Two-variable Dual Coordinate Descent Methods for Linear SVM with/without the Bias Term

Chi-Cheng Chiu[*] Pin-Yen Lin[*] Chih-Jen Lin[*]

Abstract

Coordinate descent (CD) methods have been a state-of-the-art technique for training large-scale linear SVM. The most used setting is to solve the dual problem of an SVM formulation without the bias term (or an SVM formulation by embedding the bias term in the weight vector). The reason of omitting the bias term is that dual SVM no longer has a linear constraint and the CD procedure of updating one variable at a time is very simple. However, some have criticized the decision of not considering the bias term. To understand the role of the bias term in the design of CD methods for linear SVM, we give a thorough study on two-variable CD. First, if the bias term is not considered, we develop a two-variable CD that is competitive with the commonly used one-variable CD and is superior for difficult problems. The procedure is simple and has theoretical linear-rate convergence. Second, we investigate two-variable CD for linear SVM with the bias term. Analysis shows that CD is much less efficient for such a setting. Therefore, we conclude that in using CD for linear SVM, in general the bias term should not be considered.

1 Introduction

For large and sparse data, linear support vector machines (SVM) have been effective to achieve competitive test accuracy. To train large-scale linear SVM, coordinate descent (CD) methods to solve the dual problem are a state-of-the-art approach. The basic idea is to update a small number of variables at a time while fixing others.

While CD is a classical optimization approach that can be traced back to, for example, [9] for unconstrained quadratic minimization, for linear SVM this type of techniques becomes popular mainly after [10]. They point out that by the special structure of the dual problem of linear SVM, each CD update can be cheaply conducted. Since then, CD has been widely adopted for linear SVM and many subsequent studies including theoretical investigations have been available (e.g., [22, 25]). Another hallmark of CD for linear SVM in [10] is that at each step a simple one-variable sub-problem is minimized and a closed-form solution is available. Thus besides the efficiency, a CD implementation for linear SVM is extremely simple.

The work [10] considers a formulation slightly different from the standard SVM so that the bias term is omitted or embedded in the weight vector. If the bias term is considered, the dual problem contains a linear constraint and each CD step must update at least two variables. In contrast, without the bias term the dual is bound-constrained and a simple CD of using one variable is applicable. However, recently some have criticized the use of the SVM formulation without a bias term.[1] Therefore, an important question is whether the setting in [10] is a must or not. If an effective two-variable CD can be developed for the dual with a linear constraint, then probably a bias term can always be considered.

While for linear SVM it is possible to consider the dual problem with/without a linear constraint, for problems such as linear one-class SVM [21] or SVDD [26], the dual problem must have a linear constraint. Then one-variable CD is not applicable because at least two variables must be updated at a time.

The above discussion motivates us to thoroughly study two-variable CD for linear SVM with/without the bias term. Our two main results are as follows.

- If the bias term is not considered, no works have studied in detail if two-variable CD can compete with one-variable CD by [10]. After some derivations, we develop a simple and efficient solution procedure. Theoretical linear-rate convergence is established. The resulting two-variable CD is in general competitive with one-variable CD by [10], and is superior on difficult problems. See details in Section 3.
- We study two-variable CD for linear SVM with the bias term, where the dual has a linear constraint. Analysis and experiments show that CD for such an

[*] Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

[1] For example, https://github.com/scikit-learn/scikit-learn/pull/4738

optimization problem is much slower than CD for linear SVM without the bias term. Therefore, the decision in [10] to not have the bias term is very essential for the success of CD for linear SVM. Details are in Section 4.

Results in this work imply that for linear one-class SVM or SVDD, if we would like to consider CD methods, new developments may be needed.

A related work to ours is [23], which studied one- and two-variable CD for kernel SVM with/without the bias. Our study shows that the difference between CD for linear SVM with/without a bias term is much more dramatic than the difference in the kernel situation. For other related works on CD for SVM, we cite and discuss them in various places in this paper.

Programs and supplementary materials are available at `https://www.csie.ntu.edu.tw/~cjlin/papers/2var_cd/`.

2 Coordinate Descent Method for Dual of Linear SVM

For training instances $(y_i, \boldsymbol{x}_i), i = 1, \ldots, l$, where $y_i \in \{-1, +1\}$ and $\boldsymbol{x}_i \in R^n$, linear SVM solves the following optimization problem.

$$\min_{\boldsymbol{w}, b} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum\nolimits_{i=1}^{l} \xi(\boldsymbol{w}, b; \boldsymbol{x}_i, y_i), \tag{2.1}$$

where $\xi(\boldsymbol{w}, b; \boldsymbol{x}_i, y_i)$ is a loss function and $C \in (0, \infty)$ is a penalty parameter. The following two loss functions are commonly considered for SVM.

$$\xi(\boldsymbol{w}, b; \boldsymbol{x}, y) \equiv \begin{cases} \max(0, 1 - y(\boldsymbol{w}^T\boldsymbol{x} + b)) & l1 \text{ loss}, \\ \max(0, 1 - y(\boldsymbol{w}^T\boldsymbol{x} + b))^2 & l2 \text{ loss}. \end{cases}$$

The variable b is called the bias term. It is often used for kernel SVM, but for some linear SVM works (e.g., [10]), b is omitted or embedded into $\boldsymbol{w}$ by adding one constant feature to data:

$$\boldsymbol{w} \leftarrow \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}, \ \boldsymbol{x}_i \leftarrow \begin{bmatrix} \boldsymbol{x}_i \\ 1 \end{bmatrix}. \tag{2.2}$$

In Section V of supplementary materials we have additional experiments on (2.2) though observations are generally the same as if the bias is not considered. We further show there that in most cases of our high-dimensional data, linear SVM with and without the bias give equally good models (i.e., similar test accuracy).

If (2.1) is referred to as the primal problem, then the dual optimization problem is

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & f(\boldsymbol{\alpha}) \equiv \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C_i, \forall i, \\ & \boldsymbol{y}^T\boldsymbol{\alpha} = 0 \quad (\text{if } b \text{ is considered}), \end{aligned} \tag{2.3}$$

Algorithm 1 A framework of block CD methods

```
Let $\boldsymbol{\alpha}$ be a feasible point
while $\boldsymbol{\alpha}$ is not optimal do
    Select a working set $B$
    Solve the sub-problem (2.5)
    Update $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha}_B \leftarrow \boldsymbol{\alpha}_B + \boldsymbol{d}_B$
end while
```

where $\boldsymbol{y}^T\boldsymbol{\alpha} = 0$ vanishes if b is omitted,

$$\boldsymbol{e} = [1, \ldots, 1]^T, \quad C_i = \begin{cases} C & l1 \text{ loss}, \\ \infty & l2 \text{ loss}, \end{cases} \text{ and}$$

$$Q_{ij} = \begin{cases} y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j + \frac{1}{2C_i} & l2 \text{ loss and } i = j, \\ y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j & \text{otherwise}. \end{cases} \tag{2.4}$$

While a dual problem without a linear constraint may be easier to be solved, for linear SVM no serious study has been made to confirm this conjecture. On the other hand, for kernel SVM, some past works (e.g., [23]) have conducted a detailed comparison. Therefore, a goal of this work is to fill the gap by studying if the CD method performs similarly or not for the dual of linear SVM with/without the bias.

2.1 CD Methods for Linear SVM

The basic idea of a CD method to solve (2.3) is that at the current $\boldsymbol{\alpha}$, we change elements in a small working set B while fix other components. If

$$N \equiv \{1, \ldots, l\} \setminus B \text{ and } \boldsymbol{d} = \left[\begin{smallmatrix} \boldsymbol{d}_B \\ \boldsymbol{0} \end{smallmatrix}\right],$$

then

$$f(\left[\begin{smallmatrix} \boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N \end{smallmatrix}\right] + \left[\begin{smallmatrix} \boldsymbol{d}_B \\ \boldsymbol{0} \end{smallmatrix}\right]) = \frac{1}{2}\boldsymbol{d}_B^T Q_{BB}\boldsymbol{d}_B + \nabla_B f(\boldsymbol{\alpha})^T\boldsymbol{d}_B + \text{constant},$$

where $\boldsymbol{d}_B$ is the sub-vector used to change $\boldsymbol{\alpha}$. We then minimize the following sub-problem over $\boldsymbol{d}_B$.

$$\begin{aligned} \min_{\boldsymbol{d}_B} \quad & \frac{1}{2}\boldsymbol{d}_B^T Q_{BB}\boldsymbol{d}_B + \nabla_B f(\boldsymbol{\alpha})^T\boldsymbol{d}_B \\ \text{subject to} \quad & \begin{bmatrix} \boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{d}_B \\ \boldsymbol{0} \end{bmatrix} \text{ is feasible}. \end{aligned} \tag{2.5}$$

After solving the above sub-problem, we update $\boldsymbol{\alpha}$ by

$$\boldsymbol{\alpha}_B \leftarrow \boldsymbol{\alpha}_B + \boldsymbol{d}_B. \tag{2.6}$$

A summary of the procedure is in Algorithm 1. Throughout this work, we call the process of finishing an update in (2.6) a CD step. From (2.5), important tasks at each CD step are

- constructing the gradient vector $\nabla_B(\boldsymbol{\alpha})$ in (2.5),
- selecting the working set B, and
- solving the sub-problem.

We discuss past developments in Sections 2.2 and 2.3.

2.2 Gradient Calculation for Linear and Kernel SVM To construct the sub-problem (2.5), Q_{BB} and $\nabla f(\boldsymbol{\alpha})$ must be calculated. We show that the situations between kernel and linear are very different. Form (2.3),

$$\nabla_B f(\boldsymbol{\alpha}) = Q_{B,:}\boldsymbol{\alpha} - \boldsymbol{e}_B. \tag{2.7}$$

Thus calculating $Q_{B,:}$ is the main computational cost. If kernel is used,

$$Q_{i,:}\boldsymbol{\alpha} = \sum\nolimits_{j=1}^{l} Q_{ij}\alpha_j = \sum\nolimits_{j=1}^{l} y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j,$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$ is the kernel function and $\phi(\cdot)$ maps each instance to a higher dimensional space. Assume each $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ costs $O(n)$ operations, where n is the number of features. Then (2.7) requires $O(ln)$ cost. Note that in calculating $Q_{B,:}$, the Q_{BB} sub-matrix needed in (2.5) has also been obtained.

Because $Q_{B,:}$ has been calculated, we can easily maintain the gradient by updating the current $\nabla f(\boldsymbol{\alpha})$ to $\nabla f(\boldsymbol{\alpha} + \boldsymbol{d})$:

$$\nabla f(\boldsymbol{\alpha} + \boldsymbol{d}) = \nabla f(\boldsymbol{\alpha}) + Q_{:,B}\boldsymbol{d}_B. \tag{2.8}$$

For linear SVM, [10] proposed a technique to significantly reduce the cost of constructing the sub-problem from $O(ln)$ to $O(|B|n)$. They notice that if

$$\boldsymbol{u} \equiv \sum\nolimits_{j=1}^{l} \alpha_j y_j \boldsymbol{x}_j \tag{2.9}$$

is available, then

$$Q_{i,:}\boldsymbol{\alpha} = \sum\nolimits_{j=1}^{l} y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \alpha_j = y_i \boldsymbol{u}^T \boldsymbol{x}_i \tag{2.10}$$

is a simple inner product with $O(n)$ cost. To maintain $\boldsymbol{u}$ they consider

$$\boldsymbol{u} \leftarrow \boldsymbol{u} + \sum\nolimits_{j:j\in B} d_j y_j \boldsymbol{x}_j, \tag{2.11}$$

which costs $O(|B|n)$. Eventually the vector $\boldsymbol{u}$ converges to the primal optimal solution $\boldsymbol{w}$. This technique is not applicable to kernel SVM because $\boldsymbol{x}_j$ in (2.9) becomes $\phi(\boldsymbol{x}_j)$ and $\boldsymbol{u}$ may be an infinite dimensional vector.

2.3 Working-set Selection for CD Methods We have shown that the cost for constructing (2.5) for linear SVM is much cheaper than kernel. On the other hand, from (2.8), $\nabla f(\boldsymbol{\alpha})$ can be maintained for kernel but not for linear. This situation causes differences in selecting the working set B.

We begin with discussing the working-set selection for linear SVM without the bias term. The feasible set of (2.3) is

$$\{\boldsymbol{\alpha} \mid 0 \le \alpha_i \le C_i,\ \forall i = 1, \ldots, l\}.$$

The work [10] considers the simplest setting of using cyclic coordinate descent, so each time $B = \{i\}$ is chosen. The sub-problem (2.5) can be easily solved by

$$d = \max(-\alpha_i, \min(C_i - \alpha_i, \frac{-\nabla_i f(\boldsymbol{\alpha})}{Q_{ii}})). \tag{2.12}$$

However, [10] pointed out that in practice using a random permutation at each cycle leads to faster convergence:

$$\boldsymbol{\alpha}_{\pi(1)}, \boldsymbol{\alpha}_{\pi(2)}, \ldots, \boldsymbol{\alpha}_{\pi(l)}, \tag{2.13}$$

where $\pi(1), \ldots, \pi(l)$ is a permutation of $1, \ldots, l$. Alternatively, we may randomly select an index at each CD step to achieve the randomness of the working-set selection. Some theoretical justification was given at, for example, [13, 24]. Others have even studied the setting of selecting the index by an adaptive probability distribution (e.g., [6, 4], and references therein). A summary of the one-variable CD is in Algorithm I of supplementary materials.

If a bias term is considered, the feasible set of (2.3) is

$$\{\boldsymbol{\alpha} \mid \boldsymbol{y}^T\boldsymbol{\alpha} = 0,\ 0 \le \alpha_i \le C_i,\ \forall i = 1, \ldots, l\}.$$

To have that $\left[\begin{smallmatrix}\boldsymbol{\alpha}_B\\ \boldsymbol{\alpha}_N\end{smallmatrix}\right] + \left[\begin{smallmatrix}\boldsymbol{d}_B\\ \boldsymbol{0}\end{smallmatrix}\right]$ is feasible, $\boldsymbol{d}_B$ must satisfy

$$\boldsymbol{y}_B^T(\boldsymbol{\alpha}_B + \boldsymbol{d}_B) = -\boldsymbol{y}_N^T\boldsymbol{\alpha}_N, \text{ and } 0 \le \alpha_i + d_i \le C_i, \forall i \in B.$$

Then B must contain at least two elements. Therefore, regardless of linear or kernel SVM, the above one-variable CD cannot be used.

Next we discuss the difference between working-set selection for kernel and linear SVM. While a random or a cyclic selection is possible, most kernel works (e.g., [11, 20, 12, 5, 7, 14, 16, 27, 1]) consider a greedy selection of using the gradient information.[2] Specifically, because the trick (2.9)-(2.11) for linear SVM is not applicable, (2.8) must be conducted and the whole gradient is available. In contrast, for linear SVM, the greedy setting is not suitable because calculating the gradient causes the cost of each CD step to become l times. We omit further comparisons because our focus here is on linear SVM. Interested readers can check Section 4.1 of [10].

3 Two-variable CD Method for Linear SVM without the Bias Term

To use two rather than one element in each coordinate descent step, we consider a working set $B = \{i, j\}$. For

[2] We cite works proposing greedy working-set selections here. Their convergences may be proved in other studies.

the sub-problem (2.5), we slightly modify the objective function to a proximal setting.

$$\min_{\boldsymbol{d}_B} \quad \frac{1}{2}\boldsymbol{d}_B^T Q_{BB}\boldsymbol{d}_B + \nabla_B f(\boldsymbol{\alpha})^T \boldsymbol{d}_B + \frac{\lambda}{2}\|\boldsymbol{d}_B\|^2, \tag{3.14}$$

where $\lambda > 0$ is a small positive value. In some earlier CD works (e.g., [8, 19, 15]), such a modification on the sub-problem has been considered. For the convenience of the description, we assume that the new term $\lambda\|\boldsymbol{d}_B\|^2/2$ has been absorbed to the original quadratic term by

$$Q_{ii} \leftarrow Q_{ii} + \lambda, \quad Q_{jj} \leftarrow Q_{jj} + \lambda. \tag{3.15}$$

The proximal term is added because first, the proof of linear convergence can be more easily established, and second, from (3.15),

$$Q_{BB} \text{ is positive definite} \tag{3.16}$$

and we will show that the solution procedure of the sub-problem is simpler. The two-variable sub-problem is

$$\begin{aligned} \min_{\boldsymbol{d}_B} \quad & \frac{1}{2}\boldsymbol{d}_B^T Q_{BB}\boldsymbol{d}_B + \nabla_B f(\boldsymbol{\alpha})^T \boldsymbol{d}_B \\ \text{subject to} \quad & 0 \le \alpha_i + d_i \le C_i, \ 0 \le \alpha_j + d_j \le C_j. \end{aligned} \tag{3.17}$$

Solving this sub-problem is more complicated than the one-variable case in (2.12). We give details in Section II.I of supplementary materials.

For the working-set selection, in Section 2 we mentioned that for the one-variable scenario [10] considers a permuted sequence (2.13) for cyclic updates. Now we must choose two variables at a time, so a direct extension is to cyclically consider

$$\pi(1,2), \ldots, \pi(1,l), \pi(2,1), \ldots, \pi(l-1,l), \tag{3.18}$$

which is a permutation of

$$(1,2), \ldots, (1,l), (2,1), \ldots, (l-1,l).$$

Unfortunately, this setting is not practical because the $O(l^2)$ storage to store the sequence is prohibitive for large problems. We propose three feasible settings.

- We permute $\{1,\ldots,l\}$ first, and for each $\pi(i)$, another permutation $\bar{\pi}$ of $\{1,\ldots,l\}$ is generated. The sequence considered is therefore

  $$(\pi(1),\bar{\pi}(1)), \ldots, (\pi(1),\bar{\pi}(l)), \ldots, (\pi(l),\bar{\pi}(l)), \tag{3.19}$$

  though some pairs with $\pi(i) = \bar{\pi}(j)$ must be removed. However, a concern is that such a sequence may not be random enough.
- We randomly select every (i,j) rather than permute indices.
- We permute $1,\ldots,l$ and consider the following pairs.

  $$(\pi(1),\pi(2)), (\pi(3),\pi(4)), \ldots, (\pi(l-1),\pi(l)). \tag{3.20}$$

The selection scheme turns out to be very important as shown in the experiments in Section 5. Some interesting findings will be presented, and we conclude that the setting of using (3.20) gives the fastest convergence.

3.1 Linear Convergence

If a random working-set selection is considered, we can apply the result in [18] to have linear convergence in expectation. For the setting (3.20) that is the best in our experiments, we prove in Section II.II of supplementary materials that the two-variable CD algorithm is a special case of the feasible-descent methods in [28]. Thus the algorithm is linearly convergent.

3.2 Discussion

Shrinking techniques are effective strategies in SVM literature to tentatively remove some bounded variables in the optimization process. We show that the two-variable CD considered here can incorporate such a technique. Details are in Section II.III of supplementary materials.

It is possible to consider a sub-problem without the proximal term. However, the solution procedure is more complicated because the matrix in (3.16) may not be positive definite. This occurs if $l1$-loss SVM is used. We give detailed discussion in Section VII of supplementary materials.

4 Two-variable CD for Linear SVM with the Bias Term

With the bias term, the dual problem (2.3) has a linear constraint. Let $B = \{i,j\}$ be the working set at the current CD step. For the discussion here we use α_i, α_j rather than d_i, d_j to describe the two-variable sub-problem

$$\begin{aligned} \min_{\alpha_i,\alpha_j} \quad & f(\ldots,\alpha_i,\ldots,\alpha_j,\ldots) \\ \text{subject to} \quad & 0 \le \alpha_i \le C_i, \ 0 \le \alpha_j \le C_j, \\ & y_i\alpha_i + y_j\alpha_j = -\boldsymbol{y}_N^T\boldsymbol{\alpha}_N. \end{aligned} \tag{4.21}$$

This sub-problem has been routinely solved in CD methods for kernel SVM, where the bias term is often considered. With the linear constraint, the feasible region becomes a line segment as indicated in Figure 1. In contrast, the sub-problem (3.17) has the whole box region as the feasible set. Details of the procedure to solve (4.21) can be found in for example, Section 6 of [2]. More discussion on two-variable CD for linear SVM with the bias term is in Section III of supplementary materials. Here we focus on the convergence speed in comparison with the situation without the bias term.

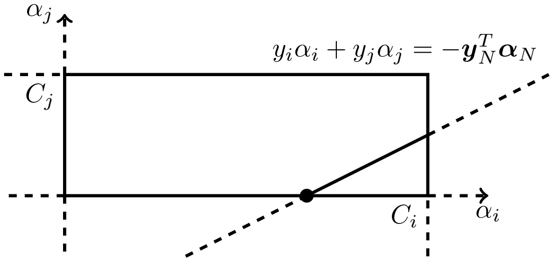

α_j
$y_i\alpha_i + y_j\alpha_j = -\boldsymbol{y}_N^T\boldsymbol{\alpha}_N$
C_j
C_i
α_i

Figure 1: An illustration of the feasible region of (4.21). We assume that the solid circle on the α_i-axis is the current iterate and (4.22) is satisfied.

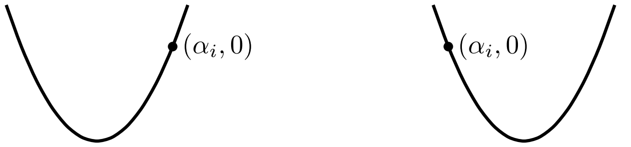
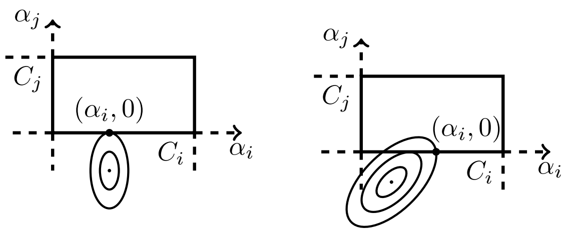


Figure 2: An illustration of the two situations in minimizing the quadratic function (4.21) over the constraint (4.23). The solid circle indicates the current iterate shown on the α_i-axis of Figure 1.



(a) $(\alpha_i, 0)$ is optimal and cannot be further changed.

(b) $(\alpha_i, 0)$, though on the boundary, can be changed to decrease the function value

Figure 3: An illustration showing that without the linear constraint, in general an iterate $(\alpha_i, 0)$ on the boundary is not optimal for the sub-problem. Thus we can improve the objective function value. The contour indicates values of the quadratic objective function

4.1 Difference Between With and Without the Bias Term

In Section 1, we mentioned that [10] did not consider the SVM formulation with a bias term. Therefore, the dual SVM is a bound-constrained problem and one-variable CD is applicable. An extension (still without considering the bias) to two-variable CD has been successfully developed in Section 3. An interesting question now is whether we can have effective two-variable CD for the dual problem with a linear constraint. If we can, then we may prefer always using the standard SVM formulation with the bias term. Unfortunately, here we explain that some subtle differences occur and two-variable CD for the dual problem with a linear constraint is often slower.

If a bias term is considered and assume that $y_i \neq y_j$, then the feasible region is shown in Figure 1. Assume the current $\boldsymbol{\alpha}$ satisfies

$$\alpha_i \in (0, C_i), \alpha_j = 0; \tag{4.22}$$

see the solid circle on the x-axis of Figure 1. We can change (α_i, α_j) only if it does not satisfy the optimality condition. Now the quadratic objective function of (4.21) over the constraint

$$y_i\alpha_i + y_j\alpha_j = -\boldsymbol{y}_N^T\boldsymbol{\alpha}_N \tag{4.23}$$

is in one of the two situations illustrated in Figure 2. From Figures 1 and 2, the objective function can be decreased only if the right sub-figure of Figure 2 occurs. Therefore, for a randomly selected $\{i, j\}$ working set, the chance we can improve the objective function value is only half.

In contrast, if the optimization problem does not have the equality constraint, then unless the relationship between the contour and the feasible region is like Figure 3a, we can always change α_i to improve the function value. For example, in Figure 3b, although $(\alpha_i, 0)$ is on the boundary of the feasible region, we can identify another point with a smaller objective function value.

We now use mathematical derivations to explain the above analysis. Assume $y_i = 1$, $y_j = -1$. With the assumption (4.22), if the bias term is used, the optimality condition is that there exists b such that

$$\nabla_i f(\boldsymbol{\alpha}) + b = 0, \quad \nabla_j f(\boldsymbol{\alpha}) - b \geq 0.$$

This is equivalent to

$$\nabla_j f(\boldsymbol{\alpha}) \geq -\nabla_i f(\boldsymbol{\alpha})$$

and as we said, the chance that it happens may be only half.

If the bias term is not considered, under the assumption (4.22) the optimality condition is

$$\nabla_i f(\boldsymbol{\alpha}) = 0 \text{ and } \nabla_j f(\boldsymbol{\alpha}) \geq 0.$$

Clearly, though the chance that $\nabla_j f(\boldsymbol{\alpha}) \geq 0$ holds may be only half, the probability that $\nabla_i f(\boldsymbol{\alpha}) = 0$ holds is measure zero. Now we see a crucial difference between with and without the bias term: for the former, in the optimality condition, (α_i, α_j) are tied together. In contrast, for the latter, the optimality conditions of α_i and α_j are independent to each other.

Our discussion so far is by assuming that the current iterates satisfies (4.22). That is, one element is free, while the other is bounded. In fact, a similar argument can be made if both elements are bounded, though we do not go through details here.

In summary, if in the optimization process many $\boldsymbol{\alpha}$ variables are bounded, we frequently have pairs like those in (4.22). Then with a high probability the CD step is wasted if the bias term is considered. In Section 5.3, we will experimentally confirm the analysis here.

Several works ([15, 17]) have considered random/cyclic working-set selections in CD for linear/kernel SVM with a bias. They focus on comparing CD between random/cyclic and greedy selections. In contrast, ours focuses on random/cyclic selections but compares CD for linear SVM with and without the bias.

5 Experiments

We consider data sets listed in Table I of supplementary materials. Some are dense sets with $l \gg n$ and some are sparse sets with both large l and n. We present results of some sets by using the $l2$ loss, while leave complete results including those of using the $l1$ loss in supplementary materials.

For each setting, we show CD steps or training time versus the relative difference to the optimal function value:

$$\frac{|f(\boldsymbol{\alpha}) - f(\boldsymbol{\alpha}^*)|}{|f(\boldsymbol{\alpha}^*)|},$$

where $\boldsymbol{\alpha}^*$ is an approximate optimal solution obtained by running many iterations of the algorithm. The regularization parameter C is set to be 1 and 8,192.

5.1 Linear SVM without Bias: Working-set Selection To identify an effective pair-selection scheme for two-variable CD, we compare the following settings discussed in Section 3.

- full: a permutation of $O(l^2)$ elements as shown in (3.18).
- semi-full: the setting in (3.19) to avoid the $O(l^2)$ storage of full.
- random: a random selection.
- perm: the setting in (3.20) to avoid the $O(l^2)$ storage of full.

In Figure III of supplementary materials, we check the number of CD steps versus the relative function-value decrease. Only small data sets are used for this experiment because of the $O(l^2)$ storage requirement of full. Results show that semi-full is significantly worse than others. Therefore, randomness in selecting the working set is very essential.

We also notice that when $C = 1$, full is worse than perm. We think this is because when C is small, the problem is easy and the number of CD steps is less than l^2, the total number of pairs by the full setting. In such a situation, each pair is considered at most once and the sequence may not be random enough.

Between random and perm we observe that perm is slightly better. Experiments in Section 5.2 include more comparisons on them.

5.2 Linear SVM without Bias: Comparison Between One-variable and Two-variable CD We compare the following settings.

- 1-CD-perm: one-variable CD by permuting all indices first and then applying cyclic updates.
- 1-CD-random: one-variable CD by a random selection of indices for update.
- 2-CD-perm: two-variable CD by permuting all indices first and then applying (3.20).
- 2-CD-random: two-variable CD by a random selection of indices for update.

Results of using $C = 1$ and $8,192$ are in Figure 4.

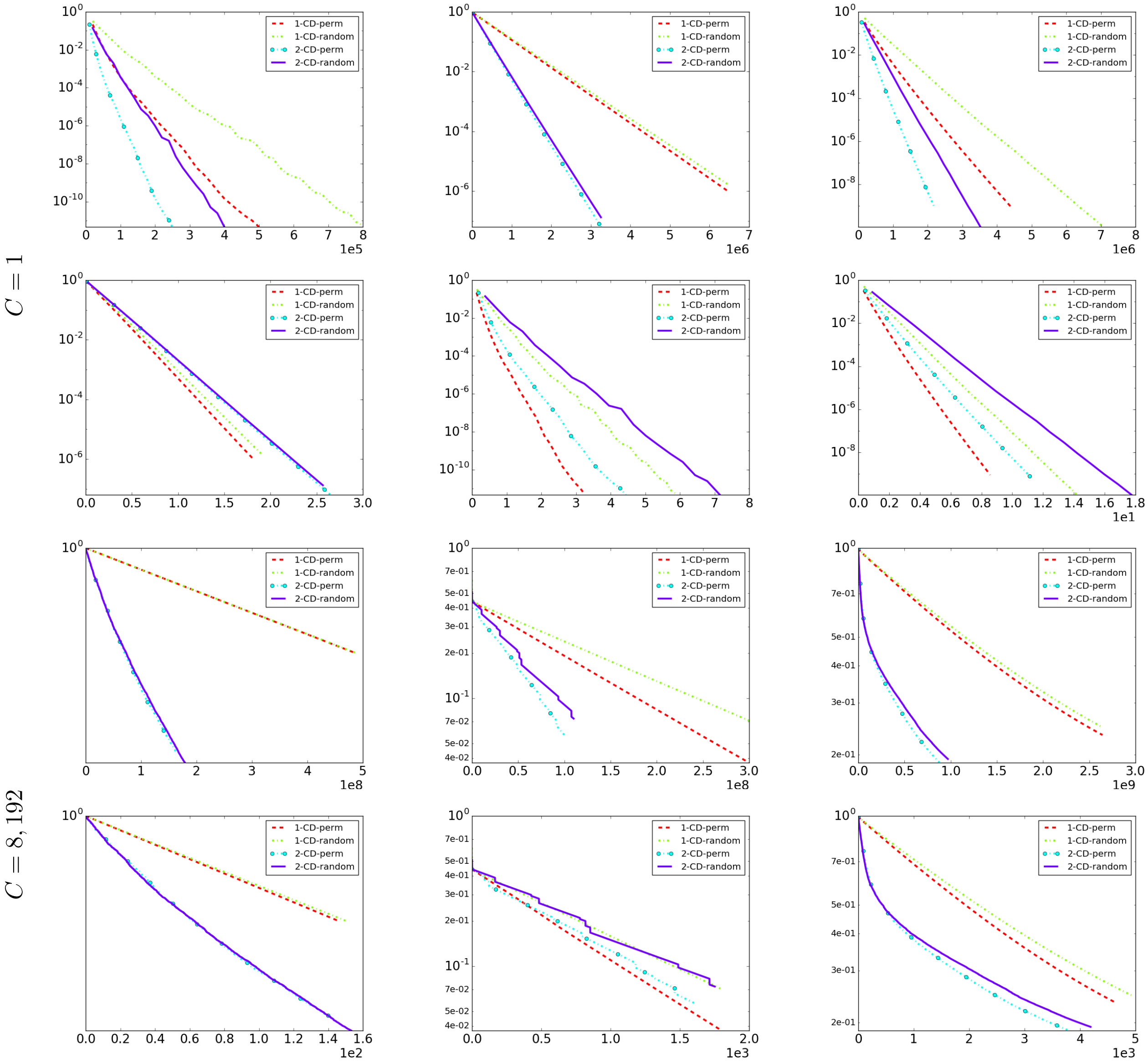
Results indicate that two-variable CD needs fewer steps than one-variable CD. The reason is apparently that more information is considered. However, the cost per CD step is also higher, so in terms of running time, two-variable CD is not faster when $C = 1$. If C is increased to $8,192$, the difference on the number of CD steps is bigger. For some problems, the running time of two-variable CD is significantly shorter. It is well known that using a larger C aims to better fit the data, so the optimization problem becomes more difficult. In such a situation, the proposed two-variable CD is very useful.

Another result in Figure 4 is the comparison between two working-set selections: perm and random. We may expect that for one-variable CD such a comparison has been conducted in some existing works, but interestingly we are not aware of any. In Figure 4, 1-CD-perm is consistently better than 1-CD-random and the gap is sometimes significant. The same result holds for two-variable CD. The reason might be that for the random selection, some variables are less frequently updated than others.

In the above experiment, shrinking is not used. We give comparison results in Section V.II of supplementary materials. Results show that shrinking for two-variable CD is generally as effective as for one-variable CD.

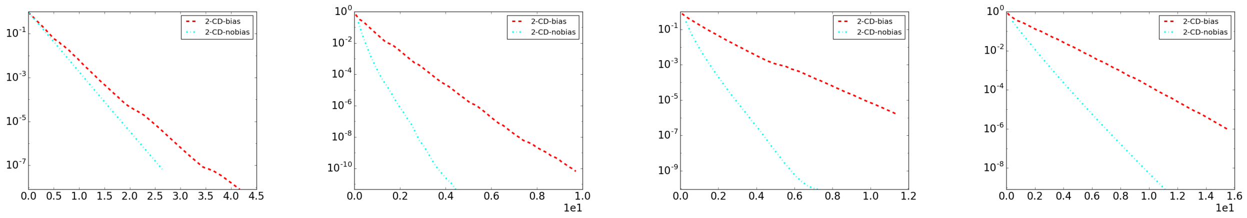
5.3 Comparison Between CD for Linear SVM with/without Bias We compare the following two settings.

- 2-CD-nobias: this is the same as 2-CD-perm in Section 5.2.
- 2-CD-bias: the two-variable CD discussed in Section 4.1 for solving (2.3).



(a) `a9a` (b) `news20.binary` (c) `yahoojp`

Figure 4: A comparison between one-variable and two-variable CD for $l2$ loss with $C = 1$ and $8,192$. For each set under a given C, x-axis in the upper sub-figure is the number of CD steps, while the x-axis in the lower sub-figure is the running time (in seconds).



(a) `a9a` (b) `news20.binary` (c) `rcv1_train.binary` (d) `yahoojp`

Figure 5: Comparison of applying two-variable CD to solve dual of SVM with/without bias. We consider the $l2$ loss and set $C = 1$. The x-axis is the running time in seconds. Shrinking is disabled.

Table 1: The percentage of CD steps that are wasted in the first 200 cycles.

Data set	2-CD-bias	2-CD-nobias
a9a	59.00%	14.77%
ijcnn1	79.88%	53.17%
yahoojp	83.11%	54.67%
rcv1_train.binary	87.82%	66.94%
real-sim	95.17%	22.40%
news20.binary	67.12%	21.00%

Table 2: A summary of CD methods' running time for SVM with/without the bias. Kernel: CD via greedy working-set selection (from [23]). Linear: CD via cyclic/random selection (this work). A > B means A is faster than B.

	SVM problems and CD methods				
	no bias				with bias
	1-CD		2-CD		2-CD
kernel		$<$		$>$ or $\approx$	
linear		$\approx$		$\gg$	

They differ only in the solved dual optimization problem: a linear equality constraint appears in the problem solved by 2-CD-bias, while does not for 2-CD-nobias. Note that their optimal objective values are different, but it is suitable to compare the convergence speed by using their relative difference to the respective optimal value. By considering the $l2$ loss, we present results in Figure 5. We show the convergence along the running time. We have also checked the relation between the running time and the test accuracy; see Section V.III of supplementary materials.

Results indicate that 2-CD-bias is significantly slower than 2-CD-nobias, an observation fully consistent with the analysis in Section 4.1. We conduct a further investigation in Table 1 by showing the percentage of CD steps that are wasted (i.e., in the CD step the selected (α_i, α_j) is already optimal for the sub-problem and cannot be further changed). The percentage of 2-CD-bias is much higher, indicating that its selected pairs frequently fail to reduce the function value.

The work [23] conducted a similar comparison for kernel SVM, where greedy working-set selections are used in two-variable CD. From Figure 5 in their work, the difference between with and without bias is much smaller than ours in Figure 5. The reason is apparently that greedy selections avoid the situation of many wasted CD steps described in Section 4.1.

We mentioned that (2.2) is a way to incorporate the bias term but still avoid the linear constraint in the dual. In Figure XII and Figure XIV of supplementary materials, we compare two-variable CD for this setting with the one having a linear constraint. Results are similar to those in Figure 5, indicating that the analysis in Section 4.1 holds even if (2.2) is applied.

6 Discussion and Conclusions

In this work we broadly discuss issues in extending one-variable CD to two-variable CD for linear SVM. For the commonly used linear-SVM setting without considering the bias term, we derive a simple procedure to solve each sub-problem. The resulting two-variable CD framework is generally competitive and is superior for difficult problems. Further, we establish the theoretical linear convergence. For the SVM formulation with a bias term, we show that because of the linear constraint in the dual optimization problem, CD methods are less effective. We summarize our findings together with those in [23] for kernel SVM in Table 2. Clearly, CD methods for kernel and linear SVM behave very differently. Overall, this work sheds many new insights on the CD methods for training large-scale linear SVM.

For linear one-class SVM and SVDD, their dual problems must have a linear constraint. Subsequent developments of CD methods for them are in [3].

Acknowledgements

This work was supported in part by MOST of Taiwan grant 105-2218-E-002-033.

References

[1] A. Beck, *The 2-coordinate descent method for solving double-sided simplex constrained minimization problems*, J. Optim. Theory Appl., 162 (2014), pp. 892–919.

[2] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM TIST, 2 (2011), pp. 27:1–27:27.

[3] H.-Y. Chou, P.-Y. Lin, and C.-J. Lin, *Dual coordinate-descent methods for linear one-class SVM and SVDD*, in SDM, 2020.

[4] D. Csiba, Z. Qu, and P. Richtárik, *Stochastic dual coordinate ascent with adaptive probabilities*, in ICML, 2015.

[5] R.-E. Fan, P.-H. Chen, and C.-J. Lin, *Working set selection using second order information for training SVM*, JMLR, 6 (2005), pp. 1889–1918.

[6] T. Glasmachers and U. Dogan, *Accelerated coordinate descent with adaptive coordinate frequencies*, in ACML, 2013.

[7] T. Glasmachers and C. Igel, *Maximum-gain working set selection for support vector machines*, JMLR, 7 (2006), pp. 1437–1466.

[8] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136.

[9] C. Hildreth, *A quadratic programming procedure*, Naval Res. Logist., 4 (1957), pp. 79–85.

[10] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, *A dual coordinate descent method for large-scale linear SVM*, in ICML, 2008.

[11] T. Joachims, *Making large-scale SVM learning practical*, in Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998.

[12] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, *Improvements to Platt's SMO algorithm for SVM classifier design*, Neural Comput., 13 (2001), pp. 637–649.

[13] C.-P. Lee and S. J. Wright, *Random permutations fix a worst case for cyclic coordinate descent*, 2016. arXiv preprint arXiv:1607.08320.

[14] C.-J. Lin, S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone, *Decomposition algorithm model for singly linearly constrained problems subject to lower and upper bounds*, J. Optim. Theory Appl., 141 (2009), pp. 107–126.

[15] S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone, *A convergent decomposition algorithm for support vector machines*, Comput. Opt. App., 38 (2007), pp. 217–234.

[16] S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone, *A convergent hybrid decomposition algorithm model for SVM training*, IEEE TNN, 20 (2009), pp. 1055–1060.

[17] I. Necoara and A. Patrascu, *A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints*, Comput. Opt. App., 57 (2013), pp. 307–337.

[18] Y. E. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362.

[19] L. Palagi and M. Sciandrone, *On the convergence of a modified version of SVM^{light} algorithm*, Optim. Methods Softw., 20 (2005), pp. 315–332.

[20] J. C. Platt, *Fast training of support vector machines using sequential minimal optimization*, in Advances in Kernel Methods - Support Vector Learning, Cambridge, MA, 1998, MIT Press.

[21] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, *Estimating the support of a high-dimensional distribution*, Neural Comput., 13 (2001), pp. 1443–1471.

[22] S. Shalev-Shwartz and T. Zhang, *Stochastic dual coordinate ascent methods for regularized loss minimization*, JMLR, 14 (2013), pp. 567–599.

[23] I. Steinwart, D. Hush, and C. Scovel, *Training SVMs without offset*, JMLR, 12 (2011), pp. 141–202.

[24] R. Sun and Y. Ye, *Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version*, Math. Program., (2019).

[25] M. Takáč, A. Bijral, P. Richtárik, and N. Srebro, *Mini-batch primal and dual methods for svms*, in ICML, 2013.

[26] D. M. J. Tax and R. P. W. Duin, *Support vector data description*, MLJ, 54 (2004), pp. 45–66.

[27] P. Tseng and S. Yun, *A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training*, Comput. Optim. Appl., 47 (2010), pp. 179–206.

[28] P.-W. Wang and C.-J. Lin, *Iteration complexity of feasible descent methods for convex optimization*, JMLR, 15 (2014), pp. 1523–1548.