and $\widetilde{U}_N$ respectively. When a new sample from $L_N \cup \widetilde{U}_N$ joins the original set $L \cup \widetilde{U}$, the change of μ_i could lead to the result that the corresponding samples violate the KKT-conditions. To handle this situation, Gu et al. [7] defined a KKT-violating set A[3]. The fundamental principle of IL-S^{3}VM is to constantly detect new samples violating the KKT conditions and add these samples into the KKT-violating set A, while pushing the samples in A to satisfy the KKT conditions (see Figure 1 Step 2: IL-S^{3}VM algorithm).

Specifically, to achieve this goal, two main issues need to be addressed for designing IL-S^{3}VM algorithm [7]:

(1) **Compute the direction of $\Delta\widetilde{\alpha}$:** Set $\Delta\widetilde{\alpha}_A$ as the changes of the weights of set A, and set the direction to $\Delta\widetilde{\alpha}_A$ as $d_A = \widetilde{C}_A - \widetilde{\alpha}_A$, where $\widetilde{C}_i = \overline{C}_i$, if $y_i = +1$, otherwise $\widetilde{C}_i = \underline{C}_i$. Thus we have $\Delta\widetilde{\alpha}_A = \eta \cdot d_A$, where η is a parameter with $0 \le \eta \le 1$ to control the adjustment qualities of $\widetilde{\alpha}_A$, and the direction of $\Delta\widetilde{\alpha}_A$ with respect to η can be obtained by solving the following linear system:

$$\begin{bmatrix} 0 & 1_M^T \\ 1_M & H_{MM} \end{bmatrix} \begin{bmatrix} d_{b'} \\ d_M \end{bmatrix} = - \begin{bmatrix} 1_A^T \\ H_{MA} \end{bmatrix} d_A \tag{10}$$

where $d_{b'}$ and d_M refer to the directions of the $\Delta b'$ and $\Delta\widetilde{\alpha}_M$ respectively. Furthermore, with the the conclusion as stated above, the linear relationship between Δg_i ($\forall i \in E \cup O \cup A$) and η can be obtained as follows:

$$d_{g_i} = \sum_{j \in A} H_{ij} d_j + \sum_{j \in M} H_{ij} d_j + d_{b'} \tag{11}$$

(2) **Compute the maximum adjustment quantity $\eta^{\max}$:** The maximum adjustment quantity η^{max} of η can be calculated by solving a series of linear inequalities based on three conditions [4] as marked by different arrow lines in Step 2 of IL-S^{3}VM algorithm in Figure 1 (the blue arrow lines show the adjustments of the sample corresponding to the change of the value of μ_i in Eq. (3)).

Based on the two issues discussed above, IL-S^{3}VM algorithm can be summarized in Algorithm 1.

Algorithm 1 IL-S^{3}VM

```
Input: $\widetilde{\alpha}, b', M, E, O, \overline{C}, \underline{C}$ and $L_N \cup \widetilde{U}_N$.
Output: $\widetilde{\alpha}, b', \underline{C}, \overline{C}, M, E$, and $O$.
while $L_N \cup \widetilde{U}_N \neq \emptyset$ do
    Read a new sample $(x_c, y_c)$ from $L_N \cup \widetilde{U}_N$.
    Remove $(x_c, y_c)$ from $L_N \cup \widetilde{U}_N$.
    Initialize $\widetilde{\alpha}_c = 0$ and comput $\overline{C}_c$, $\underline{C}_c$ and $g_c$.
    Add $(x_c, y_c)$ into $M$, $O$ or $A$ according to $g_c$.
    while $A \neq \emptyset$ do
        Compute $d_{b'}, d_M, d_A$ and $d_g$.
        Compute the maximal quantity $\eta^{max}$.
        Update $\widetilde{\alpha}_A$, $\widetilde{\alpha}$, $b'$, $g$, $\underline{C}$, $\overline{C}$, $A$, $M$, $E$, $O$.
    end while
end while
```

3 NEW INCREMENTAL S^{3}VM LEARNING ALGORITHM WITH BALANCING CONSTRAINT

In this section, we first introduce the principle of our IL-BCS^{3}VM algorithm, then present the IL-BCS^{3}VM algorithm in detail.

[3] The KKT-violating set A is defined as a subset of an union of $\widetilde{U}$ and an added sample (x_c, y_c), such that all the samples violating the KKT conditions are included in A.

[4] The three conditions are 1) a sample migrate among the sets M, E, O; 2) the KKT conditions for one sample in A will be satisfied; 3) the sample in $O \cup E$ violate the KKT conditions, *i.e.* we need to update the values of μ_i in Eq. (3).

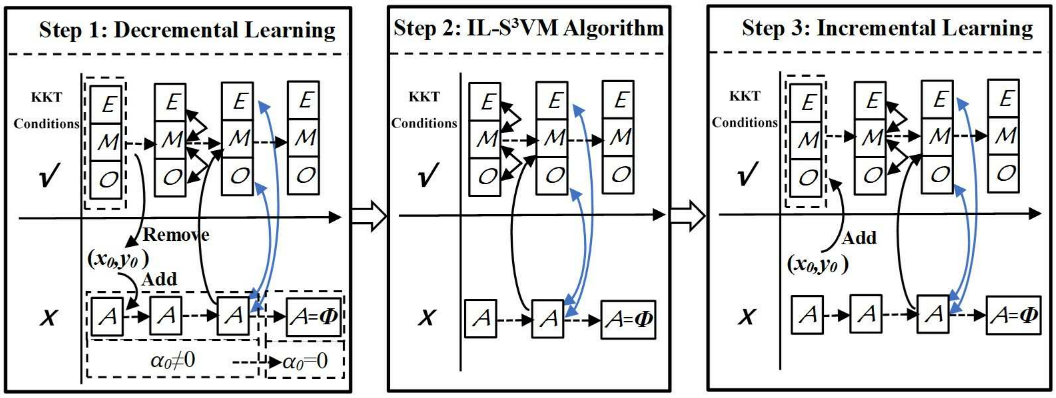


Figure 1: IL-BCS^{3}VM algorithm. Three steps (*i.e.*, decremental learning, IL-S^{3}VM and incremental learning respectively) are involved in IL-BCS^{3}VM.

3.1 Principle of our IL-BCS^{3}VM Algorithm

As discussed before, BCS^{3}VM can effectively avoid imbalanced classification problems, but it has a high computational complexity same as S^{3}VM. Thus it is highly desired to design an efficient incremental algorithm for the BCS^{3}VM problem which can reduce the computational complexity significantly. Inspired by the IL-S^{3}VM algorithm which is an effective method to scale up S^{3}VM, we also introduce an incremental learning method to BCS^{3}VM. By overcoming the challenge of dynamic relationships of balancing constraint with previous labeled and unlabeled samples during incremental learning process, we propose a new incremental learning algorithm based on the path following technique for S^{3}VM with balancing constraint (denoted as IL-BCS^{3}VM), which can not only effectively handle the balancing constraint but also significantly improve the algorithmic efficiency of standard BCS^{3}VM.

To apply the balancing constraint to IL-S^{3}VM, we introduce a new Lagrangian multiplier α_0 corresponding to the balancing constraint to the original optimization problem of IL-S^{3}VM (*i.e.*, Eq. (4)). After a series of Lagrangian transformations, we can obtain the dual CIL problem for IL-BCS^{3}VM as follows [1].

$$\begin{aligned} \min_{\widetilde{\alpha}} \quad & \frac{1}{2}\widetilde{\alpha}^T H\widetilde{\alpha} - y^T\widetilde{\alpha} \\ s.t. \quad & \sum_{i=0}^{l+2u}\widetilde{\alpha}_i = 0; \quad \underline{C}_i \le \widetilde{\alpha}_i \le \overline{C}_i,\ i = 1,\cdots,l+2u \end{aligned} \tag{12}$$

Note that different from the dual CIL problem for IL-S^{3}VM (*i.e.*, Eq. (4)), the value of parameter i in Eq. (12) starts from 0, and sample (x_0, y_0) is generated from the balancing constraint. The parameters of virtual sample (x_0, y_0) are set as follows [1]:

$$\widetilde{\alpha}_0 = \alpha_0, \quad y_0 = \frac{1}{l}\sum_{i=1}^{l} y_i \quad \phi(x_0) = \frac{1}{u}\sum_{i=l+1}^{l+u}\phi(x_i) \tag{13}$$

If the Lagrangian multiplier α_0 for balancing constraint constantly equals to 0, the dual CIL problem for IL-BCS^{3}VM degenerates to the CIL problem for IL-S^{3}VM.

According to the convex optimization theory [8], by introducing the Lagrangian multiplier b' to Eq. (12), we can obtain the CIL problem for IL-BCS^{3}VM as follows:

$$W = \min_{\underline{C}\le\widetilde{\alpha}\le\overline{C}} \max_{b'} \frac{1}{2}\widetilde{\alpha}^T H\widetilde{\alpha} - y^T\widetilde{\alpha} + b'\left(\sum_{i=0}^{l+2u}\widetilde{\alpha}_i\right) \tag{14}$$

We can find the first-order partial derivative of W which leads to the following KKT conditions in Eqs. (15)-(17), similar to Eqs. (8)-(9). Note that the virtual sample (x_0, y_0) needs a special consideration due to its unique features.

$$\frac{\partial W}{\partial b'} = \sum_{i=0}^{l+2u}\widetilde{\alpha}_i = 0 \tag{15}$$

$$g_i \overset{\text{def}}{=} \frac{\partial W}{\partial \widetilde{\alpha}_i} = \sum_{j=0}^{l+2u}\widetilde{\alpha}_j H_{ij} + b' - y_i, \quad \forall i \ge 1 \tag{16}$$

When $i = 0$, according to the balancing constraint, *i.e.* $\frac{1}{u}\sum_{i=l+1}^{l+u} f(x_i) = \frac{1}{l}\sum_{i=1}^{l} y_i$, we can have the relationship for g_0 as follows.

$$\begin{aligned} g_0 = \frac{\partial W}{\partial \widetilde{\alpha}_0} &= \sum_{j=0}^{l+2u}\widetilde{\alpha}_j H_{0j} + b' - y_0 \\ &= -b + b' \end{aligned} \tag{17}$$

According to the value of g_i in Eq. (16) and Eq. (17), we also partition the extended dataset $L \cup \widetilde{U}$ in three categories as $S = \{M, E, O\}$ as follows:

$$\begin{aligned} M &= \left\{i \in L \cup \widetilde{U} : g_i = 0,\ \underline{C}_i \le \widetilde{\alpha}_i \le \overline{C}_i\right\} \\ E &= \left\{i \in L \cup \widetilde{U} : g_i < 0,\ \widetilde{\alpha}_i = \overline{C}_i\right\} \\ O &= \left\{i \in L \cup \widetilde{U} : g_i > 0,\ \widetilde{\alpha}_i = \underline{C}_i\right\} \end{aligned} \tag{18}$$

In order to explain our IL-BCS^{3}VM algorithm more clearly, we first define the entire new data set which will be added into the original set as S_t. S_t consists of labeled data set L_t and unlabeled data set $\widetilde{U_t}$, i.e, $S_t = L_t \cup \widetilde{U_t}$. l_t and u_t are the number of elements in the sets L_t and set U_t respectively. Every time a batch of data S_t is added into the original set, the balancing constraint need to be adjusted once to ensure data balance in the entire dataset including the new added one. After the new dataset S_t and the original dataset $L \cup \widetilde{U}$ are merged, the original labeled set L becomes L_{new}, the original unlabeled set $\widetilde{U}$ becomes $\widetilde{U}_{new}$. Parameters l and u are updated to $l + l_t$ and $u + u_t$ respectively. Thus, how to handle the dynamic relationships between balancing constraint with previous labeled and unlabeled samples is the main challenge for designing the incremental learning algorithm of BCS^{3}VM.

From Eq. (13), we can find that the balancing sample (x_0, y_0) represents, in a way, the mean value of the rest of the samples in the new dataset $L_{new} \cup \widetilde{U}_{new}$, and the balancing constraint has turned into the form of sample (x_0, y_0). Thus the problem of how to apply balancing constraint to the optimization problem can be transformed into how to deal with sample (x_0, y_0). To simplify the process of our algorithm, we define a virtual balancing dataset V and classify sample (x_0, y_0) into set V separately. Balancing dataset V is formally defined as follows:

$$V = \{(x_0, y_0)\} \tag{19}$$

Thus the virtual balancing dataset V and the new input dataset S_t are two independent sets which should be considered carefully by the incremental learning algorithm.

However, before adding the two datasets (*i.e.*, V and S_t) into the original dataset $L \cup \widetilde{U}$, we need to find out whether an original balancing constraint is already applied to $L \cup \widetilde{U}$, so as to avoid the conflict between the original and the new balancing constraint. If so, the original balancing constraint need to be eliminated first by using decremental learning methods. After the above operation is completed, we can add the two datasets into the original dataset $L \cup \widetilde{U}$ by using incremental learning methods. Note that the virtual balancing dataset V should be added after the new input dataset S_t, because balancing constraint is applied once after a bach of data S_t have all been added into the original set $L \cup \widetilde{U}$.

When a sample is added into or removed from the original set $L \cup \widetilde{U}$, our fundamental principle is to ensure all the samples meet KKT conditions simultaneously by constantly detecting KKT-violating samples to add into set A (defined in section 2.2) and pushing these samples satisfying KKT-conditions same to [7].

3.2 IL-BCS^{3}VM Algorithm

Our IL-BCS^{3}VM Algorithm consists of three steps (see Figure 1). First of all, in Step 1 (Section 3.2.1), we need to find out whether an original balancing constraint is already applied to the original dataset $L \cup \widetilde{U}$, *i.e.* check whether the Lagrangian multiplier α_0 for balancing constraint equals 0. If not, we need to eliminate the original balancing constraint using decremental learning methods. Then in Step 2, we add the new input dataset S_t into the original set $L \cup \widetilde{U}$ using IL-S^{3}VM algorithms (please see Section 2.2). After set S_t becomes an empty set and the original dataset becomes $L_{new} \cup \widetilde{U}_{new}$, in Step 3 (Section 3.2.2), we can apply balancing constraint to set $L_{new} \cup \widetilde{U}_{new}$ by adding balancing dataset V into $L_{new} \cup \widetilde{U}_{new}$ using incremental learning methods. When the above three steps are completed, we can expand the local balance on the original dataset $L \cup \widetilde{U}$ to the global balance on the entire new dataset $L_{new} \cup \widetilde{U}_{new}$.

3.2.1 Step 1: Eliminate the Balancing Constraint in S^{3}VM. In order to eliminate the original balancing constraint imposed on the original dataset $L \cup \widetilde{U}$, we apply decremental learning method in this step to decrease the Lagrangian multiplier α_0 of the balancing constraint to 0, so that we can remove the original balancing sample (x_0, y_0) out of set $L \cup \widetilde{U}$ (see Step 1: Decremental Learning in Figure 1).

Similar to the incremental learning process in IL-S^{3}VM algorithm, during the decremental learning process, the migrations of samples among sets also could lead to the changes of the weights of sets and corresponding parameters. The update of the parameters (*i.e.* updating $\widetilde{\alpha}_A \leftarrow \widetilde{\alpha}_A + d_A\eta^{max}, \alpha_M \leftarrow \alpha_M + d_M\eta^{max}, b' \leftarrow b' + d_{b'}\eta^{max}, g \leftarrow g + d_g\eta^{max}$) and the migration of samples among the sets M, E, O, A in decremental learning is the same as incremental learning as discussed in Section 2.2. Note that the direction of the changes of the parameters in decremental learning is contrary to incremental learning method. Thus when $(x_i, y_i) \in A$, we set $\widetilde{C}_i = \underline{C}_i$, if $y_i = +1$, otherwise $\widetilde{C}_i = \overline{C}_i$. Especially, the update of the parameters for the balancing sample (x_0, y_0) can be simplified due to its special features. When we update $g_0((x_0, y_0) \in E \cup O \cup A)$, instead of solving Eq. (11), we can directly obtain $d_{g_0} = d_{b'}$ from Eq. (17). So that when $i = 0$, g_i and b' can be updated simultaneously.

In the complete decremental learning process, first of all, we suppose (x_0, y_0) is a KKT-violating sample, and remove it from M, E or O to add into A. Then we compute the directions of the parameters, *i.e.* $d_{b'}, d_M, d_g, d_A$, and find the maximum adjustment quantity η^{max} of η. After that, we can update $\alpha, b', \underline{C}, \overline{C}, A, M, E$ and O correspondingly. Repeating the above procedures until the Lagrangian multiplier α_0 is reduced to 0 and the set A becomes an empty set. At this point, the original balancing constraint is eliminated and all the samples in $L \cup \widetilde{U}$ satisfy KKT conditions simultaneously. The decremental learning process is summarized in Algorithm 2.

Algorithm 2 Decremental Learning for α_0

```
Input: $\widetilde{\alpha}, b', \underline{C}, \overline{C}, M, E, O$
Output: $\widetilde{\alpha}, b', \underline{C}, \overline{C}, M, E, O$
while $\alpha_0 \neq 0$ do
    Remove $(x_0, y_0)$ from $M, E$ or $O$.
    Add $(x_0, y_0)$ into $A$.
    while $A \neq \emptyset$ do
        Compute $d_b, d_M, d_A$ and $d_g$.
        Compute the maximal quantity $\eta^{max}$.
        Update $\widetilde{\alpha}_A, \widetilde{\alpha}, b', g, \underline{C}, \overline{C}, A, M, E, O$.
    end while
end while
```

3.2.2 Step 3: Add the Balancing Constraint in S^{3}VM. In Step 3, to apply the balancing constraint to S^{3}VM, we add the balancing dataset V into $L_{new} \cup \widetilde{U}_{new}$ using the incremental learning method (see Step 3: Incremental Learning in Figure 1). The update of the parameters and the migration of samples among the sets M, E, O, A in this step also remain the same as the incremental learning process as discussed in Section 2.2. Especially, for the update of the parameters of the balancing sample (x_0, y_0), please refer to Step 1.

In the complete incremental learning process, we first remove (x_0, y_0) from set V and add it into $L_{new} \cup \widetilde{U}_{new}$. Then we compute $d_{b'}, d_M, d_A$ and d_g and find the maximum adjustment quantity

η^{max} of η. After that, we can update the $\alpha_c, \alpha, b', \underline{C}, \overline{C}, A, M, E$ and O correspondingly. Repeating the above procedures, until set A is empty. This procedure is summarized in Algorithm 3.

Algorithm 3 Incremental Learning for α_0

```
Input: $\widetilde{\alpha}, b', \underline{C}, \overline{C}, M, E, O$
Output: $\widetilde{\alpha}, b', \underline{C}, \overline{C}, M, E, O$
Initialize $\widetilde{\alpha}_0 = 0$ and compute $\underline{C}, \overline{C}, g_0$.
Add $(x_0, y_0)$ into $M, E, O$ or $A$ according to $g_0$.
while $A \neq \emptyset$ do
    Compute $d_{b'}, d_M, d_A$ and $d_g$.
    Compute the maximal quantity $\eta^{max}$.
    Update $\widetilde{\alpha}_A, \widetilde{\alpha}, b', g, \underline{C}, \overline{C}, A, M, E, O$.
end while
```

4 ANALYSIS AND DISCUSSION

In this section, we first prove the finite convergence of IL-BCS^{3}VM, then provide the time complexity analysis of IL-BCS^{3}VM.

4.1 Finite Convergence Analysis for IL-BCS^{3}VM Algorithm

In this section, we prove that IL-BCS^{3}VM can converge to a local minimal in a finite number of iterations (Theorem 3).

Our IL-BCS^{3}VM algorithm consists of three steps, *i.e.* decremental learning (*i.e.*, Step 1), IL-S^{3}VM algorithm (*i.e.*, Step 2) and incremental learning (*i.e.*, Step 3) as discussed in Section 3.2. The finite convergence of IL-S^{3}VM was already proven in [7]. Thus we only need to prove the finite convergence of the decremental learning (*i.e.*, Step 1) and the incremental learning (*i.e.*, Step 3). We first prove Theorem 1 as follows.

Theorem 1. *During the process of decremental learning (*i.e.*, Step 1) and incremental learning (*i.e.*, Step 3), any sample from $L \cup \widetilde{U} \cup V$ cannot migrate back and forth in successive adjustment steps among the sets M, E, O and A.*

Sketch of Proof As discussed in Section 3.2, the migration of the samples during incremental and decremental learning process is the same. Thus similar to the proof of Theorem 2 in [7], for sample (x_t, y_t) where $t \geq 0$, it is easy to verify the following four sub-conclusions: 1) if a sample (x_t, y_t) is added into the set M, then (x_t, y_t) will not be removed from M in the immediate next adjustment. 2) If (x_t, y_t) is removed from the set M, then (x_t, y_t) will not be added into M in the immediate next adjustment. 3) If (x_t, y_t) is removed from the set E or O and added into the set A, then (x_t, y_t) will not be removed from A in the immediate next adjustment. 4) If (x_t, y_t) is removed from the set A and added into the set E, M or O, then (x_t, y_t) will not be removed from E, M or O in the immediate next adjustment. □

According to Theorem 1, we can have Corollary 1 as follows.

Corollary 1. *For each adjustment of IL-BCS^{3}VM, the maximum adjustment η^{max} is greater than zero.*

Similar to the proof of Corollary 1 in [7] and Lemma 4 in [28], Corollary 1 can be easily proven. Based on this corollary, we can prove that the objective function W (see Eq. (14)) is strictly monotonically decreasing and increasing under different conditions in Theorem 2 as follows:

Theorem 2. *During the process of decremental learning and incremental learning, the objective function W has the following properties.*

(1) If A only includes the new added balancing sample (x_0, y_0), i.e., *$A = (x_0, y_0)$, W is strictly monotonically decreasing.*
(2) If A does not include the new added balancing sample (x_0, y_0) and $A \neq \emptyset$, W is strictly monotonically increasing.

Sketch of Proof Suppose that the previous adjustment is indexed by k, the immediate next is indexed by $k+1$, and let $\alpha_E = 0$, $\alpha_O = 0$, $S = M \cup E \cup O \cup A$, $V = \{(x_0, y_0)\}$ as we explained before. Then similar to the proof of Theorem 3 in [7], we can have the conclusion that $W^{[k+1]} - W^{[k]} = \eta^{max} \sum_{i \in A} d_i (g_i^{[k]} + \frac{1}{2} d_{g_i}{}^{[k]} \eta^{max})$ and $\sum_{j \in A} d_j d_{g_j} \geq 0$. Consequently, if $A = (x_0, y_0)$, $d_0 g_0^{[k]} < 0$ can be easily verified. Thus we have $W^{[k+1]} - W^{[k]} < 0$. If A does not include (x_0, y_0) and $A \neq \emptyset$, $\sum_{i \in A} d_i g_i^{[k]} > 0$ can be easily verified. Thus we have $W^{[k+1]} - W^{[k]} > 0$. □

Based on Theorem 2, we can prove the convergence of our IL-BCS^{3}VM in Theorem 3.

Theorem 3. *IL-BCS^{3}VM can converge to a local minimal in a finite number of iterations.*

Sketch of Proof Similar to the proof of Theorem 4 in [7], the finite convergence of the process of decremental learning (Step 1) and incremental learning (Step 3) can be easily proven. Therefore, all three steps of IL-BCS^{3}VM converge to a local minimal, and the finite convergence of our IL-BCS^{3}VM can be proven. □

4.2 Time Complexity Analysis

We will analyse the computational complexity according to the three steps (stated in section 3.2) of our IL-BCS^{3}VM respectively. The time complexity of IL-S^{3}VM is $O(|M|(l+2u)^2 + |M|^2(l+2u))$ [7]. The update of the parameters and sets in incremental learning is the same as decremental learning except for the directions, so the computational complexity of the two learning process is the same. Similar to IL-S^{3}VM [7], the time complexity of each iteration during incremental learning and decemental learning process is $O(|M|(l+2u) + |M|^2)$. Our IL-BCS^{3}VM can converge to a local minimal in a finite number of iterations which is proven in Theorem 3. Besides, the number of iteration steps is rather a small number compared with $l+2u$, which can be verified by the experiments. We define the number of iterations as c, where $c \ll l+2u$, then the time complexity of both incremental learning process and decremental learning process is $O(c(|M|(l+2u) + |M|^2))$. Therefore, the computational complexity of IL-BCS^{3}VM is $O(|M|(l+2u)^2 + |M|^2(l+2u))$, which still scales the same as IL-S^{3}VM. That is to say, our IL-BCS^{3}VM can improve the algorithmic efficiency and reduce the computational complexity significantly compared with existing BCS^{3}VM algorithms.

5 EXPERIMENTS

In this section, we first provide the experimental setup, and then provide the experimental results and discussions.

5.1 Experimental Setup

Design of experiments: In the experiments, we first show the effectiveness of our IL-BCS^{3}VM, and then demonstrate the advantage of our IL-BCS^{3}VM in terms of computational efficiency and classification accuracy.

To verify the effectiveness of our IL-BCS^{3}VM, we investigate the convergence of IL-BCS^{3}VM by counting the numbers of iterations during the adjustments whenever a sample is added into the datasets, over 20 trails.

In order to demonstrate the great algorithmic efficiency and classification accuracy of our IL-BCS^{3}VM over other batch and incremental S^{3}VM algorithms, we compare the running time and unlabeled accuracy of our IL-BCS^{3}VM with other algorithms. Specifically, the compared algorithms are summarized as follows:

1. BL-S^{3}VM (also called UniverSVM [27]): the state-of-the art batch S^{3}VM algorithm based on the CCCP algorithm and SMO algorithm.
2. BCS^{3}VM (also called CCCP-TSVM [1]): a S^{3}VM algorithm with balancing constraint based on the CCCP algorithm.
3. IL-BCS^{3}VM: our proposed incremental S^{3}VM learning algorithm with balancing constraint.

Implementation: We implement our IL-BCS^{3}VM in MATLAB. BL-S^{3}VM based on CCCP algorithm proposed by Sinz and Roffilli was implemented in C/C++. To compare the run-time in the same platform, we implement BL-S^{3}VM in MATLAB. Besides, BCS^{3}VM is also implemented in MATLAB. For kernels, the linear kernel, polynomial kernel $K(x_1, x_2) = (x_1 \cdot x_2 + 1)^d$ with $d = 2$, and Gaussian kernel $K(x_1, x_2) = exp(-k||x_1 - x_2||^2)$ with $k = 0.1$ are used in all the experiments. The parameters C and C^* are fixed at 10 and 5 respectively.

For the experiments of showing the effectiveness of our IL-BCS^{3}VM, we add a labeled or unlabeled sample into the original training dataset at a time and count the average number of iterations on different datasets. For the experiments which compare the running time and classification accuracy, we add 20 labeled or unlabeled samples into the training dataset. Our IL-BCS^{3}VM can update the current solution to merge the 20 new (labeled or unlabeled) samples and the original dataset, while the two other algorithms need to recompute a solution from scratch. Besides, compared with BL-S^{3}VM, BCS^{3}VM and our IL-BCS^{3}VM can handle the imbalanced classification problem of the new emerged dataset and improve the accuracy by using the balancing constraint.

Datasets: Table 1 shows the nine benchmark datasets used in our experiments, which are derived from LIBSVM [5] and Olivier [6] sources. Originally, the Usps dataset has ten classes from 0 to 9. We created a binary version of Usps dataset by classifying digits 0 to 4 versus 5 to 9. Originally, these datasets are used for supervised learning. To conduct the experiments of semi-supervised learning, we transfer these fully labeled datasets to the partially labeled datasets, by randomly dropping the labels of a part of samples out. The numbers of unlabeled samples are listed in the column "Unlabeled" of Table 1.

Table 1: The real-world dasetsets used in the experiments.

Dataset	Dimensionality	Samples	Unlabeled	Source
W6a	300	17188	17000	LIBSVM
Text	7511	1946	1800	Olivier
CodRNA	8	59535	59035	LIBSVM
Usps	256	2007	1800	LIBSVM
Madelon	500	2000	1800	LIBSVM
IJCNN1	22	49990	49790	LIBSVM
A9a	123	32561	32200	LIBSVM
Mushrooms	112	8124	7900	LIBSVM
Phishing	68	11055	10800	LIBSVM

5.2 Results and Discussions

Table 2 shows the average and standard deviation of the numbers of iterations during the running time of our IL-BCS^{3}VM by adding a labeled or unlabeled sample over 20 trails. The experiments results verified that the number of the iteration steps is limited for both labeled and unlabeled samples, which means that by effectively handling the balancing sample, our IL-BCS^{3}VM can guarantee to converge to a local minimal in a finite number of iterations.

Figure 2 shows the average running time (in seconds) of BCS^{3}VM, BL-S^{3}VM and IL-BCS^{3}VM. In the notation$(\cdot)$, the abbreviations L, P and G stand for the linear, polynomial and Gaussian kernels respectively. The results of the experiments are an excellent proof that our IL-BCS^{3}VM is much faster than BCS^{3}VM and BL-S^{3}VM. That is because, BCS^{3}VM and BL-S^{3}VM need to rebuild the solution of S^{3}VM from scratch. However, by introducing the incremental learning method based on path following technique, our IL-BCS^{3}VM can directly update the solution of BCS^{3}VM to converge to a local minimal and effectively handle the balancing constraint.

Figure 3 presents the unlabeled accuracy of BCS^{3}VM, BL-S^{3}VM and IL-BCS^{3}VM over 10 trails with notched box plot, when 20 labeled and unlabeled samples are added into the original dataset using the Gaussian kernel. The results show that our IL-BCS^{3}VM has much higher accuracy than BL-S^{3}VM and almost achieve the same accuracy as BCS^{3}VM on most unlabeled dataset. These results demonstrate that by using incremental learning methods to handle balancing constraint, our IL-BCS^{3}VM is much faster than most existing batch and incremental learning algorithms while retaining the same high classification accuracy as BCS^{3}VM.

6 CONCLUSION

Although existing BCS^{3}VM algorithms can effectively avoid the trivial solution of classifying all the unlabeled examples to a same class, they still have rather high computational complexity which impedes the applications of BCS^{3}VM in large-scale problems. In this paper, we propose a new incremental algorithm for BCS^{3}VM (IL-BCS^{3}VM) based on IL-S^{3}VM. Our new IL-BCS^{3}VM algorithm can effectively handle the balancing constraint and incorporate new samples to update the solution of BCS^{3}VM by overcoming the challenge of the dynamic relationships of balancing constraint

[5] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html.
[6] http://olivier.chapelle.cc/lds/.

Table 2: Average results with the standard deviation of IL-BCS^{3}VM (adding a labeled sample and adding an unlabeled sample) over 20 trials, where linear, polynomial and Gaussian kernels were used.

Dataset	Size	Iterations (labeled)			Iterations (unlabeled)		
		Linear	Polynomial	Gaussian	Linear	Polynomial	Gaussian
W6a	4000	22.0±15.1	27.6±14.4	20.3±11.1	28.3±20.5	25.4±9.3	19.1±11.3
	8000	61.8±43.5	51.6±28.9	24.7±13.2	57.7±30.1	32.9±14.8	28.4±15.5
	12000	65.4±39.9	72.2±40.0	37.4±23.0	52.7±37.5	42.8±23.1	42.0±26.1
	16000	120.7±82.7	64.9±53.7	61.8±44.4	88.9±72.5	64.8±39.7	41.9±32.3
Text	400	15.1±1.1	8.2±25.7	2.5±1.7	11.4±2.2	3.7±1.3	2.2±1.3
	800	60.7±6.5	8.9±4.4	2.8±1.6	31.8±12.0	10.9±6.6	3.6±0.8
	1200	83.1±14.4	39.4±15.5	2.9±1.9	64.0±24.2	31.6±16.0	2.5±0.9
	1600	151.0±37.1	69.7±34.2	2.1±1.3	110.2±33.1	79.5±31.5	2.5±1.3
CodRNA	4000	18.0±11.5	3.6±2.3	16.5±2.0	31.1±5.4	26.6±7.0	17.6±4.6
	8000	22.1±17.1	11.1±7.7	21.9±2.9	23.7±12.1	32.1±12.1	22.9±6.7
	12000	22.2±14.8	14.1±8.0	26.2±6.9	59.4±23.4	60.0±42.8	27.7±13.8
	16000	39.1±13.2	21.9±13.8	32.8±7.9	74.5±40.4	59.7±37.1	30.2±14.5
Usps	400	14.7±29.9	2.7±1.8	2.8±1.7	9.4±4.7	6.5±8.9	2.8±1.1
	800	21.5±27.2	2.8±1.6	3.2±6.3	14.1±10.9	3.8±8.8	3.1±1.4
	1200	25.4±27.6	13.1±14.6	17.4±21.0	28.6±12.7	1.2±3.8	2.9±1.2
	1600	17.4±26.9	12.3±16.0	2.4±1.3	39.0±17.5	2.6±4.7	3.5±2.1
Madelon	400	34.0±60.2	1.1±1.1	1.2±1.6	15.2±22.5	1.3±0.7	1.5±1.8
	800	27.0±38.5	1.6±1.2	1.9±2.2	16.7±47.1	1.3±0.6	1.6±1.7
	1200	40.0±65.0	1.2±1.2	1.1±1.6	5.2±9.4	2.5±3.4	1.0±1.5
	1600	35.4±67.4	2.0±3.0	0.8±1.2	3.6±8.0	10.4±26.5	0.4±0.7
Ijcnn1	4000	103.4±44.2	94.8±38.0	45.7±14.6	85.6±69.9	92.5±62.7	50.8±28.0
	8000	129.0±46.3	107.0±33.5	53.4±21.4	194.4±100.2	114.4±60.8	58.6±44.0
	12000	242.8±89.1	100.1±58.0	55.2±24.1	228.8±144.2	107.0±99.1	50.6±42.1
	16000	317.8±176.0	114.6±36.2	40.0±26.8	210.6±202.1	111.4±67.8	42.7±49.0
A9a	3000	62.8±37.1	22.9±16.6	10.7±4.6	61.1±36.6	26.4±19.1	23.4±16.3
	6000	58.3±46.4	43.7±29.7	21.7±15.6	63.9±30.3	35.9±27.8	47.3±27.4
	9000	76.8±55.8	54.9±39.2	39.0±27.8	101.8±70.2	92.1±46.0	39.1±26.1
	12000	95.0±55.9	90.4±53.0	67.6±40.1	103.2±56.9	90.2±75.6	58.1±32.6
Mushrooms	1500	6.4±3.9	4.5±0.7	3.8±1.3	11.4±5.6	1.4±1.0	4.3±2.2
	3000	3.1±2.0	2.0±1.2	6.5±3.1	16.6±8.1	1.4±1.1	7.0±4.7
	4500	7.3±4.4	2.4±1.4	6.3±2.1	8.1±5.4	5.1±2.1	6.5±4.9
	6000	8.0±5.1	4.3±2.8	3.6±2.2	27.8±10.2	10.2±5.5	4.2±3.9
Phishing	1500	4.5±14.2	13.2±29.4	7.8±5.8	2.6±8.2	5.5±7.2	8.0±7.8
	3000	25.4±42.3	29.7±89.1	4.7±7.2	62.8±122.3	15.3±30.8	2.3±5.6
	4500	58.5±82.7	26.4±65.7	3.6±9.0	69.3±132.7	11.8±30.9	9.2±14.4
	6000	97.6±152.5	96.8±122.4	15.0±16.4	109.1±182.7	15.0±47.4	7.7±11.2

with previous labeled and unlabeled samples. IL-BCS^{3}VM improves the efficiency of BCS^{3}VM algorithms significantly while retaining almost the same high classification accuracy as BCS^{3}VM. What's more, we provide the finite convergence analysis for IL-BCS^{3}VM. Experimental results on a variety of benchmark datasets not only verify the finite convergence of our IL-BCS^{3}VM, but also show a huge reduction of computational time compared with existing batch and incremental learning algorithms, while retaining the similar generalization performance.

REFERENCES

[1] R. Collobert, F. Sinz, J. Weston, L. Bottou. 2006. Large scale transductive SVMs. *Journal of Machine Learning Research*, 7, Aug:1687–1712.

[2] Glenn Fung and Olvi L Mangasarian. 2001. Semi-superyised support vector machines for unlabeled data classification. *Optimization methods and software*, 15(1):29–44.

[3] Junhui Wang, Xiaotong Shen, and Wei Pan. 2007. On transductive support vector machines. *Contemporary Mathematics*, 443: 7–20.

[4] Bin Gu, Victor S Sheng, Keng Yeow Tay, Walter Romano, and Shuo Li. 2015. Incremental support vector learning for ordinal regression. *IEEE Transactions on Neural networks and learning systems*, 26(7):1403–1416.

[5] Mario Martin. 2002. On-line support vector machine regression. *European Conference on Machine Learning*, 282–294.

[6] T. Poggio and G. Cauwenberghs. 2001. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, 409–415.

[7] B. Gu, XT. Yuan, S. Chen, H. Huang. 2018. New Incremental Learning Algorithm for Semi-Supervised Support Vector Machine. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1475–1484.

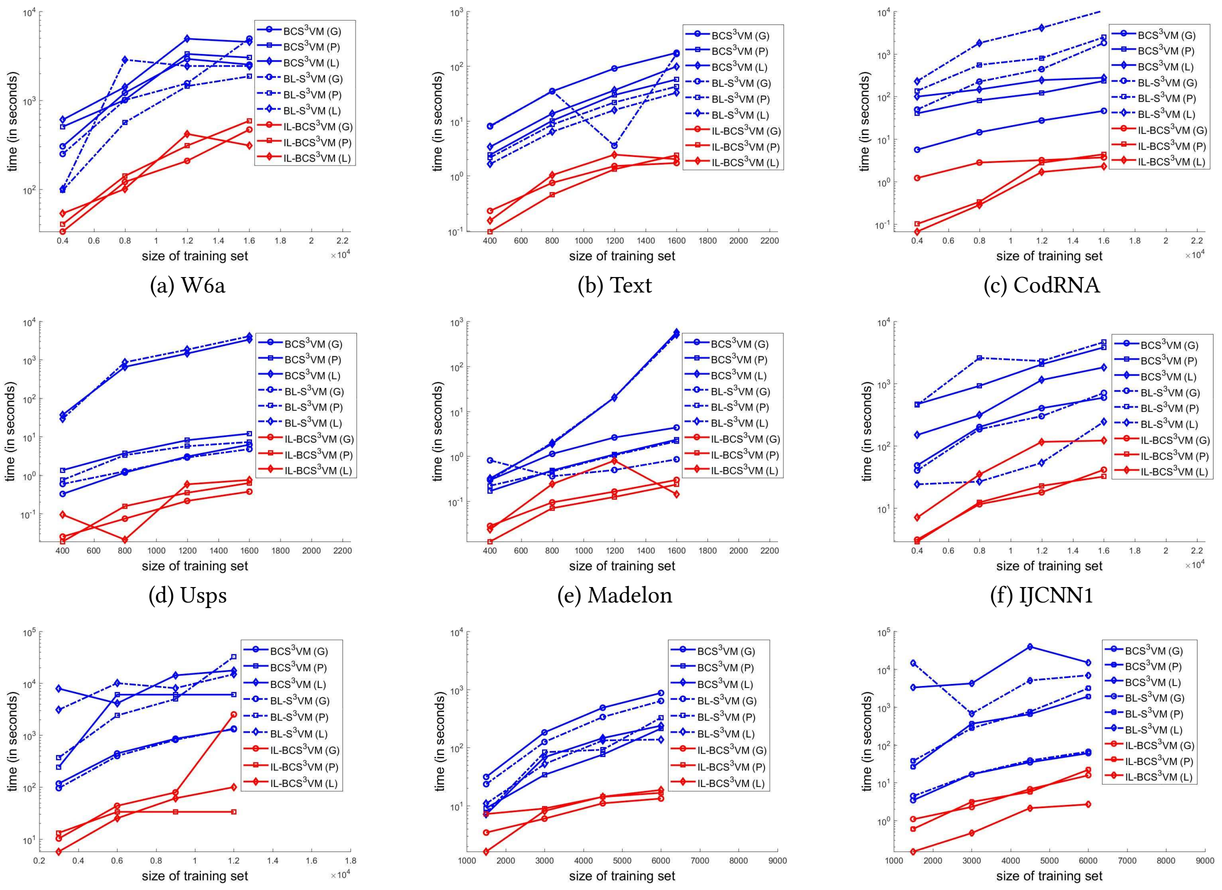
[8] Stephen Boyd and Lieven Vandenberghe. 2010. Convex optimization. Cambridge university press.

[9] O. Chapelle and A. Zien. 2005. Semi-supervised classification by low density separation. *AISTATS*, 2005:57–64.

[10] T. Joachims. 1999. Transductive inference for text classification using support vector machines. *ICML*, 99:200–209.

[11] Vikas Sindhwani and S. Sathiya Keerthi. 2006. Large scale semi-supervised linear SVMs. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 477–484.

[12] Xilan Tian, Gilles Gasso and Stéphane Canu. 2012. A multiple kernel framework for inductive semi-supervised SVM learning. *Neurocomputing*, 90:46–58.



(a) W6a

(b) Text

(c) CodRNA

(d) Usps

(e) Madelon

(f) IJCNN1

(g) A9a

(h) Mushrooms

(i)Phishing

Figure 2: Average running time (in seconds) of BCS^{3}VM, BL-S^{3}VM and IL-BCS^{3}VM over 20 trails

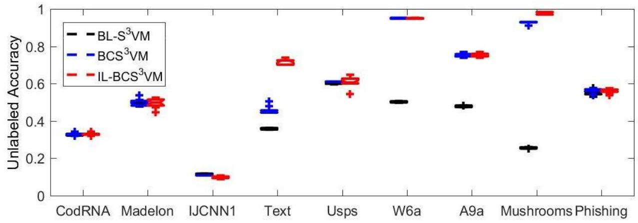


Figure 3: Unlabeled accuracy of BCS^{3}VM, BL-S^{3}VM and IL-BCS^{3}VM

[13] Olivier Chapelle, Vikas Sindhwani, Sathiya S. Keerthi. 2008. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9, Feb:203–233.

[14] A. L. Yuille and Anand Rangarajan. 2003. The concave-convex procedure. *Neural computation*, 15(4):915–936.

[15] Pabitra Mitra, C. A. Murthy and Sankar K. Pal. 2000. Data condensation in large databases by incremental learning with support vector machines. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2:708–711.

[16] Leichen Chen, Zhihua Cai, Lu Chen, Qiong Gu. 2010. A novel differential evolution-clustering hybrid resampling algorithm on imbalanced datasets. *Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on*, 81–85.

[17] G. Valentini. 2005. An experimental bias-variance analysis of SVM ensembles based on resampling techniques. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1252–1271.

[18] S. Li, Z. Wang, G. Zhou, SYM. Lee. 2011. Semi-supervised learning for imbalanced sentiment classification. *IJCAI proceedings-international joint conference on artificial intelligence*, 22(3): 1826.

[19] R. Pearson, G. Goney, and J. Shwaber. 2003. Imbalanced clustering for microarray time-series. *Proceedings of the ICML*, 3.

[20] G. Wu and E. Y. Chang. 2003. Class-boundary alignment for imbalanced dataset learning. *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*, 49–56.

[21] John Langford, Lihong Li, and Tong Zhang. 2009. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10, Mar:777–801.

[22] Léon Bottou and Yann Le Cun. 2004. Large scale online learning. *Advances in neural information processing systems*, 217–224.

[23] JC. Ang, H. Haron, HNA Hamed. 2015. Semi-supervised SVM-based feature selection for cancer classification using microarray gene expression data. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 468–477.

[24] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. 2004. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5, Oct:1391–1415.

[25] Gang Wang, Dit-Yan Yeung, and Frederick H Lochovsky. 2008. A new solution path algorithm in support vector regression. *IEEE Transactions on Neural Networks*, 19(10):1753–1767.

[26] William Karush. 1939. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago.*

[27] Fabian Sinz and Matteo Roffilli. 2012. UniverSVM. *http://mloss.org/ software/view/19/.*

[28] B.Gu and V.S.Sheng. 2013. Feasibility and finite convergence analysis for accurate on-line v-support vector machine. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(8):1304–1315