Stochastic Gradient Descent with Positive Defined Stabilized Barzilai-Borwein method

Weijuan Shi, Adibah Shuib and Zuraida Alwadood

Abstract—**As society advances, machine learning holds increasing significance. Optimization, a crucial aspect in machine learning, has garnered considerable research attention. Addressing optimization challenges has become pivotal as models grow in complexity alongside the exponential rise in data volume. In the existing algorithms like stochastic gradient descent (SGD), a common practice is to reduce step sizes or manually adjust step sizes which is inappropriate and time-consuming. In order to address this issue, researchers have put significant efforts, such as adopting the Barzilai-Borwein (BB) method. However, the BB method has its drawbacks, with the denominator potentially approaching zero or even becoming negative. In order to address this problem, this study uses the Positive Defined Stabilized Barzilai-Borwein (PDSBB) method and combined SGD algorithm with the method to create new algorithms, namely SGD-PDSBB. Following that, the algorithm's convergence is analyzed. Subsequently, its effectiveness is confirmed through numerical experiments, where is compared to the original SGD algorithm, as well as SGD-BB, in terms of step size, sub-optimality, and classification accuracy. The numerical experiments indicate that the new algorithm exhibits numerical performance similar to SGD or SGD-BB on some datasets, and on some other datasets, the new algorithms even perform better.**

Index Terms—**stochastic gradient descent, machine learning, adaptive step size, Positive Defined Stabilized Barzilai-Borwein method**

I. Introduction

MACHINE learning, as an application of artificial intelligence (AI), equips systems with the capability to access data and leverage it to perform cognitive functions. This is achieved through learning from past experiences, enabling machines to continually improve and effectively address complex problems. The applications of machine learning mainly include Data Analysis and Mining, Pattern Recognition, Application in Bioinformatics, Machine Brain with Human Wisdom and Specific applications like Virtual assistant, Navigation Assistant and Filter spam and Malware.

Optimization is a fundamental aspect of machine learning. The first step in machine learning methods involves establishing a model and formulating a reasonable objective function. Once the objective function is determined, suitable numerical or analytical optimization methods are typically employed to solve the optimization problem.

In recent years, first-order stochastic optimization problems have attracted significant interest due to their advantages, such as high speed and low computational complexity. The primary motivation of this paper stems from the field of machine learning, with a specific emphasis on first-order stochastic optimization problems and their application in the context of step size.

Stochastic gradient descent (SGD), introduced by Robbins and Monro [1], finds extensive applications in the training of deep learning models [2], large-scale natural language processing [3], and matrix factorization [4]. Recently, many researchers have been exploring issues related to the selection of step sizes in existing algorithms, yielding several promising outcomes. Reference [5] integrated the Barzilai-Borwein (BB), proposed by Barzilai and Borwein [6], into the Mini-Batch Semi-Stochastic Gradient Descent (mS2GD) method [7], resulting in a new mini-batch method, mS2GD-BB. Reference [8] utilized the BB method to automatically calculate the step sizes for SGD and its variant, Stochastic Variance Reduced Gradient (SVRG), resulting in the creation of two algorithms: SGD- BB and SVRG-BB. Reference [9] introduced a variation of the adaptive step size strategy, known as the Stabilized Barzilai-Borwein (SBB) step size. This variation involves adding a positive term to the original BB step size's absolute denominator, aiming to address the instability issue of the BB step size. Meanwhile, the stabilized BB method was introduced in [10], which defined a boundary for the distance between each pair of successive iterations. This boundary enables a reduction in the number of BB iterations. A new variant of the BB method, Positive Defined Stabilized Barzilai-Borwein (PDSBB), was proposed to be incorporated into SVRG, resulting in the creation of a new algorithm, named SVRG-PDSBB [11].

$$\min_{w\in R^d} F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) \tag{1}$$

In equation (1), w denotes the model parameter, n represents the sample size , and $f_i(w)$ is a sequence of loss functions that assess the cost of the current parameter w. In this case, each $f_i : R^d \to R$ signifies the cost function corresponding to the $i-th$ sample data. Usually, $f_i(w)$ depends on training data (a_i, b_i) (supervised learning). In machine learning, to avoid overfitting, a

Manuscript received February 17, 2024; revised December 11, 2024.

This work was supported in part by Scientific Research Project of the Hunan Provincial Department of Education in 2024 under Grant 24B0819.

Weijuan Shi is a lecturer at College of Mathematics and Finance, Hunan University of Humanities, Science and Technology Loudi, 417000, China. (e-mail: shiweijuan2007@sina.com).

Adibah Shuib is an Associate Professor at School of Mathematical Sciences. College of Computing, Informatics and Media (KPPIM), Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia. (corresponding author to provide phone: +60192622348; fax: +603-55435501; e-mail: adibah253@uitm.edu.my).

Zuraida Alwadood is a senior lecturer at School of Mathematical Sciences. College of Computing, Informatics and Media (KPPIM), Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia. (e-mail: zuraida794@uitm.edu.my).

regularization term is usually added to objective function (1).

II. Gap Analysis

The SGD method offers significant theoretical and empirical advantages in machine learning [12]-[13], compressed sensing [14], wireless sensor networks [15], matrix factorization [4], [16], and large-scale natural language processing [3]. The SGD algorithm features frequent updating of the model for each training, which leads to faster learning, and compromises between fast computation per iteration and slow convergence.

Reference [8] stated that it is inappropriate and time-consuming to choose a constant or decreasing step size manually. Therefore, the BB method is utilized to calculate the step size for both SGD and SVRG. This method does not require any parameters and it dynamically computes the step size. However, this method cannot avoid situations where the denominator might be close to zero. Adding a positive term to the absolute value of the denominator of the original BB step size is a variant of the adaptive step size strategy proposed in [9] to overcome instability issue associated with the BB step size. This method does not need singular value decomposition since it drops convexity, making use of stochastic variance reduced gradient. However, this method does not provide an appropriate value or rule of the parameter.

There are also many algorithms that improve the BB step size. Reference [17] merged the online step size (OSS) into the mini-batch nonconvex stochastic variance reduced gradient (MSVRG) approach, resulting in a newly devised method known as MSVRG-OSS. It was shown that MSVRG-OSS possesses a linear convergence rate. Applying the OSS step size to MSVRG algorithm, when the sample size is relatively small, the numerical performance is sensitive to the choice of the initial step size and mini-batch size. However, with a larger sample size, it inevitably increases the computational workload and reduces the convergence speed. Reference [18] has introduced a variant of the Barzilai-Borwein (BB) method, known as the Random Barzilai-Borwein (RBB) method, to determine the step size for mini-batch nonconvex stochastic variance reduced gradient (SARAH) in the mini-batch setting. It has been demonstrated that the newly developed MB-SARAH-RBB converges linearly in expectation for strongly convex objective functions. Table I summarizes the characteristics of each algorithm.

TABLE I
Gap Analysis on the Variants of BB Step Size

Algorithm	Author	Step Size	Denominator Approaching Zero	Sensitivity to Initial Step Size	Sample Size Requirement
MSVRG-OSS	Yang et al.	OSS	No	Yes	Relatively high
mS2GD-RBB	Yang et al.	RBB	Yes	No	Minimal
SVRG-SBB	Ma et al.	SBB	No	No	Minimal
MB-SARAH-RBB	Yang et al.	RBB	Yes	No	Minimal

Meanwhile, the Stabilized Barzilai-Borwein (SBB) was also introduced in [10]. The SBB method achieves global convergence without the need for any line search, which has inspired the current study to modify the SBB method by introducing a dynamic adaptive step size. This aims to stimulate the development of more efficient algorithms. Adaptive step size methods, such as ADAptive Moment estimation (ADAM) [19] and AdaGrad [20], are commonly used to handle noisy gradients in optimization by dynamically adjusting the step size.

III. Barzilai-Borwein (BB) method and Positive Defined Stabilized Barzilai -Borwein (PDSBB) Method

The BB method, inspired by Newton’s method, is a gradient technique with modified step sizes and is often paired with a non-monotone line search. As described in [6], it employs a two-point step size for the steepest descent method by approximating the secant equation. Nowadays, the BB gradient method is recognized as an effective method for addressing large-scale unconstrained problems with modest accuracy and can be readily adapted for various constrained optimization problems.

Suppose model (2) is an unconstrained optimization problem needs to be addressed,

$$\min f\left(w\right) \tag{2}$$

where $f\left(w\right)$ is differentiable. The iterative equation for the quasi-Newton method applied to optimization problem (2) is as follows:

$$w_{t+1} = w_t - B_t^{-1}\nabla f\left(w_t\right) \tag{3}$$

where B_t is an approximation of the Hessian matrix of f at w_t .

The Hessian matrix ($\eta_t > 0$) is approximated by using $B_t = \frac{1}{\eta_t} I$, and it is substituted into the secant equation $B_t s_t = y_t$,where $y_t = \nabla f\left(w_t\right) - \nabla f\left(w_{t-1}\right)$, $s_t = w_t - w_{t-1}$, $t > 1$. By solving the residual of the secant equation, that is,

$$min\left(\frac{1}{\eta_t s_t} - y_t\right)^2 \tag{4}$$

The BB step size can be determined by

$$\eta_t^{BB1} = \left\|s_t\right\|^2 / \left(s_t^T y_t\right) \tag{5}$$

The second form of BB step size,

$$\eta_t^{BB2} = \left(s_t^T y_t\right) / \left\|y_t\right\|^2 \tag{6}$$

is obtained by solving

$$min\left\|s_t - \eta_t y_t\right\|^2 \tag{7}$$

Generally speaking, Equation (5) often performs numerically better than Equation (6) in practice. Therefore, in this study, Equation (5) serves as the main starting point, and the variant step size is based on Equation (5).

A new dynamic adaptive step size, named Positive Defined Stabilized Barzilai-Borwein (PDSBB) [11], has been developed based on modifications to the BB method to automatically calculate step sizes. The aim of the PDSBB is to address the issue that arises when the denominator of the BB step size approaches zero. The detailed description is as follows:

The first step is to calculate $\eta_t = \frac{\left\|s_t\right\|^2}{s_t^T y_t}$, where $y_t = \nabla f\left(w_t\right) - \nabla f\left(w_{t-1}\right)$ and $s_t = w_t - w_{t-1},\ t > 1$.

Secondly, if the denominator is close to 0, compare $s_t^T y_t$ to the given positive parameter ε, if $s_t^T y_t \le \varepsilon$, and set $\eta_t = \frac{1}{t}\sum_{i=0}^{t-1}\eta_i$.

In summary, that is,

$$\eta_t = \begin{cases} \frac{\left\|s_t\right\|^2}{s_t^T y_t}, s_t^T y_t > \varepsilon \\ \frac{1}{t}\sum_{i=0}^{t-1}\eta_i, s_t^T y_t \le \varepsilon \end{cases} \tag{8}$$

IV. Stochastic Gradient Descent with Positive Defined Stabilized Barzilai-Borwein (SGD-PDSBB) Method

The proposed PDSBB method was first integrated with SGD. Since SGD does not compute the full gradient $\nabla F\left(w\right)$, the PDSBB method has not been applied directly to SGD. In SGD, when uniform sampling is adopted, the stochastic gradient $\nabla f_{i_t}\left(w_t\right)$ is an unbiased estimation for $\nabla F\left(w_t\right)$. For further references on the importance of sampling and sampling procedures, the studies done by [21] and [22] can be referred to. Note that these studies did not utilize uniform sampling. Therefore, when computing the PDSBB step size using Equation (8), the estimation of $\nabla f_{i_{t+1}}\left(w_{t+1}\right) - \nabla f_{i_t}\left(w_t\right)$ was done using the expression $\nabla F\left(w_{t+1}\right) - \nabla F\left(w_t\right)$. However, due to the variance in stochastic gradient estimates, this method does not perform well. The study conducted in [23] introduced several variations for estimating a BB step size using stochastic gradients. However, these methods lack of theoretical justification, and numerical results have indicated that they are inferior to existing approaches such as averaged SGD [24]. The pseudo-code for the new SGD-PDSBB algorithm is described in Table II.

A few remarks about SGD-PDSBB are noted below:

1) The average of the stochastic gradients within one epoch for SGD-PDSBB is used as an estimation of the full gradient.
2) Every m iterations of SGD are referred to as an epoch.
3) The step size generated by $\left\|\tilde{w}_k - \tilde{w}_{k-1}\right\|_2^2 / \left(\tilde{w}_k - \tilde{w}_{k-1}\right)^T\left(g_k - g_{k-1}\right)$ may be close to zero or even negative. So, the following restriction is set.
$$\gamma_k = \frac{1}{m}\left\|\tilde{w}_k - \tilde{w}_{k-1}\right\|_2^2 / \left|\left(\tilde{w}_k - \tilde{w}_{k-1}\right)^T\left(g_k - g_{k-1}\right)\right|$$

the positive parameter ε should not be too small. If the denominator of γ_k is smaller than ε, choose $\eta_k = \frac{1}{k}\sum_{i=0}^{k-1}\eta_i$ to ensure that the step size remains within a reasonable range.

4) To make sure the average of stochastic gradients g_k close to $\nabla F\left(\tilde{w}_k\right)$, use
$$g_{k+1} = \beta\nabla f_{i_t}\left(w_k\right) + \left(1-\beta\right)g_{k+1} \tag{9}$$

to update g_{k+1} recursively, starting from $g_{k+1} = 0$, where $\beta \in \left(0,1\right)$ is a weighting parameter.

SGD-PDSBB requires the average of stochastic gradients from two epochs to compute the PDSBB step size. Consequently, the step sizes for the first two epochs, η_0 and η_1, need to be determined. Numerical experiments indicate that the performance of SGD-PDSBB is not significantly affected by the selection of these initial step sizes.

TABLE II
THE PSEUDO-CODE FOR THE SGD-PDSBB ALGORITHM

```
Algorithm: SGD with PDSBB step size (SGD-PDSBB)

Parameters: update frequency $m$, step size $\eta_0$ and $\eta_1$, initial point
$\tilde{w}_0$, weighting parameter $\beta \in \left(0,1\right)$, a small positive $\varepsilon$
For $k = 0,1,\cdots$ do
    if $k > 0$, then
        $\gamma_k = \frac{1}{m}\left\|\tilde{w}_k - \tilde{w}_{k-1}\right\|_2^2 / \left|\left(\tilde{w}_k - \tilde{w}_{k-1}\right)^T\left(g_k - g_{k-1}\right)\right|$
        if $\left|\left(\tilde{w}_k - \tilde{w}_{k-1}\right)^T\left(g_k - g_{k-1}\right)\right| > \varepsilon$
            $\eta_k = \gamma_k$
        else
            $\eta_k = \frac{1}{k}\sum_{i=0}^{k-1}\eta_i$
        end if
    end if
    $w_0 = \tilde{w}_k$
    $g_{k+1} = 0$
    for $t = 0,1,\cdots,m-1$ do
        Randomly pick $i_t \in \left\{1,\cdots,n\right\}$
        $w_{t+1} = w_t - \eta_k\nabla f_{i_t}\left(w_t\right)$
        $g_{k+1} = \beta\nabla f_{i_t}\left(w_t\right) + \left(1-\beta\right)g_{k+1}$
    end for
    $\tilde{w}_{k+1} = w_m$
end for
```

A. *Convergence Analysis for SGD-PDSBB Algorithm*

In the SGD-PDSBB algorithm, two important assumptions are made:

Assumption 1: Equation (10) holds for any w_t, the objective function $F\left(w\right)$ is $\mu-$ strongly convex, which means,

$$F\left(v\right) \ge F\left(w\right) + \nabla F\left(w\right)^T\left(v-w\right) + \frac{\mu}{2}\left\|w-v\right\|_2^2, \forall w,v \in R^d.$$

Assumption 2: The gradient of $f_i\left(w\right)$ is $L-$ Lipschitz continuous, that is,

$$\left\|\nabla f_i\left(w\right) - \nabla f_i\left(v\right)\right\|_2 \le L\left\|w-v\right\|_2, \forall w,v \in R^d.$$

It follows that $\nabla F\left(w\right)$ is also L-Lipschitz continuous:

$$\left\|\nabla F(w)-\nabla F(v)\right\|_2 \le L\left\|w-v\right\|_2,\ \forall w,v\in R^d.$$

In this study, ∇f_{i_t} is often assumed to be an unbiased estimate of ∇F, that is;

$$E\left[\nabla f_{i_t}(w_t)\middle| w_t\right]=\nabla F(w_t) \tag{10}$$

There are several important parameters of the algorithm, namely epoch k, initial step size η_0, positive parameter ε, and regularization parameter λ. The convergence performance was measured based on step size, sub-optimality, and accuracy.

Following a similar approach to [8], the convergence of the proposed SGD-PDSBB algorithm is subsequently analyzed. The distance between w_{k+1} to w^* is bounded.

$$E\left\|w_{t+1}-w^*\right\|_2^2=E\left\|w_t-\eta_k\nabla f_{i_t}(w_t)-w^*\right\|_2^2$$

$$=E\left\|w_t-w^*\right\|_2^2-2\eta_k E\left[\left(w_t-w^*\right)^T\nabla f_{i_t}(w_t)\right]$$

$$+\eta_k{}^2 E\left\|\nabla f_{i_t}(w_t)\right\|_2^2$$

$$=E\left\|w_t-w^*\right\|_2^2-2\eta_k\left(w_t-w^*\right)^T\nabla F(w_t)+\eta_k{}^2E\left\|\nabla f_{i_t}(w_t)\right\|_2^2$$

$$\left(E\left[\nabla f_{i_t}(w_t)\right]=\nabla F(w_t)\right)$$

$$\le\left(1-2\eta_k\mu\right)E\left\|w_t-w^*\right\|_2^2+\eta_k{}^2E\left\|\nabla f_{i_t}(w_t)\right\|_2^2$$

$$\text{(Strong convexity of } F(x)\text{)}$$

$$=\left(1-2\eta_k\mu\right)E\left\|w_t-w^*\right\|_2^2$$

$$+\eta_k{}^2E\left\|\nabla f_{i_t}(w_t)-\nabla F(w_t)+\nabla F(w_t)\right\|_2^2$$

$$\le\left(1-2\eta_k\mu+\eta_k{}^2L^2\right)E\left\|w_t-w^*\right\|_2^2+\eta_k{}^2E\left\|\nabla f_{i_t}(w_t)-\nabla F(w_t)\right\|_2^2$$

That is,

$$E\left\|w_{t+1}-w^*\right\|_2^2\le\underbrace{\left(1-2\eta_k\mu+\eta_k{}^2L^2\right)E\left\|w_t-w^*\right\|_2^2}_{A}$$
$$+\underbrace{\eta_k{}^2E\nabla f_{i_t}(w_t)-\nabla F(w_t)_2^2}_{B} \tag{11}$$

In Inequality (11), part A represents the expected multiple of the distance from w_t to w^*. Meanwhile, part B stands for a multiple of the variance in the gradient estimation. The next step demonstrates how part B affects the convergence rate. Reducing part B is a common method to accelerate convergence.

By choosing $\eta_k=\frac{1}{m}\frac{s_k{}^2}{s_k^Ty_k}$ or $\eta_k=\frac{1}{m}\frac{1}{k}\sum_{i=0}^{k-1}\eta_i$, and using the strong convexity of $F(w)$, the derivation of the upper bound for the PDSBB step size in SGD-PDSBB method is given by,

$$\eta_k=\frac{1}{m}\cdot\frac{\left\|\tilde{w}_k-\tilde{w}_{k-1}\right\|^2}{\left(\tilde{w}_k-\tilde{w}_{k-1}\right)^T\left(g_k-g_{k-1}\right)}$$

$$\le\frac{1}{m}\cdot\frac{\left\|\tilde{w}_k-\tilde{w}_{k-1}\right\|^2}{\left\|\mu\tilde{w}_k-\tilde{w}_{k-1}\right\|^2}=\frac{1}{m\mu}$$

or

$$\eta_k=\frac{1}{m}\cdot\frac{1}{k}\sum_{i=0}^{k-1}\eta_i<\frac{1}{m}\cdot k\cdot\frac{1}{k\mu}=\frac{1}{m\mu}.$$

Thus, it can be concluded that the upper bound of PDSBB step size is $\frac{1}{m\mu}$. Similarly, using the $L-$ Lipschitz continuity of $\nabla F(x)$, it is known that $\eta_k>\frac{1}{mL}$, then, Inequality (11**)** can be presented as the following:

$$E\left\|w_{t+1}-w^*\right\|_2^2\le\underbrace{\left(1-\frac{2\mu}{mL}+\frac{L^2}{m^2\mu^2}\right)\left\|w_t-w^*\right\|_2^2}_{A}$$
$$+\underbrace{\frac{1}{m^2\mu^2}E\left\|\nabla f_{i_t}(w_t)-\nabla F(w_t)\right\|_2^2}_{B} \tag{12}$$

Note that item B in Inequality (12) is some kind of variance in gradient estimation, which can lead to a relatively slow convergence rate. However, in many machine learning applications, the actual convergence speed of the SGD-PDSBB algorithm may be somewhat faster. This is primarily because many applications do not require extremely high accuracy, and at the beginning, the variance is small, that is, $B\ll A$. As a result, an approximate Q-linear convergence rate can be observed. As the number of iteration steps increases, the variance gradually rises. Therefore, to obtain a relatively fast asymptotic convergence rate, many studies focus on reducing the variance term B to accelerate convergence. In practice, using a BB-type step size often decreases the number of iterations required. In the next section, the performance of the newly introduced SGD-PDSBB algorithm is presented and discussed.

B. Numerical Experiments for SGD-PDSBB Algorithm

The SGD-PDSBB is applied to address two standard testing problems, referred to as Support Vector Machine (SVM) and Logistic Regression (LR) in the context of machine learning. Subsequently, the new algorithm is evaluated using several standard real-world datasets, including real-sim, a9a, w8a, ijcnn1, and covtype.binary, all of which are obtained from the LIBSVM website.

MATLAB software was utilized to normalize the data, aiming to enhance the model's performance and stability. Data normalization involves proportionally scaling data to fit a specific range, typically $[0,1]$ or $[-1,1]$. This approach ensures that data from diverse features share similar scales, preventing certain features from exerting an excessively large influence on the model.

To evaluate the performance of the SGD-PDSBB algorithm on specific problems, a series of numerical experiments were conducted. These experiments are aimed to verify the effectiveness of the SGD-PDSBB algorithm across different datasets and initial step sizes. This section validates the proposed PDSBB method and compares it to the deterministic SGD algorithm and the stochastic optimization algorithm SGD-BB through numerical experiments. Specifically, the following Support Vector Machine (SVM) with l_2 -norm regularization model equation (13) was utilized in the experiments.

$$\min_{w \in R^d} P(w) = \frac{1}{n}\sum_{i=1}^{n}\left(\left[1 - b_i a_i^T w\right]_+\right)^2 + \frac{\lambda}{2}\|w\|_2^2 \qquad (13)$$

The Logistic Regression model with l_2 -norm regularization model, as follows:

$$\min_{w} F(w) = \frac{1}{n}\sum_{i=1}^{n}\log\left[1 + exp\left(-b_i a_i^T w\right)\right] + \frac{\lambda}{2}\|w\|_2^2 \qquad (14)$$

are chosen. The $(a_i, b_i) \in R^d \times \{+1, -1\}, i = 1, \cdots, n$ represents the feature vectors and corresponding feature labels. The symbol $\lambda > 0$ is the regularization parameter. Meanwhile, each f_i is convex and differentiable, and the function F, which is strongly convex, is assumed.

Table III provides the datasets details of the computational experiments related to the models presented in Equation (13) and Equation (14). For the numerical experiments of this part, $m = n$, $\beta = 0.8$, $\varepsilon = 10^{-5}$, and $\eta_0 = \eta_1$, have been pre-specified.

In the following subsections, the results of the algorithm in solving the SVM with l_2 -norm regularization model, presented in Equation (13), and the Logistic Regression model with l_2 -norm regularization model, presented in Equation (14), on various datasets with different initial step sizes are primarily presented. The performance of the SGD-PDSBB algorithm was evaluated based on three measures: step size, sub-optimality, and classification accuracy.

TABLE III
DATA AND MODEL INFORMATION OF THE EXPERIMENTS ON SGD-PDSBB

Datasets	n	d	λ	Solving Model
real-sim	72,309	20958	10^{-4}	(13) and (14)
a9a	32,561	123	10^{-4}	(13) and (14)
covtype.binary	581,012	54	10^{-4}	(13)
w8a	49,749	300	10^{-5}	(14)
ijcnn1	49,990	22	10^{-4}	(13) and (14)

Note: n represents number of samples and d represents data dimension

C. Performance of SGD-PDSBB in solving model (13)

i) Step Size

The initial step size setting varies slightly across different datasets. For the real-sim, covtype.binary, and ijcnn1 datasets, three different initial step sizes, $\eta_0 = \eta_1 = 0.1$, 1, and 10, were selected as shown in Figure 1 (a)-(d). Meanwhile, for the a9a dataset, the selected initial step sizes were $\eta_0 = \eta_1 = 0.01$, 0.1, and 1. In Figure 1 (a) and Figure 1 (c) , the solid lines represent $\eta_0 = \eta_1 = 0.1$, the dashed lines represent $\eta_0 = \eta_1 = 1$, and the dotted lines represent $\eta_0 = \eta_1 = 10$. Aside from this, in the second sub-figures, the solid line represents $\eta_0 = \eta_1 = 0.1$, the dashed line represents $\eta_0 = \eta_1 = 1$, and the dotted line represents $\eta_0 = \eta_1 = 0.01$.

Figure 1 (a)-(d) illustrates the performance of the step size for SGD-PDSBB with different initial step sizes on the four selected datasets. In all four sub-figures, the x -axis corresponds to the number of epochs, representing the number of outer loops as stayed in Table II, and the y -axis stands for the step size.

In Figure 1 (a)-(d), there is little variation in the results across different initial step sizes. This suggests that the new SGD-PDSBB algorithm is insensitive to the initial step size.

The step size exhibits noticeably different behavior in the third sub-figure with the covtype.binary dataset compared to the other three datasets shown in Figure 1 (a)-(d). On the covtype.binary dataset, regardless of the initial step size, the step size converges to around 1.8 after several epochs. However, on the other three datasets, the step size appears more erratic, and demonstrated in subsequent comparison experiments with other algorithms, namely SGD and SGD-BB.

ii) Sub-optimality

Figure 2 (a)-(d) displays the sub-optimality of the SGD-PDSBB algorithm with different initial step sizes on four datasets: real-sim, a9a, covtype.binary, and ijcnn1. The x -axis represents the epoch, and the y -axis represents the sub-optimality $F(w_k) - F(w^*)$. The solid line represents the sub-optimality performance when $\eta_0 = \eta_1 = 0.1$ is used. The dashed line indicates the sub-optimality when $\eta_0 = \eta_1 = 1$ is used, and the dotted line denotes the sub-optimality when $\eta_0 = \eta_1 = 10$ is used.

In Figure 2 (a)-(d), it can be observed that the sub-optimality performance varies across different datasets. For the ijcnn1 dataset, the sub-optimality reaches 10^{-8}, for the covtype.binary dataset, it reaches 10^{-10}, while for the a9a dataset, it only reaches 10^{-2} . This suggests that the algorithm's performance is dataset dependent. However, it can be observed that the algorithm's sub-optimality performance on the same dataset does not differ significantly regardless of the initial step size chosen. This further indicates that the algorithm is insensitive to the choice of the initial step size.

D. Performance of SGD-PDSBB in solving model (14)

i) Step Size

Similar to the experiment on model (13), the performance of the algorithm is also evaluated from the aspects of step size and suboptimality. The step size results of SGD-PDSBB in solving model (14) in four datasets with different initial step sizes are shown in Figure 3 (a)-(d). In this analysis, the initial step sizes were consistently set to $\eta_0 = \eta_1 = 0.1, 1,$ and 10 on all four datasets.

Figure 3 (a)-(d) shows that the step size does not converge to a fixed step size. However, it always increases to a positive value when the step size is close to zero. This suggests that the algorithm introduced in this paper effectively prevents the step size from approaching zero by controlling the denominator of the step size. On these four datasets, the performance of the step size is almost the same and shows some oscillations. For the ijcnn1 dataset, the performance is relatively stable when both η_0 and η_1 are set to 1. This indicates that the algorithm performs slightly differently on different datasets.

ii) Sub-optimality

Figure 4 (a)-(d) shows the four-optimality results of the SGD-PDSBB algorithm in solving model (14) across four datasets with different initial step sizes. In Figure 4 (a)-(d), the initial step sizes are consistently set to $\eta_0 = \eta_1 = 0.1$, 1 and 10 for all four datasets.

From the four sub-figures in this analysis, it can be observed that the SGD-PDSBB algorithm exhibits slightly varying degrees of sub-optimality across different datasets. For example, for the ijcnn1 dataset, the sub-optimality can reach 10^{-8} after 30 epochs. In contrast, on other datasets, the sub-optimality only reaches 10^{-3}. Thus, the performance does not always achieve satisfactory precision.

The effectiveness of the SGD-PDSBB algorithm depends on the particular model and datasets being used. In general, the SGD-PDSBB algorithm has been demonstrated to perform effectively for tasks where extremely high precision is not a critical requirement. Consequently, its stability and convergence behavior are typically considered satisfactory and acceptable.

E. Comparison between SGD-PDSBB and both SGD and SGD-BB in solving model (13)

In this subsection, this study compared the SGD-PDSBB algorithm with other algorithms (SGD and SGD-BB) in terms of step size, sub-optimality, and classification accuracy when solving model (13).

i) Step size

The datasets used in this study are real-sim, a9a, covtype.binary, and ijcnn1, with small difference in initial step size settings for each datasets. Detailed information on the initial step sizes is provided in Table IV.

From the choice of initial step sizes, it can be observed that the initial step size for the SGD algorithm varies across different datasets. This variation is because during the numerical experiments, it was found that the algorithm SGD is sensitive to the initial step size, and different initial step size leads to different results. For example, on the real-sim and a9a datasets, an initial step size of 10 causes SGD to fail to converge and prevents it from computing the optimal value. Therefore, a relatively small initial step size was chosen for these two datasets. However, the SGD-BB and SGD-PDSBB algorithms are not sensitive to the choice of initial step size.

In (a), (b), (c) and (d) of Figure 5, constant step sizes are represented by dotted lines. The step sizes for the SGD-BB and SGD-PDSBB algorithms are represented by solid lines. It should be noted that the choice of the initial step size is consistent in both Figure 5 (a)-(d) and Figure 6 (a)-(d) .

From Figure 5 (c), it can be observed that on the covtype.binary dataset , both the SGD-BB and SGD-PDSBB algorithms converge to a step size of approximately 1.8. On the other three datasets, the step size did not converge to a specific value, which is consistent with the convergence analysis. However, Figure 5 (a), (b), and (d) also shows that on these three datasets, the step size of the SGD-BB algorithm dropped to 10^{-10} after 10 epochs. In contrast, the new SGD-PDSBB algorithm avoids this situation. Although the step size exhibits fluctuations, it generally stays within a relatively small range. As stated earlier, a step size that is too small can slow down the convergence of the algorithm and potentially lead to it getting stuck in a local optimum.

ii) Sub-optimality

Figure 6 (a)-(d) displays the sub-optimality results of the SGD, SGD-BB, and SGD-PDSBB algorithms on different datasets with varying initial step sizes when solving model (13). The results for the SGD algorithm are represented by dotted lines, the SGD-BB algorithm by dashed lines, and the SGD-PDSBB algorithm by solid lines. Different initial step sizes are consistent with the selection presented in Table IV. Specific details can be referred to in the legend of each sub-figure.

TABLE IV
INITIAL STEP SIZE IN ALGORITHM SGD, SGD-BB AND SGD-PDSBB

Datasets	SGD	SGD-BB	SGD-PDSBB
real-sim	0.01	0.1	0.1
	0.1	1	1
	1	10	10
covtype.binary	0.1	0.1	0.1
	1.8	1	1
	10	10	10
a9a	0.001	0.01	0.01
	0.01	0.1	0.1
	0.1	1	1
ijcnn1	0.1	0.1	0.1
	1	1	1
	10	10	10

Figure 6 (a)-(d) shows the sub-optimality results, indicating a significant difference in performance across datasets. The SGD-PDSBB algorithm achieves a sub-optimality of 10^{-10} on covtype.binary dataset, while its performance is the worst on the a9a dataset with a sub-optimality of only 10^{-2} . The performance on the a9a dataset is the most stable in terms of sub-optimality, whereas the SGD-PDSBB algorithm exhibits the strongest oscillations on the covtype.binary dataset. All four sub-figures demonstrate that the SGD-BB and SGD-PDSBB algorithms achieve similar sub-optimality to SGD with a fixed step size of 1.

As a whole, the performance of the SGD-PDSBB algorithm is better than that of the SGD and SGD-BB algorithms on the real-sim dataset. The performance of SGD-PDSBB algorithm is similar to SGD-BB and superior to SGD on the a9a and covtype.binary datasets. For the ijcnn1 dataset, SGD-PDSBB algorithm outperforms SGD in both stability and sub-optimality aspects, although its stability is not as good as that of SGD-BB.

iii) Classification Accuracy

Table V, Table VI, Table VII and Table VIII present the classification accuracy of SGD, SGD-BB, and SGD-PDSBB algorithms on different datasets with varying initial step sizes at the end of 35 epochs. The SGD algorithm utilizes a fixed step size of η , while the SGD-BB and SGD-PDSBB algorithms utilize the BB step size and the PDSBB step size with $\eta_0 = \eta_1$.

In general, the algorithms achieve reasonably good results on the real-sim and covtype.binary datasets which exceeds 77% and 60%, respectively. The classification accuracy of the SGD-BB algorithm for the a9a dataset remained constant at 0, indicating that the SGD-BB algorithm cannot perform classification on this dataset. Based on the observation of the experimental procedure, the step size of the SGD-BB algorithm becomes very small, reaching 10^{-10} , after approximately 10 steps. The program exits the loop, which indicates the failure of the SGD-BB algorithm, thus explaining why the accuracy is 0.0000. For all instances where the accuracy is 0.0000, repeated runs of the program show that after the SGD-BB algorithm fails, the accuracy is 0.0000. However, the SGD-PDSBB algorithm effectively solves this problem by providing a stable adaptive step size whenever the step size becomes extremely small. Therefore, the new SGD-PDSBB algorithm proposed in this study has shown promising accuracy which exceeds 84%. The classification accuracy of SGD, SGD-BB, and SGD-PDSBB algorithms on the ijcnn1 dataset is poor, with values either at 0 or below 10%. This is a relatively poor numerical result, and it is related to the inherent characteristics of the original SGD algorithm.

TABLE V
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (13) ON REAL-SIM DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.01		0.7513
SGD-BB		0.1	0.7795
SGD-PDSBB		0.1	0.7791
SGD	0.1		0.7791
SGD-BB		1	0.7789
SGD-PDSBB		1	0.7791
SGD	1		0.7700
SGD-BB		10	0.7787
SGD-PDSBB		10	0.7790

TABLE VI
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (13) ON A9A DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.001		0.8438
SGD-BB		0.01	0.0000
SGD-PDSBB		0.01	0.8460
SGD	0.01		0.7936
SGD-BB		0.1	0.0000
SGD-PDSBB		0.1	0.8424
SGD	0.1		0.8436
SGD-BB		1	0.0000
SGD-PDSBB		1	0.8444

TABLE VII
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (13) ON COVTYPE.BINARY DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.1		0.6044
SGD-BB		0.1	0.6044
SGD-PDSBB		0.1	0.6044
SGD	1.8		0.6044
SGD-BB		1	0.6044
SGD-PDSBB		1	0.6044
SGD	10		0.6044
SGD-BB		10	0.6044
SGD-PDSBB		10	0.6044

TABLE VIII
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (13) ON IJCNN1 DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.1		0.0960
SGD-BB		0.1	0.0000
SGD-PDSBB		0.1	0.0960
SGD	1		0.0960
SGD-BB		1	0.0000
SGD-PDSBB		1	0.0960
SGD	10		0.0960
SGD-BB		10	0.0960
SGD-PDSBB		10	0.0960

F. Comparison between SGD-PDSBB and both SGD and SGD-BB in solving model (14)

In this subsection, the SGD-PDSBB algorithm is compared to other algorithms (SGD, SGD-BB) in terms of step size, sub-optimality, and classification accuracy when solving model (14) on the datasets real-sim, a9a, w8a, and ijcnn1.

i) Step size

Figure 7 (a)-(d) presents a comparison of step sizes in the SGD-PDSBB, SGD-BB, and SGD algorithms. The figure includes four subplots, with the solid lines representing the SGD-PDSBB algorithm, dashed lines representing the SGD-BB algorithm, and dotted lines representing the SGD algorithm. For all four datasets, the initial step sizes of 0.1, 1, and 10 were consistently selected.

The plot indicates that the step size of the SGD-BB algorithm dropped close to zero shortly after the initial few epochs, while the SGD-PDSBB algorithm maintains a consistent step size through all epochs. Whenever the step size approaches zero, the configuration of this algorithm ensures that the step size is set to the average of the step sizes from the previous k epochs. This prevents the algorithm from becoming ineffective due to excessively small step sizes. The figure demonstrates that the SGD-PDSBB algorithm effectively prevents the issues of ineffectiveness that can arise with the SGD-BB algorithm due to too small step sizes.

ii) Sub-optimality

Figure 8 (a)-(d) presents a comparison of suboptimality performance between the SGD-PDSBB, SGD and SGD-BB algorithms on the real-sim, a9a, w8a, and ijcnn1 datasets using varying initial step sizes. The plots display the results of the SGD algorithm with dotted lines, the SGD-BB algorithm with dashed lines, and the SGD-PDSBB algorithm with solid lines.

On the a9a dataset, it is observed that after 30 epochs, the efficacy of the SGD-BB algorithm diminishes, while the SGD-PDSBB algorithm continues to perform well. This indicates that the SGD-PDSBB algorithm improves upon the SGD-BB algorithm. When the SGD-BB algorithm fails in certain specific cases, the SGD-PDSBB algorithm maintains a consistent level of suboptimal.

On the real-sim and ijcnn1 datasets, the SGD-PDSBB algorithm achieved comparable sub-optimality for the SGD algorithm with a step size of $\eta = 1$. Meanwhile, on the w8a dataset, the sub-optimality of the SGD-PDSBB algorithm exceeded both SGD and SGD-BB algorithms.

In summary, the sub-optimality performance of the SGD-PDSBB algorithm is similar to the SGD and SGD-BB algorithms on certain datasets, while it outperforms them on others.

iii) Classification Accuracy

Table IX, Table X, Table XI, and Table XII display the classification accuracy of the SGD, SGD-BB, and SGD-PDSBB algorithms on the real-sim, a9a, w8a, and ijcnn1 datasets. The SGD algorithm used fixed step sizes of 0.1, 1, and 10 during the runtime process, while $\eta_0 = \eta_1$ was implemented for the SGD-BB and SGD-PDSBB algorithms.

TABLE IX
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (14) ON REAL-SIM DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.1		0.7700
SGD-BB		0.1	0.7694
SGD-PDSBB		0.1	0.7702
SGD	1		0.7707
SGD-BB		1	0.7700
SGD-PDSBB		1	0.7706
SGD	10		0.7703
SGD-BB		10	0.7708
SGD-PDSBB		10	0.7703

TABLE X
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (14) ON A9A DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.1		0.8498
SGD-BB		0.1	0.0000
SGD-PDSBB		0.1	0.8474
SGD	1		0.8428
SGD-BB		1	0.0000
SGD-PDSBB		1	0.8491
SGD	10		0.7369
SGD-BB		10	0.0000
SGD-PDSBB		10	0.8488

TABLE XI
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (14) ON W8A DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.1		0.9920
SGD-BB		0.1	0.9908
SGD-PDSBB		0.1	0.9920
SGD	1		0.9911
SGD-BB		1	0.0677
SGD-PDSBB		1	0.9922
SGD	10		0.9806
SGD-BB		10	0.9919
SGD-PDSBB		10	0.9922

TABLE XII
CLASSIFICATION ACCURACY FOR SGD, SGD-BB AND SGD-PDSBB IN SOLVING MODEL (14) ON IJCNN1 DATASET

Algorithm	Step size η	Step size $\eta_0 = \eta_1$	Accuracy
SGD	0.1		0.0960
SGD-BB		0.1	0.0960
SGD-PDSBB		0.1	0.0960
SGD	1		0.0960
SGD-BB		1	0.0000
SGD-PDSBB		1	0.0960
SGD	10		0.0960
SGD-BB		10	0.0000
SGD-PDSBB		10	0.0960

It should be noted that accuracy varies across the different datasets. On the real-sim dataset, the three algorithms employ fixed or initial step sizes, achieving classification accuracy of approximately 77%. In the a9a dataset, the SGD-BB algorithm experiences a failure, resulting in a zero-accuracy score, while the SGD-PDSBB algorithm performs comparably or even superiorly to the SGD algorithm. The classification accuracy results of the three algorithms on the w8a dataset are relatively satisfactory with accuracy reaching about 99%. However, on the ijcnn1 dataset, the performance of all three algorithms is unsatisfactory, with accuracy less than 10%. The SGD-BB algorithm even completely fails with zero accuracy. This underscores the dependence of the numerical performance of SGD-type algorithms on the dataset. The SGD-PDSBB algorithm performs comparably better as compared to the SGD and SGD-BB algorithm.

This analysis introduces the improved BB step size, known as the PDSBB step size, combined with the SGD step size to create a new algorithm called SGD-PDSBB. A theoretical analysis of the convergence of this new algorithm is presented, along with numerical experiments to demonstrate its effectiveness. Furthermore, comparisons are made with the SGD and the SGD-BB algorithms in terms of sub-optimality, step size, and classification accuracy in solving model (13) and model (14). Based on the numerical experiments, the proposed SGD-PDSBB algorithm demonstrates effectiveness with comparable or better performance in step size, sub-optimality, and classification accuracy in contrast to the original SGD and SGD-BB algorithms. In fact, it also achieved a more desirable outcome even in the case when the SGD-BB algorithm fails.

V. Conclusion

This study integrates the PDSBB method with the original SGD algorithm, resulting in a new algorithm named SGD-PDSBB. This study provides an analysis of the convergence properties of the SGD-PDSBB algorithm. It was found that its convergence is particularly sensitive to the variance of the stochastic gradients.

Numerical experiments were conducted on optimization model (13) and (14) using five datasets, primarily demonstrating the algorithm's performance in terms of step size and sub-optimality. Regarding step size, the SGD-PDSBB algorithm is found to be insensitive to the choice of initial step size. Concerning sub-optimality, the SGD-PDSBB algorithm exhibits varying performances across different datasets.

Comparisons were made between the SGD-PDSBB, SGD and SGD-BB algorithms in terms of step size, sub-optimality, and classification accuracy, with different initial step sizes across various datasets. It is concluded that SGD-PDSBB is effective, insensitive to the choice of initial step size, and prevents excessively small step sizes. While the performance in terms of sub-optimality varies across datasets, SGD-PDSBB consistently outperforms the original SGD algorithm and achieves better or similar sub-optimality as compared to SGD-BB. In terms of classification accuracy, SGD, SGD-BB, and SGD-PDSBB show similar performance, with SGD-PDSBB outperforming SGD and SGD-BB on some datasets. Since SGD-BB shows zero accuracy in certain cases, suggest that SGD-PDSBB consistently maintains stable suboptimal performance.

This study integrates a stabilized BB step size called PDSBB into the existing SGD algorithm, addressing the drawbacks of the original algorithms that used fixed or decreasing step sizes. Improvements were also made to the BB step size to prevent the denominator from approaching zero, thereby enhancing stability. The new proposed algorithm which is named SGD-PDSBB improves the stability issues that could arise when original algorithms employ BB step sizes. In addition, the convergence properties of the new algorithm, and the reasons for the unstable convergence of SGD-type algorithms are also analyzed.

Data Availability Statement

The five actual standard data sets used in this study were obtained from the LIBSVM website, http://www.csie.ntu.edu.tw/~cjlin/libsvm

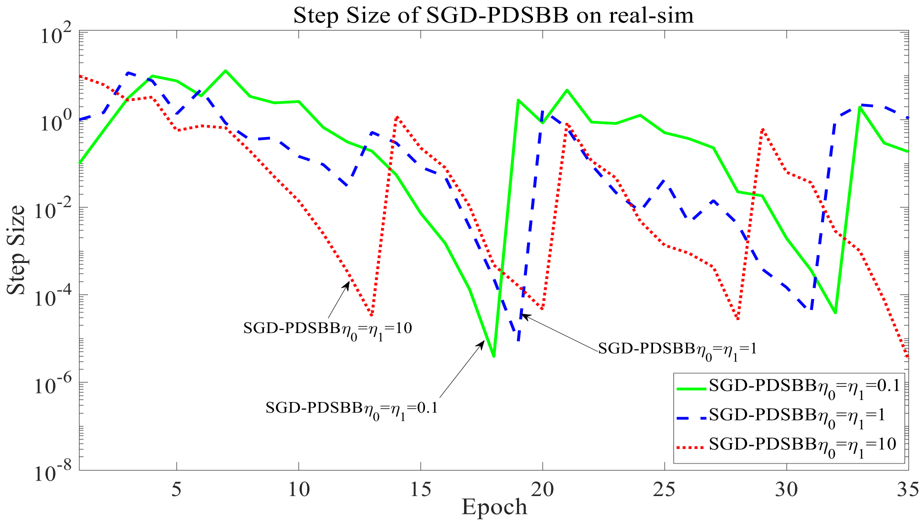

Fig. 1. Step Size Results of SGD-PDSBB for Model (13)
(a). Results on the real-sim Dataset with Different Initial Step Sizes

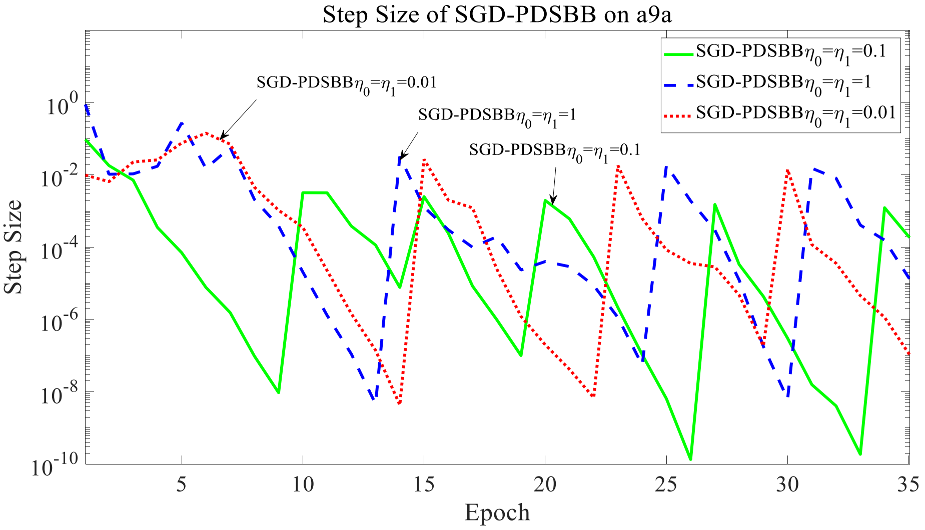

Fig. 1. Step Size Results of SGD-PDSBB for Model (13)
(b). Results on the a9a Dataset with Different Initial Step Sizes

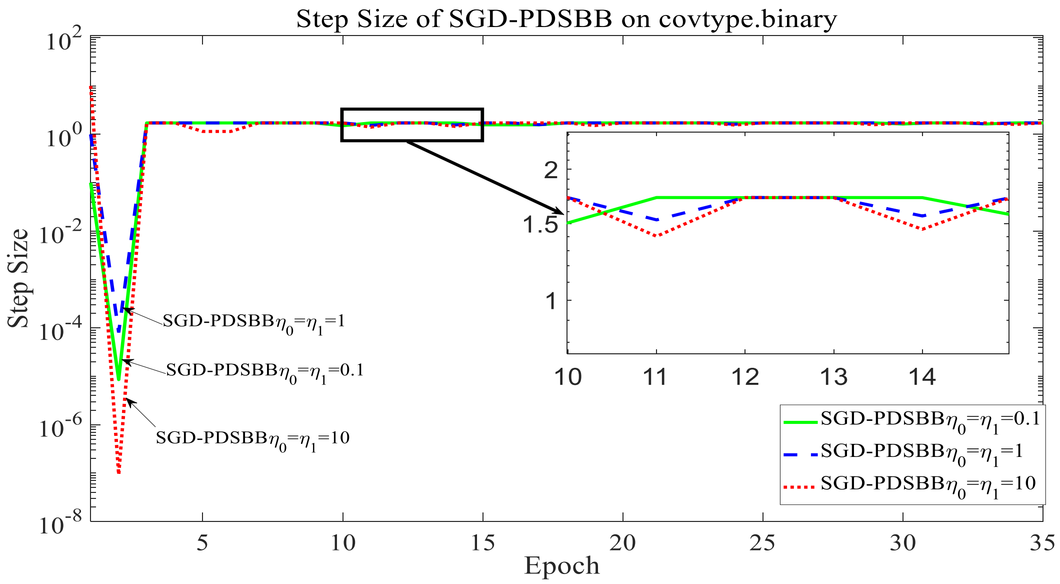

Fig. 1. Step Size Results of SGD-PDSBB for Model (13)
(c). Results on the covtype.binaty Dataset with Different Initial Step Sizes

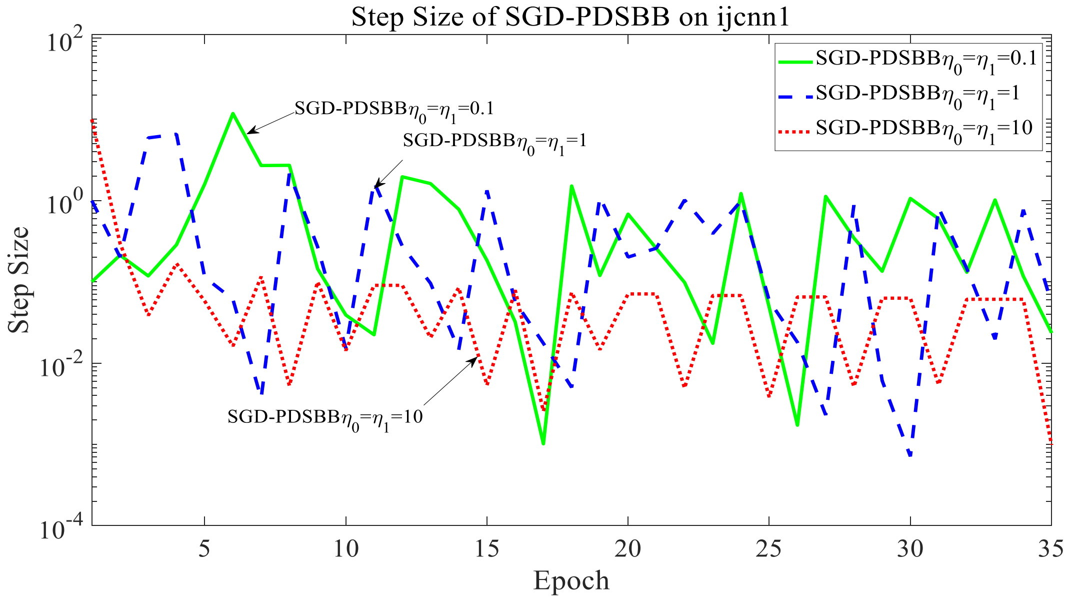

Fig. 1. Step Size Results of SGD-PDSBB for Model (13)
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

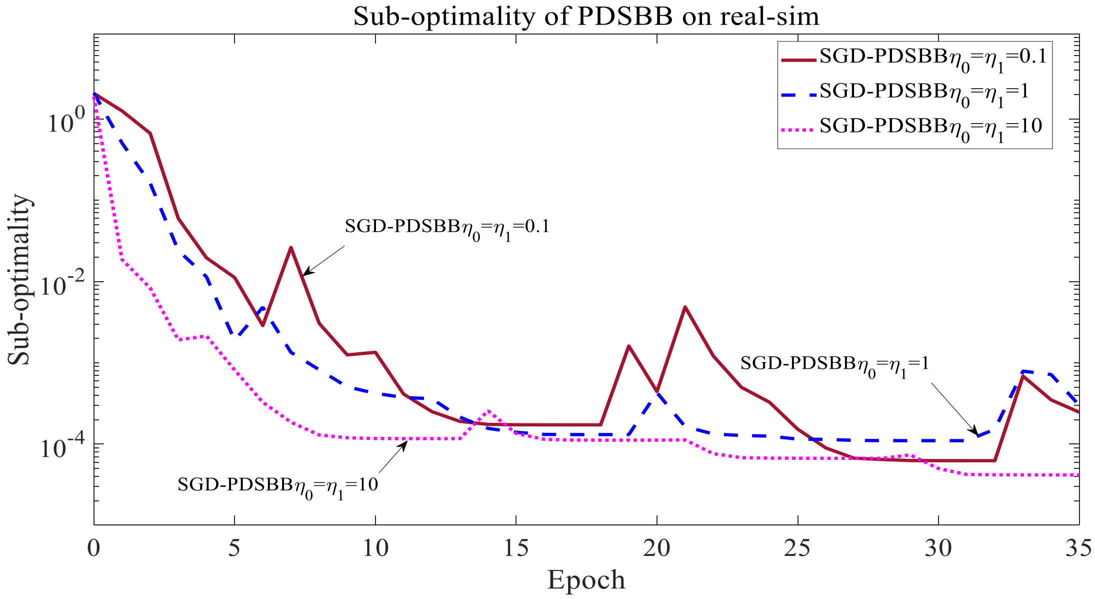

Fig. 2. Sub-Optimality Results of SGD-PDSBB for Model (13)
(a). Results on the real-sim Dataset with Different Initial Step Sizes

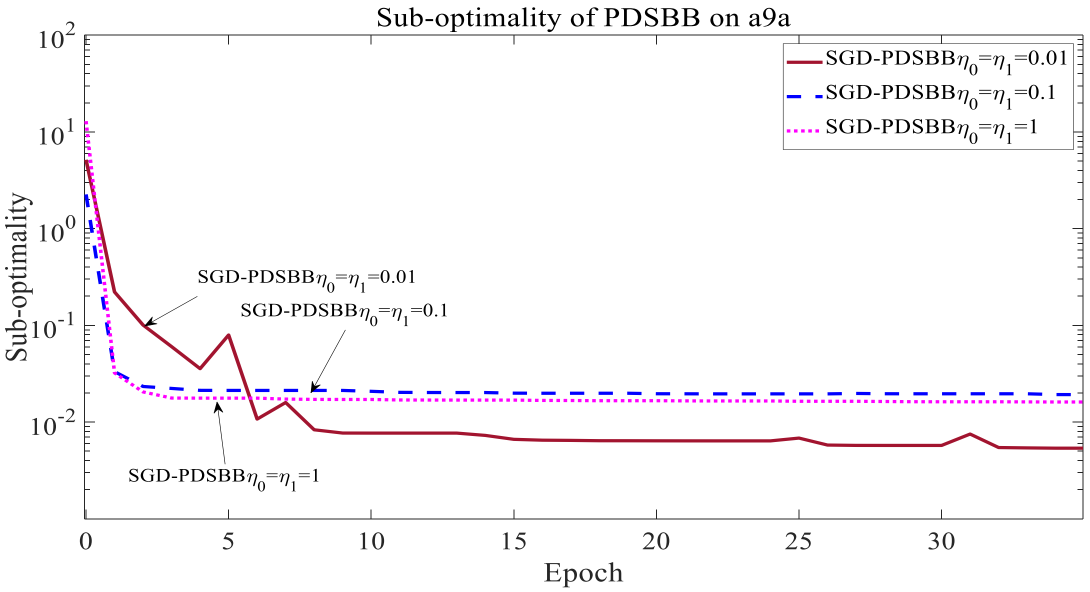

Fig. 2. Sub-Optimality Results of SGD-PDSBB for Model (13)
(b). Results on the a9a Dataset with Different Initial Step Sizes

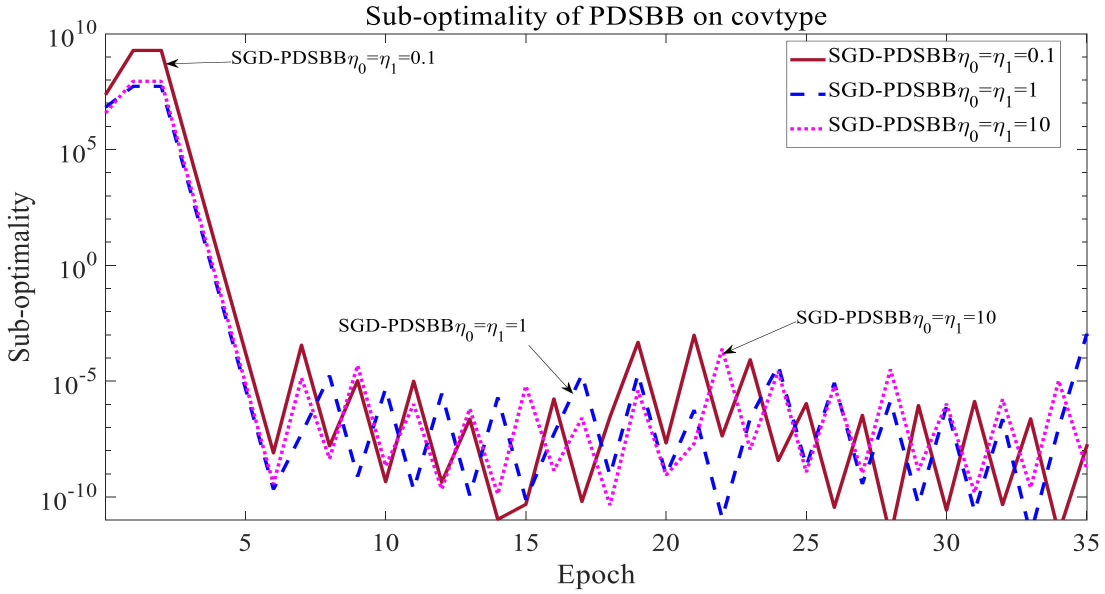

Fig. 2. Sub-Optimality Results of SGD-PDSBB for Model (13)
(c). Results on the covtype.binary Dataset with Different Initial Step Sizes

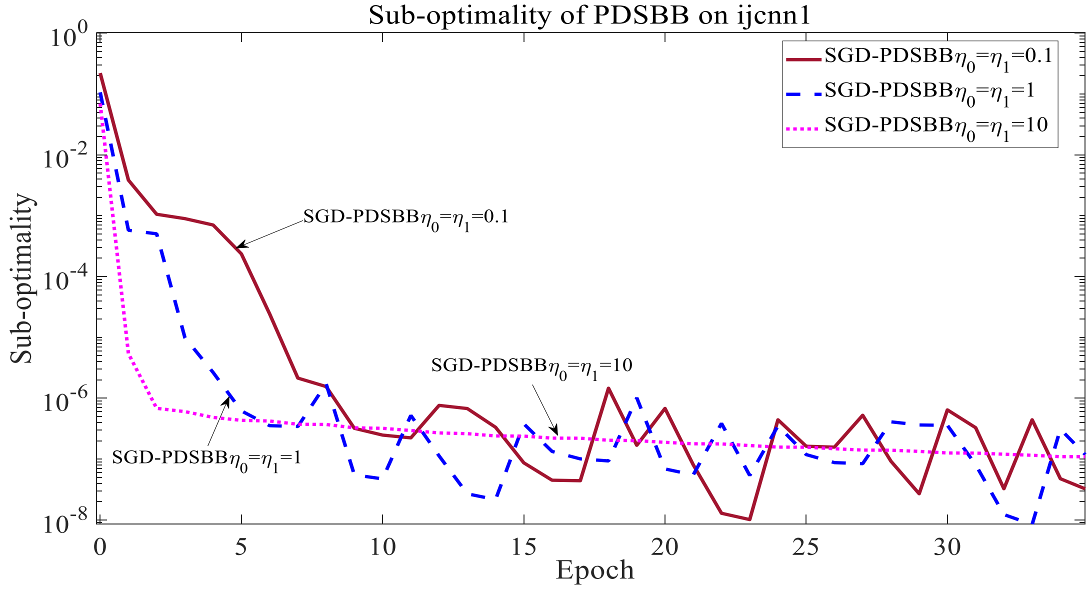

Fig. 2. Sub-Optimality Results of SGD-PDSBB for Model (13)
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

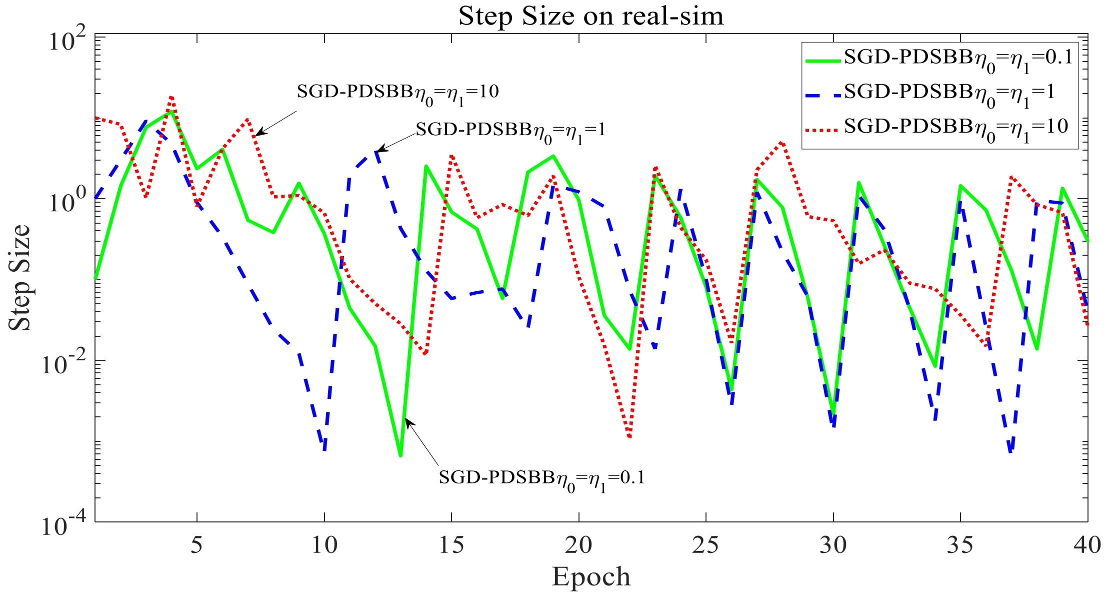

Fig. 3. Step Size Results of SGD-PDSBB for Model (14)
(a). Results on the real-sim Dataset with Different Initial Step Sizes

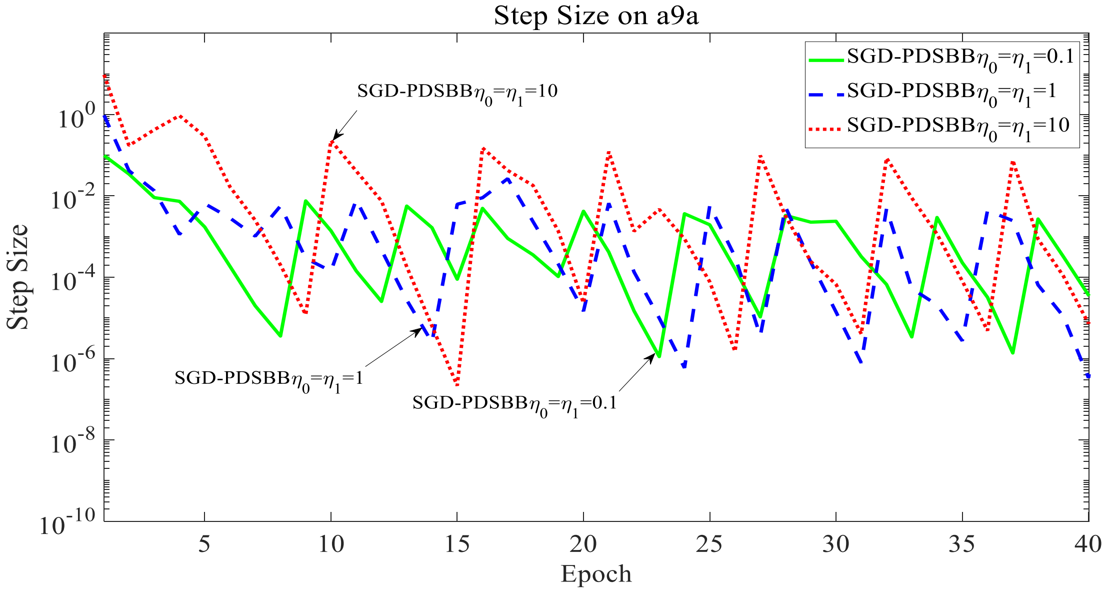

Fig. 3. Step Size Results of SGD-PDSBB for Model (14)
(b). Results on the a9a Dataset with Different Initial Step Sizes

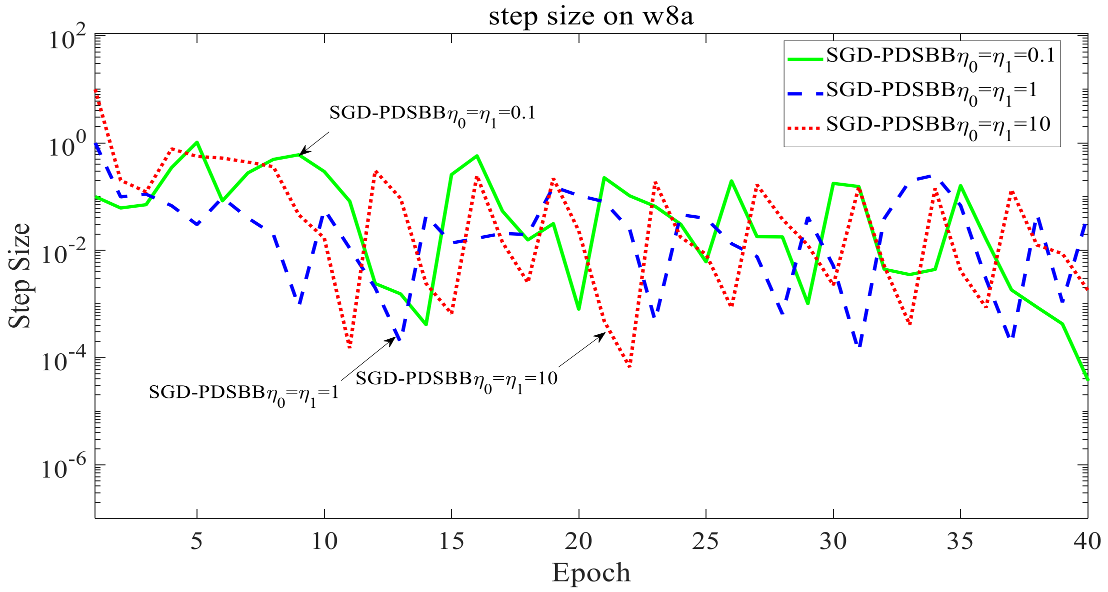

Fig. 3. Step Size Results of SGD-PDSBB for Model (14)
(c). Results on the w8a Dataset with Different Initial Step Sizes

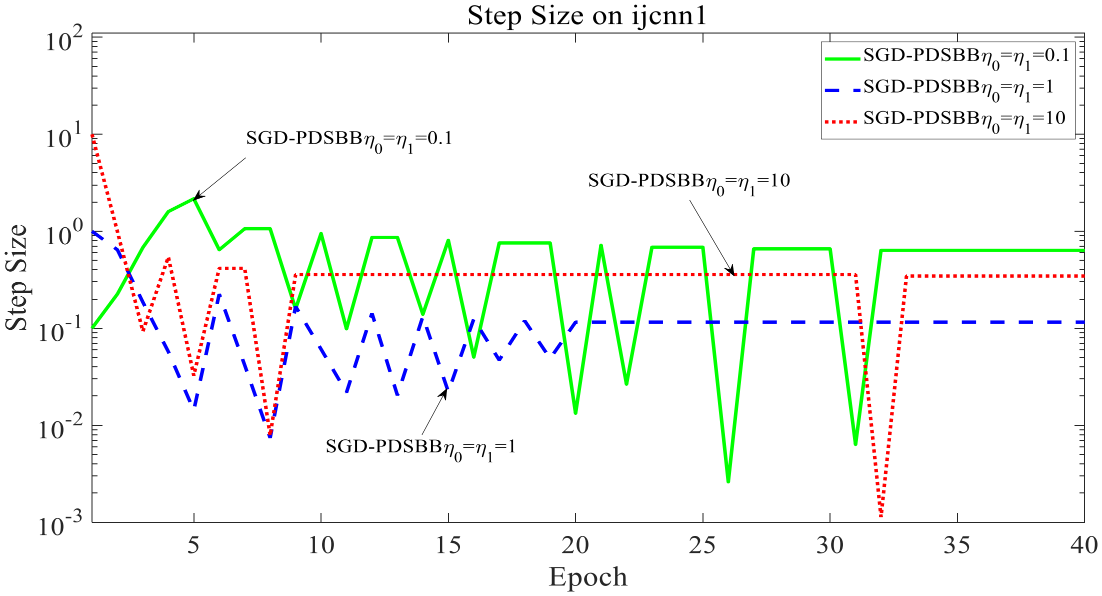

Fig. 3. Step Size Results of SGD-PDSBB for Model (14)
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

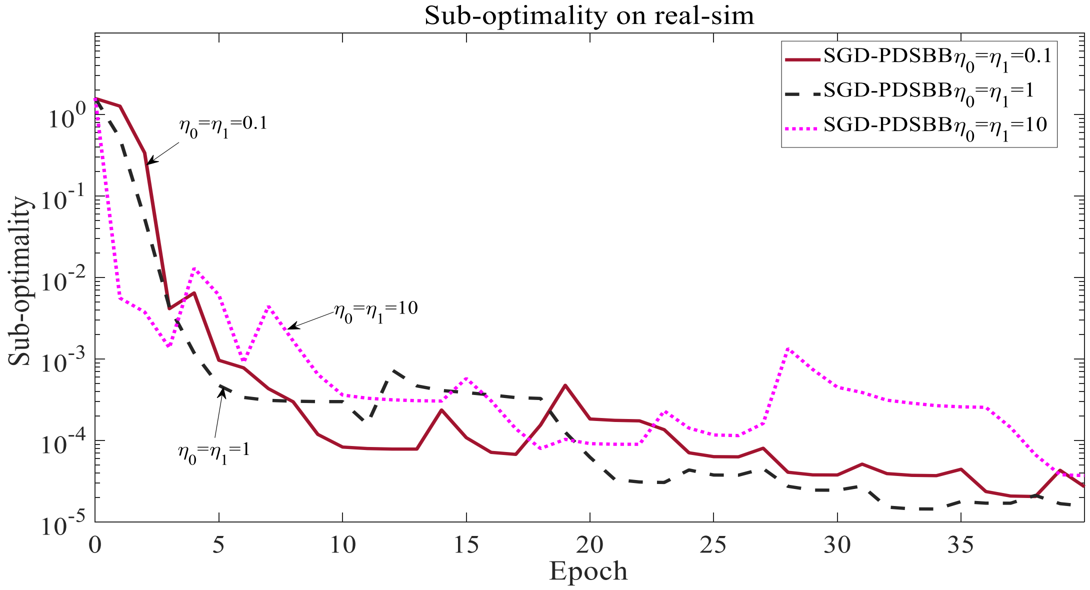

Fig. 4. Sub-Optimality Results of SGD-PDSBB for Model (14)
(a). Results on the real-sim Dataset with Different Initial Step Sizes

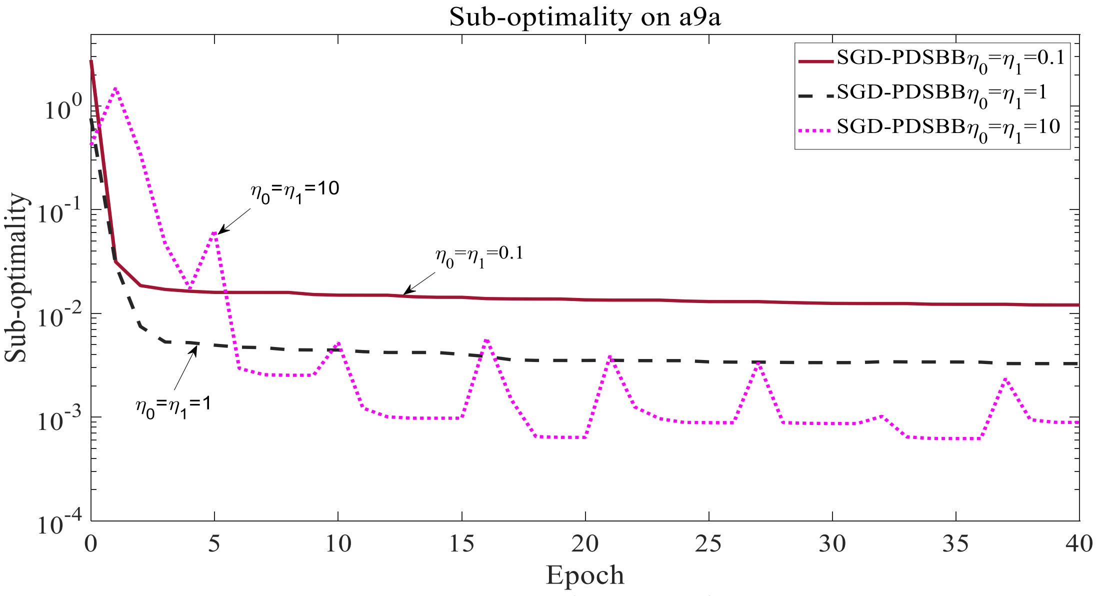

Fig. 4. Sub-Optimality Results of SGD-PDSBB for Model (14)
(b). Results on the a9a Dataset with Different Initial Step Sizes

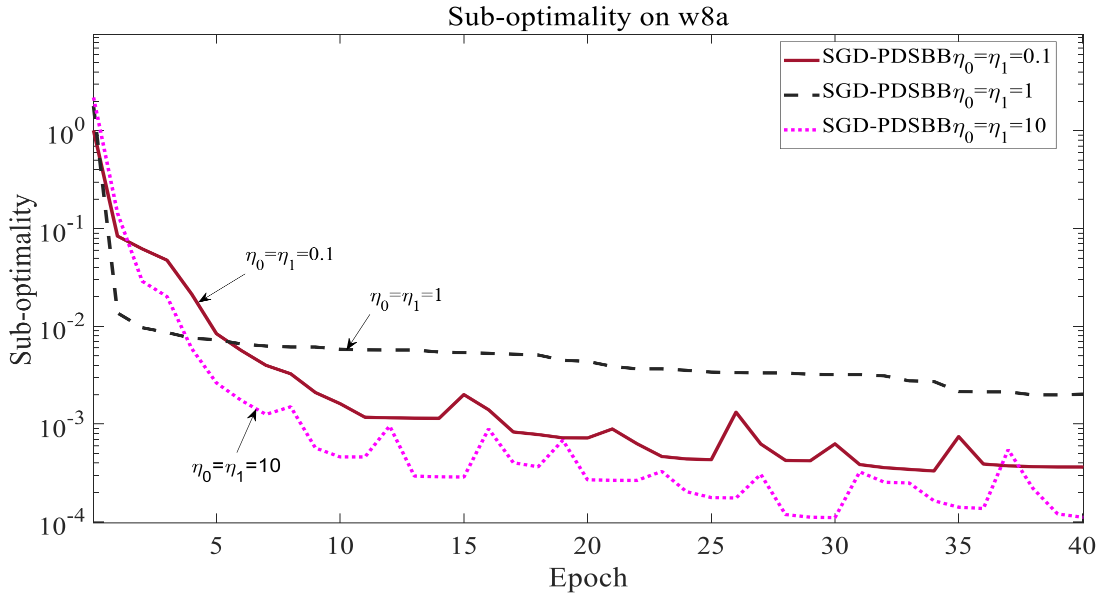

Fig. 4. Sub-Optimality Results of SGD-PDSBB for Model (14)
(c). Results on the w8a Dataset with Different Initial Step Sizes

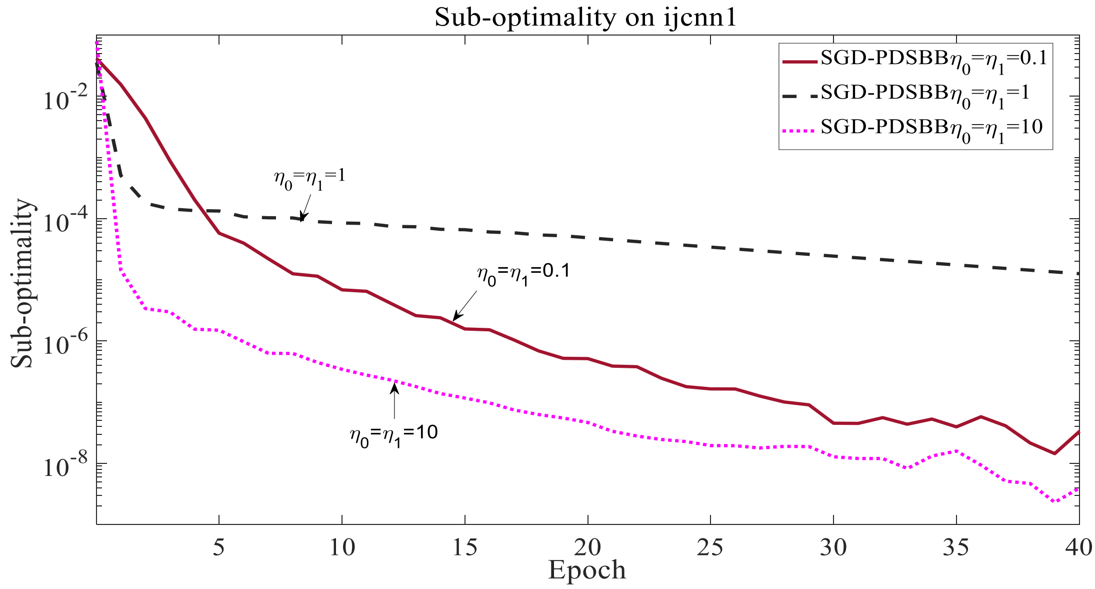

Fig. 4. Sub-Optimality Results of SGD-PDSBB for Model (14)
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

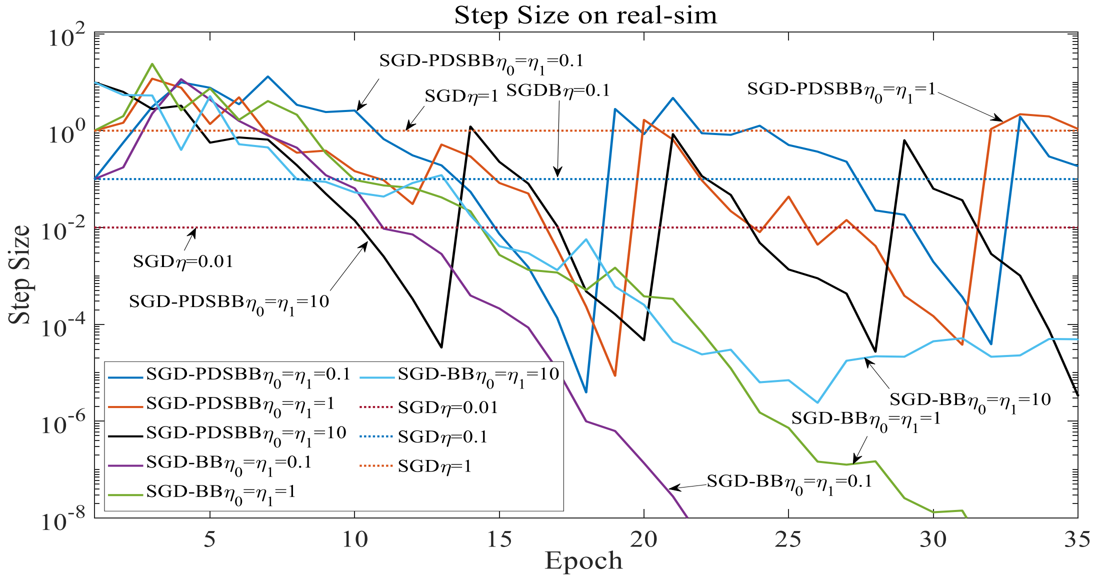

Fig. 5. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(a). Results on the real-sim Dataset with Different Initial Step Sizes

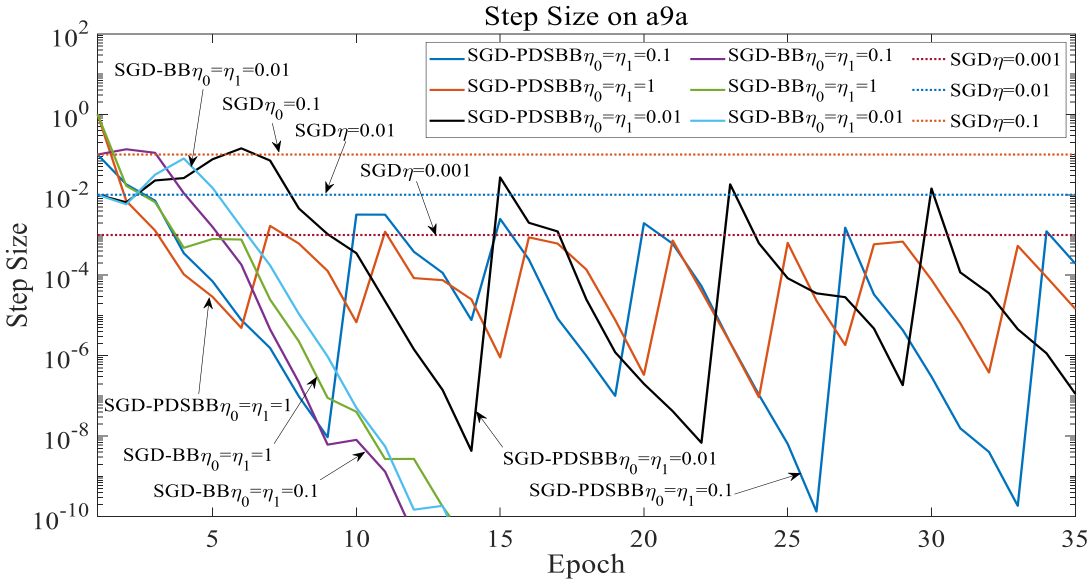

Fig. 5. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(b). Results on the a9a Dataset with Different Initial Step Sizes

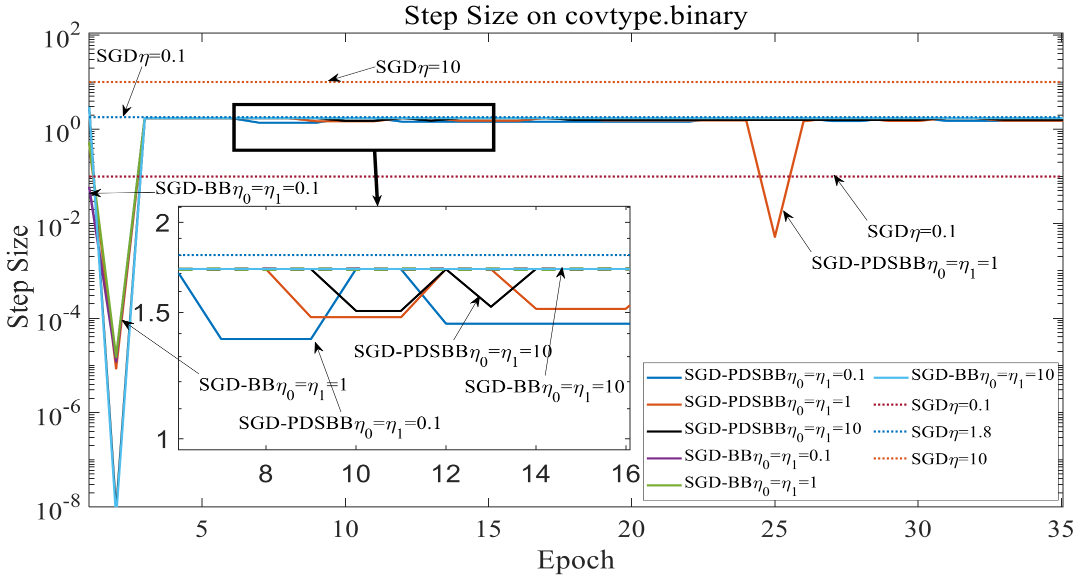

Fig. 5. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(c). Results on the covtype.binary Dataset with Different Initial Step Sizes

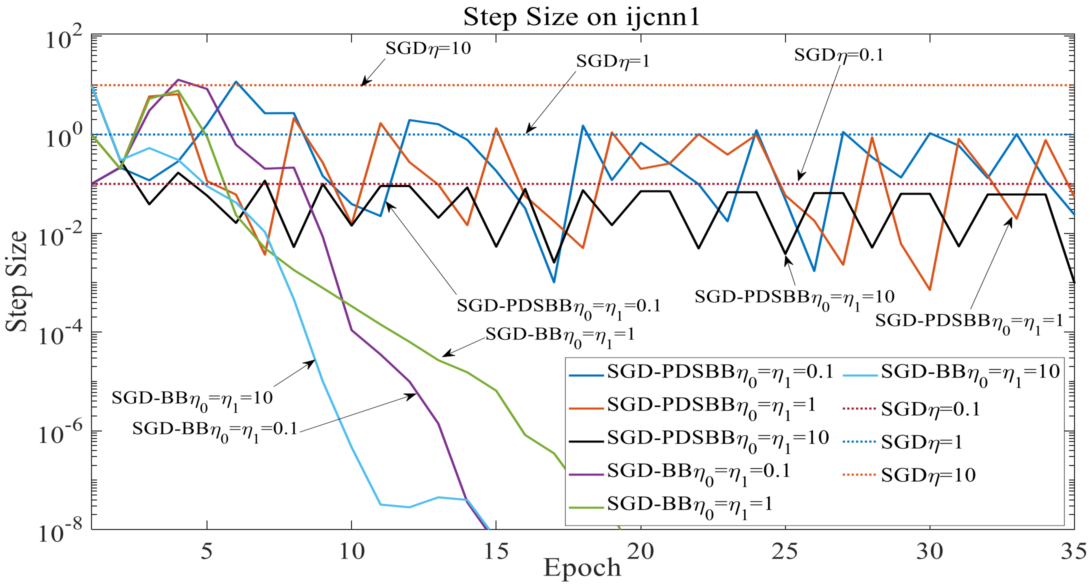

Fig. 5. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

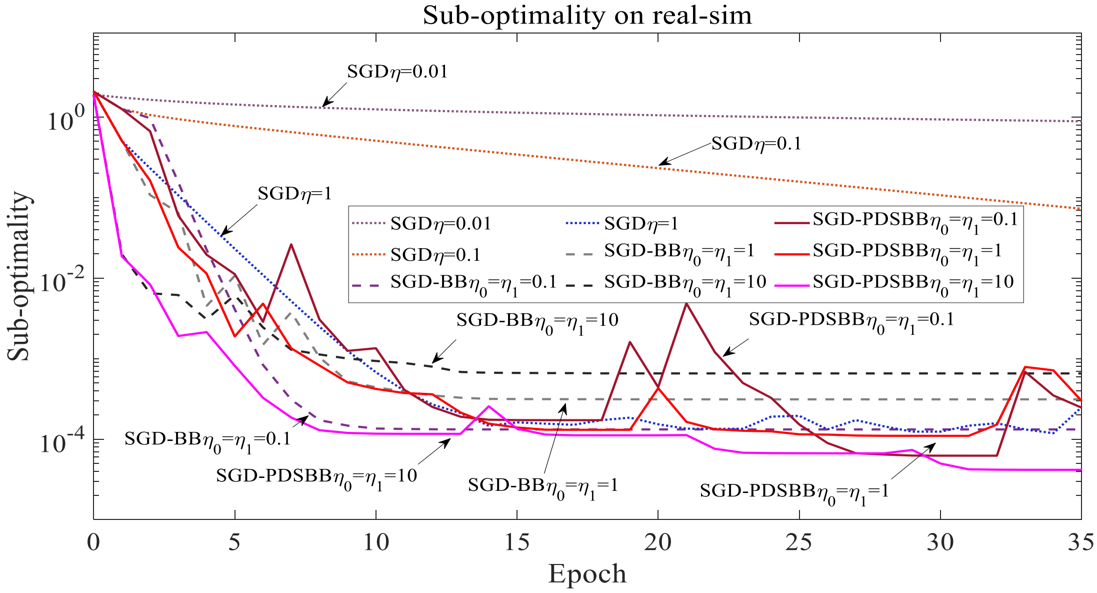

Fig. 6. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(a). Results on the real-sim Dataset with Different Initial Step Sizes

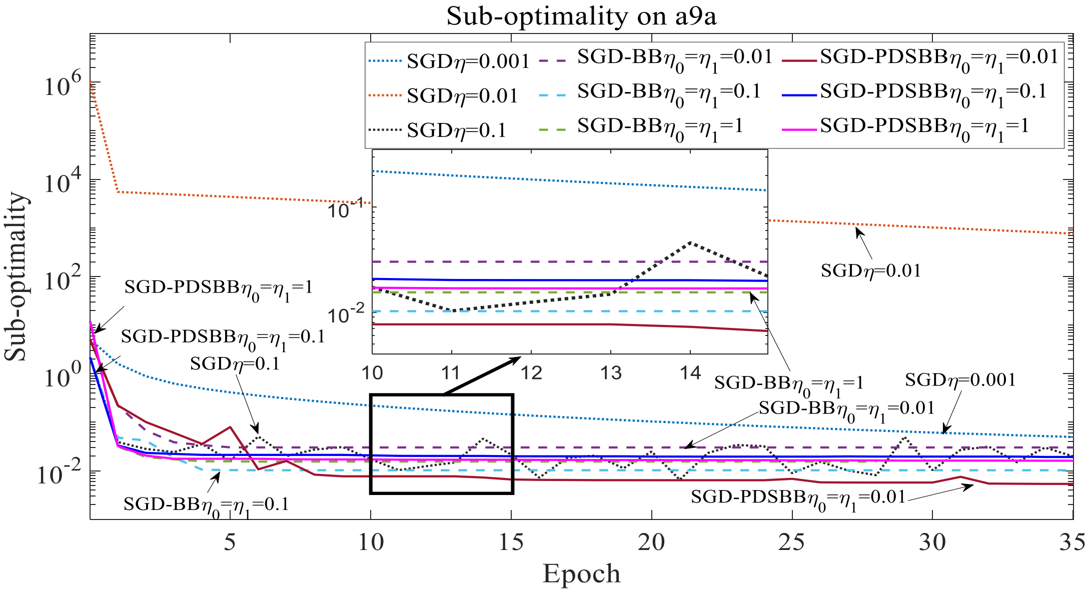

Fig. 6. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(b). Results on the a9a Dataset with Different Initial Step Sizes

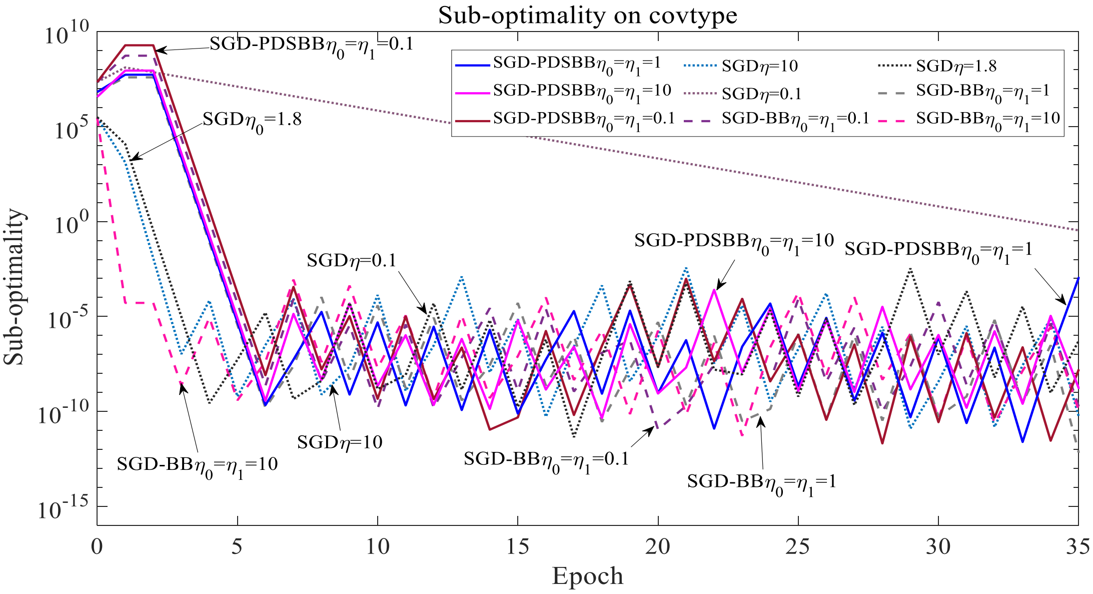

Fig. 6. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(c). Results on the covtype.binary Dataset with Different Initial Step Sizes

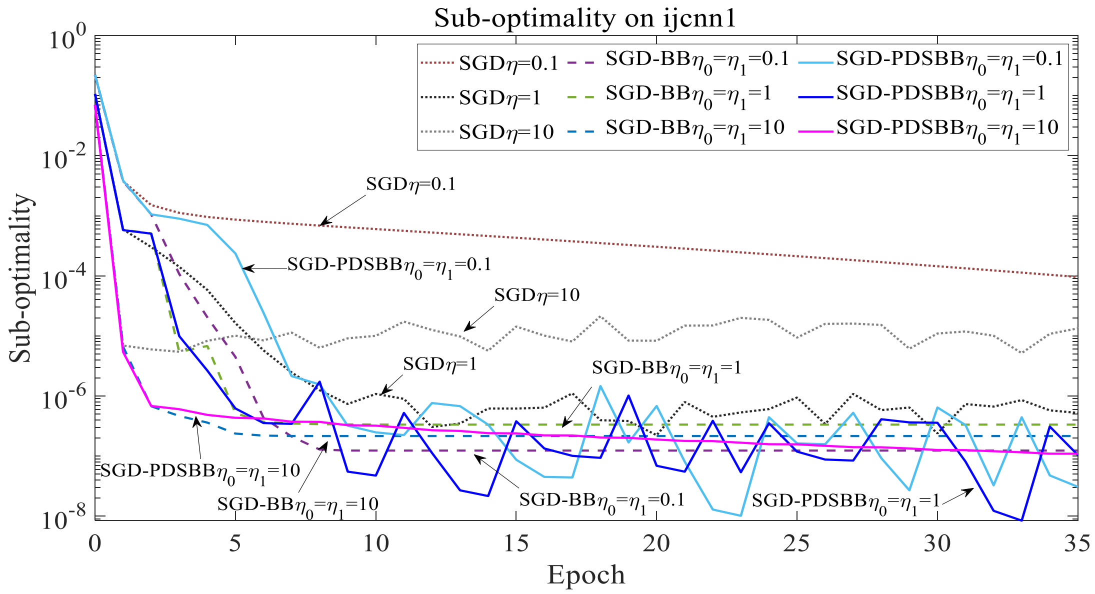

Fig. 6. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (13) Across Datasets
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

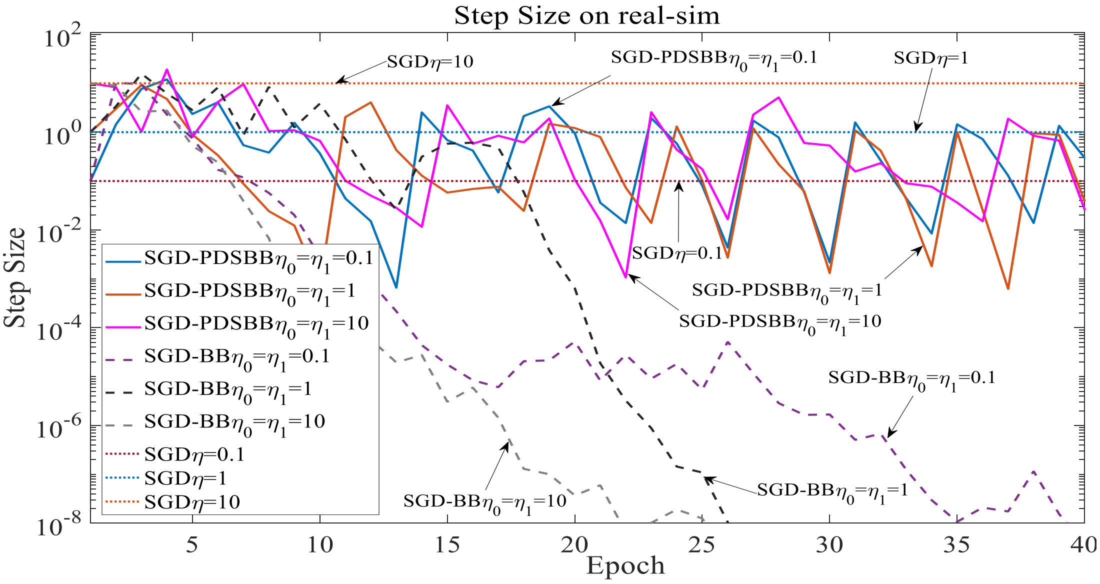

Fig. 7. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(a). Results on the real-sim Dataset with Different Initial Step Sizes

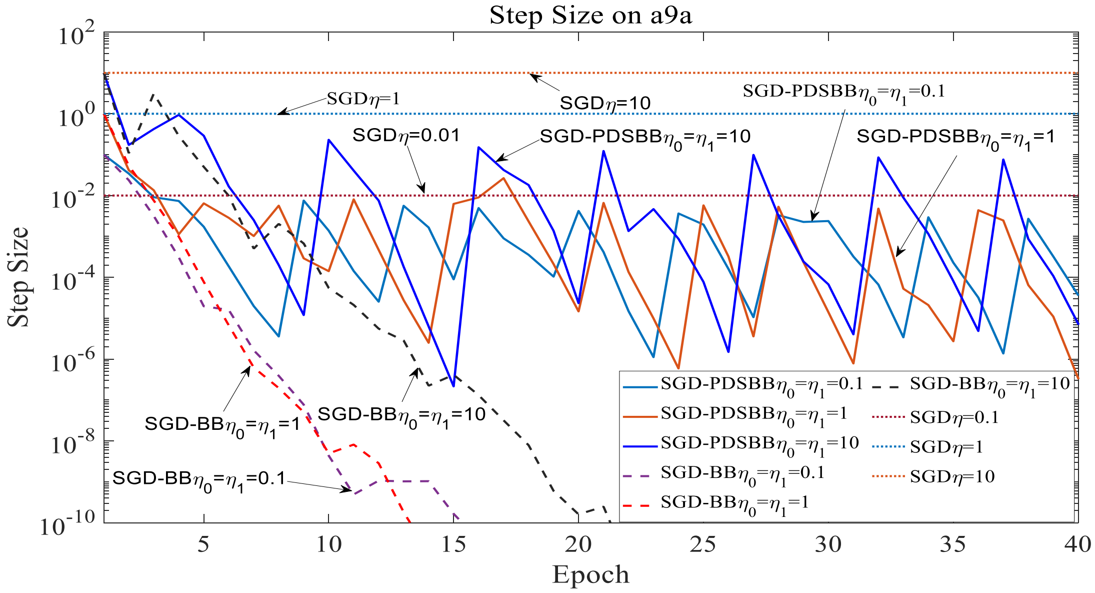

Fig. 7. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(b). Results on the a9a Dataset with Different Initial Step Sizes

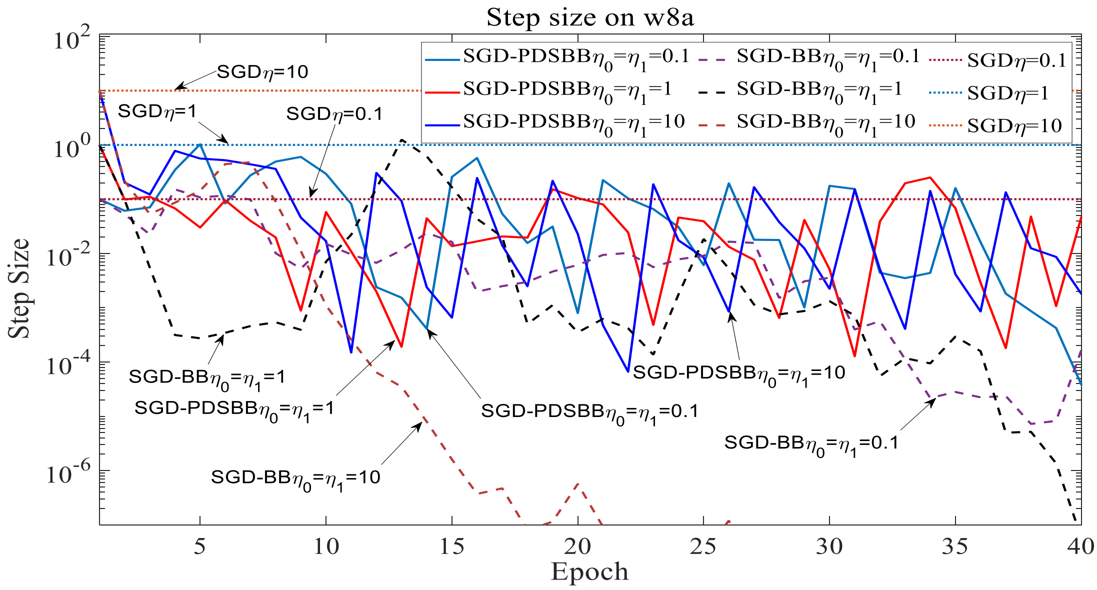

Fig. 7. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(c). Results on the w8a Dataset with Different Initial Step Sizes

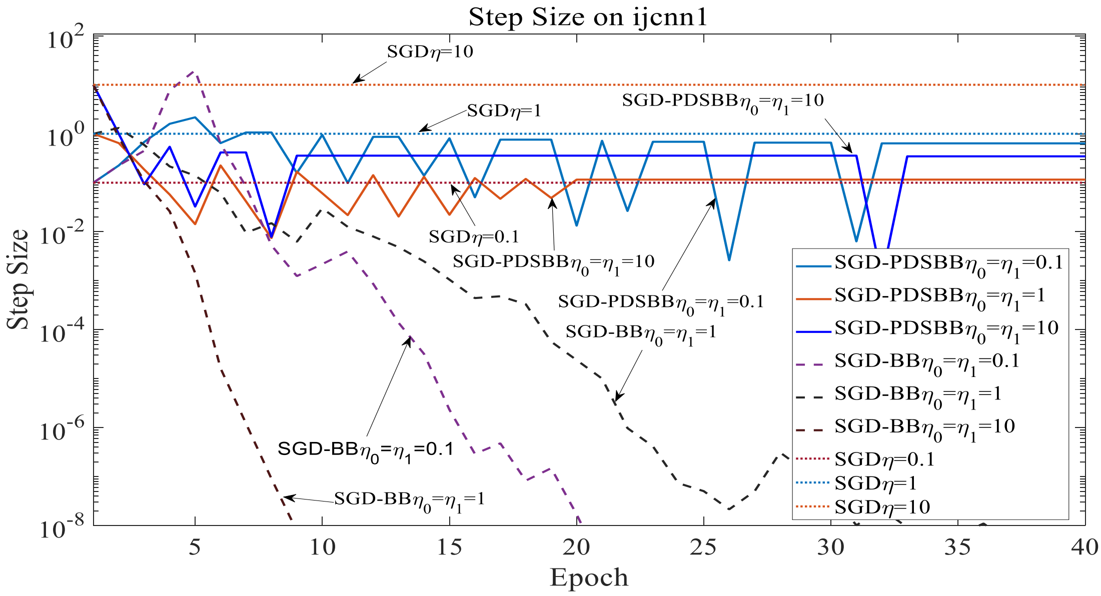

Fig. 7. Step Size Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

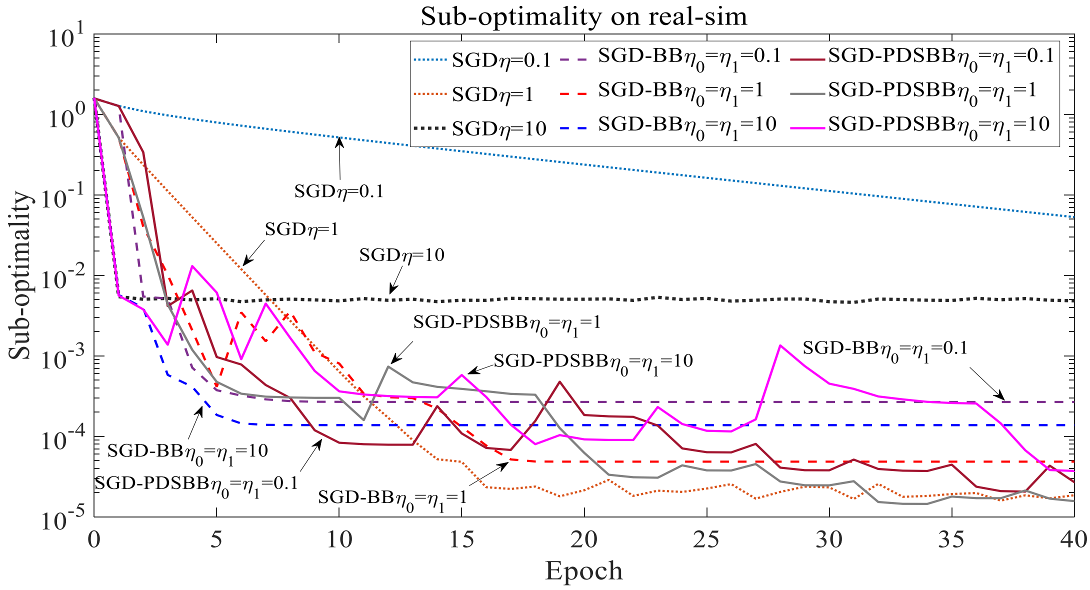

Fig. 8. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(a). Results on the real-sim Dataset with Different Initial Step Sizes

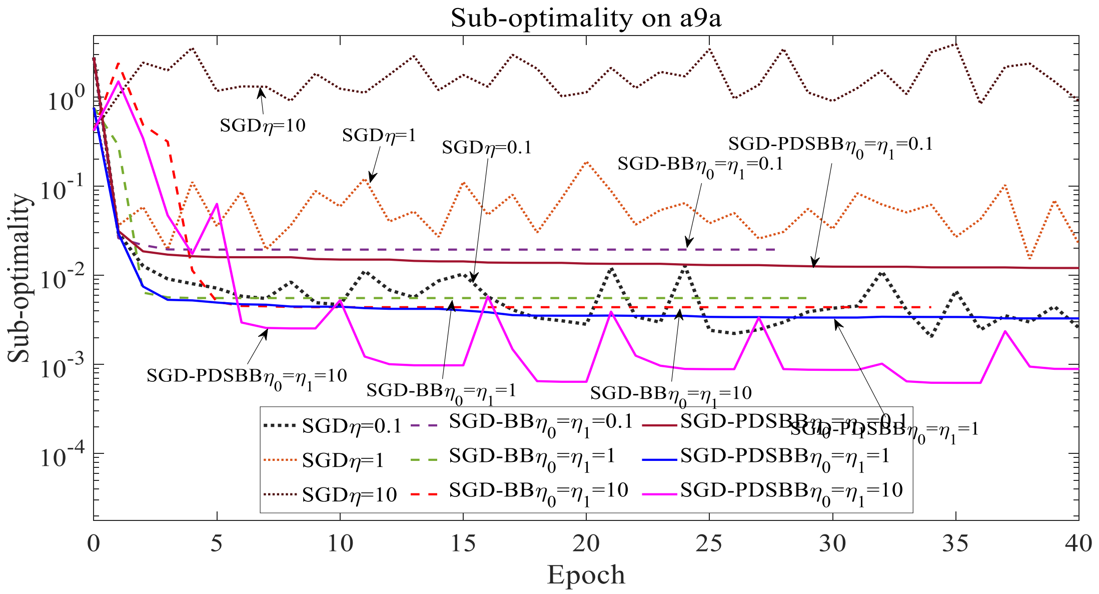

Fig. 8. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(b). Results on the a9a Dataset with Different Initial Step Sizes

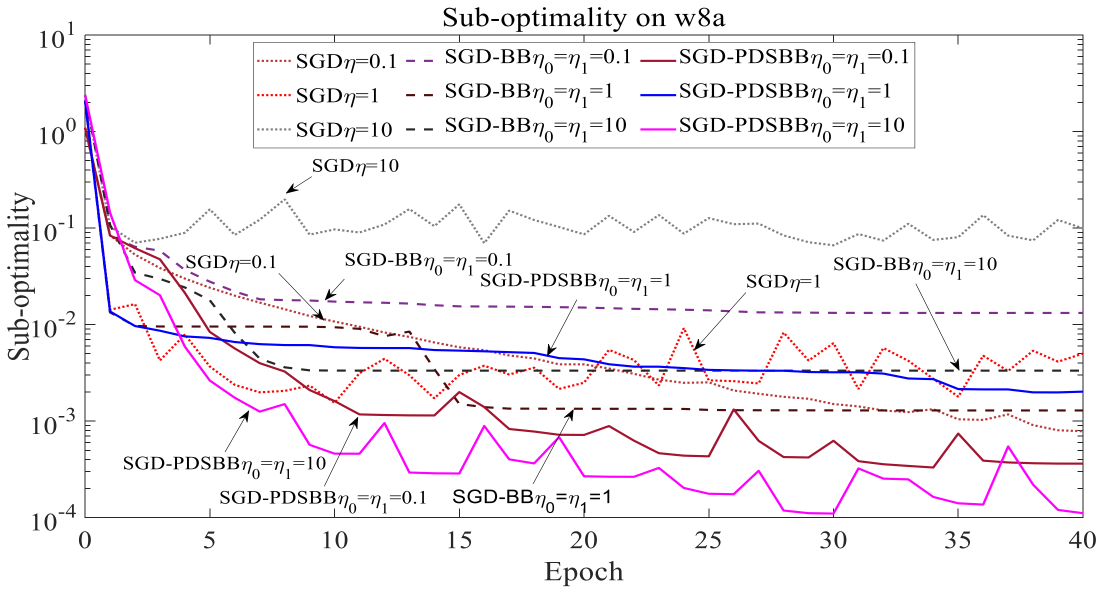

Fig. 8. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(c). Results on the w8a Dataset with Different Initial Step Sizes

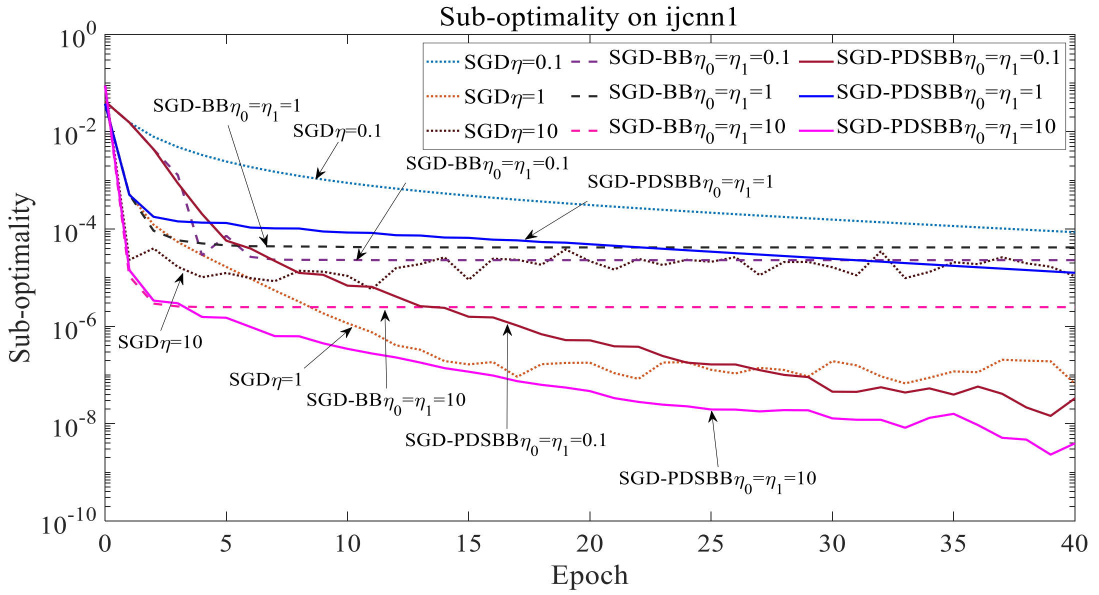

Fig. 8. Sub-Optimality Results of SGD-PDSBB, SGD-BB, and SGD for Model (14) Across Datasets
(d). Results on the ijcnn1 Dataset with Different Initial Step Sizes

REFERENCES

[1] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[2] J. Dean et al., “Large scale distributed deep networks,” in Conf. Rec. 25th Int. Conf. Neural Information Processing Systems - Volume 1, 2012, pp. 1223–1231.

[3] K. Gimpel, D. Das, and N. A. Smith, “Distributed asynchronous online learning for natural language processing,” in Conf. Rec. Fourteenth Conf. Computational Natural Language Learning, 2010, pp. 213–222.

[4] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, “Large-scale matrix factorization with distributed stochastic gradient descent,” in Conf. Rec. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2011, pp. 69–77.

[5] Z. Yang, C. Wang, Y. Zang, and J. Li, “Mini-batch algorithms with Barzilai–Borwein update step,” *Neurocomputing*, vol. 314, pp. 177–185, 2018.

[6] J. Barzilai and J. (Jon) Borwein, “Two-Point Step Size Gradient Methods,” *IMA Journal of Numerical Analysis*, vol. 8, pp. 141–148, Jan. 1988.

[7] J. Konečný, “Stochastic, Distributed and Federated Optimization for Machine Learning,”, Ph.D. dissertation, University of Edinburgh, 2017.

[8] C. Tan, S. Ma, Y. H. Dai, and Y. Qian, “Barzilai-borwein step size for stochastic gradient descent,” *Advances in Neural Information Processing Systems*, pp. 685–693, 2016.

[9] K. Ma et al., “Stochastic non-convex ordinal embedding with stabilized Barzilai-Borwein step size,” in Conf. Rec. 32nd AAAI Conf. Artificial Intelligence, 2018, pp. 3738–3745.

[10] O. Burdakov, Y. Dai, and N. Huang, “Stabilized Barzilai-Borwein method,” *Journal of Computational Mathematics*, vol. 37, no. 6, pp. 916–936, 2019.

[11] Weijuan Shi, Adibah Shuib, and Zuraida Alwadood, "Stochastic Variance Reduced Gradient Method Embedded with Positive Defined Stabilized Barzilai-Borwein," IAENG International Journal of Applied Mathematics, vol. 53, no.4, pp1682-1687, 2023 .

[12] R. Bekkerman, M. Bilenko, and J. Langford, Eds., “Subject Index,” in *Scaling up Machine Learning: Parallel and Distributed Approaches*, Cambridge: Cambridge University Press, 2011, pp. 471–475.

[13] X. Wang and M. Han, “Improved extreme learning machine for multivariate time series online sequential prediction,” *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 28–36, 2015.

[14] A. Carpentier and R. Munos, “Bandit Theory meets Compressed Sensing for high-dimensional Stochastic Linear Bandit,” *Journal of Machine Learning Research*, vol. 22, pp. 190–198, 2012.

[15] D. Manjarres *et al.*, “On the design of a novel two-objective harmony search approach for distance- and connectivity-based localization in wireless sensor networks,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, pp. 669–676, 2013.

[16] X. Luo, H. Liu, G. Gou, Y. Xia, and Q. Zhu, “A parallel matrix factorization based recommender by alternating stochastic gradient decent,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 7, pp. 1403–1412, 2012.

[17] Z. Yang, C. Wang, Z. Zhang, and J. Li, “Mini-batch algorithms with online step size,” *Knowledge-Based Systems*, vol. 165, pp. 228–240, 2019.

[18] Z. Yang, C. Wang, Z. Zhang, and J. Li, “Random Barzilai–Borwein step size for mini-batch algorithms,” *Engineering Applications of Artificial Intelligence*, vol. 72, November 2017, pp. 124–135, 2018.

[19] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in Proc. 2015 3rd Int. Conf. Learning Representations, ICLR, pp. 1–15.

[20] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” in 2010 Proc. 23rd Conf. on Learning Theory (COLT), vol. 12, pp. 257–269.

[21] D. Needell, N. Srebro, and R. Ward, “Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm,” *Mathematical Programming*, vol. 155, no. 1–2, pp. 549–573, 2015.

[22] P. Z. Zhao, “Reactor loop for continuous production of gaseous fission product radioisotopes— I. 235U-loaded molecular sieve target,” in Proc. 32nd Int. Conf. Machine Learning (ICML 2015), vol. 37, 2015, pp. 1–9.

[23] K. Sopyła and P. Drozda, “Stochastic Gradient Descent with Barzilai–Borwein update step for SVM,” *Information Sciences*, vol. 316, pp. 218–233, 2015.

[24] B. T. Polyak and A. B. Juditsky, “Acceleration of Stochastic Approximation by Averaging,” *SIAM Journal on Control and Optimization*, vol. 30, no. 4, pp. 838–855, 1992.

Weijuan Shi is a PhD candidate in Mathematics of Universiti Teknologi MARA, Malaysia. Her main research interests include stochastic optimization and machine learning. She is a lecturer at College of Mathematics and Finance, Hunan University of Humanities, Science and Technology since 2014.

Adibah Shuib is the supervisor of the first author Weijuan Shi. She is an Associate Professor at Universiti Teknologi MARA, Malaysia. She received her PhD degree in Mathematics from University of Birmingham, United Kingdom. Her main research fields include Mathematical Programming / Optimization Models, Graph Theory, Transportation, Logistics and Supply Chain.

Zuraida Alwadood is the co-supervisor of the first author Weijuan Shi. She received her PhD degree in Information Technology and Quantitative Sciences from the Universiti Teknologi MARA, Malaysia. Her top skills are Mathematical Programming/Modelling.