Research Article
An Accelerated Proximal Gradient Algorithm for Singly Linearly Constrained Quadratic Programs with Box Constraints

Congying Han, 1 Mingqiang Li, 2 Tong Zhao, 1 and Tiande Guo 1

1 School of Mathematical Sciences, University of Chinese Academy of Sciences, Shijingshan District, Beijing 100049, China
2 College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Tong Zhao; zhaotong@ucas.ac.cn

Received 4 August 2013; Accepted 23 September 2013

Academic Editors: I. Ahmad and P.-y. Nie

Copyright © 2013 Congying Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, the existed proximal gradient algorithms had been used to solve non-smooth convex optimization problems. As a special nonsmooth convex problem, the singly linearly constrained quadratic programs with box constraints appear in a wide range of applications. Hence, we propose an accelerated proximal gradient algorithm for singly linearly constrained quadratic programs with box constraints. At each iteration, the subproblem whose Hessian matrix is diagonal and positive definite is an easy model which can be solved efficiently via searching a root of a piecewise linear function. It is proved that the new algorithm can terminate at an $\varepsilon$-optimal solution within $O(1/\sqrt{\varepsilon})$ iterations. Moreover, no line search is needed in this algorithm, and the global convergence can be proved under mild conditions. Numerical results are reported for solving quadratic programs arising from the training of support vector machines, which show that the new algorithm is efficient.

1. Introduction

In this paper, we mainly consider the following quadratic programming problem:

$$\min_{x \in \Omega} f(x) = \frac{1}{2} x^T G x - b^T x,$$

(1)

where $G \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite, $x, b \in \mathbb{R}^n$, and the feasible region $\Omega$ is defined by

$$\Omega = \{ x \in \mathbb{R}^n, l \leq x \leq u, c^T x = d \},$$

(2)

where $l, u, c \in \mathbb{R}^n$ and $d$ is a scalar.

The singly linearly constrained quadratic programs with box constraints appear in a wide range of applications such as image processing, biological information and machine learning. Specifically, support vector machine (SVM) is one of the most classical models of (1). It is a promising technique for solving a variety of machine learning and function estimation problems. The SVM learning methodology has been shown to give good performance in a wide variety of problems such as face detection, text categorization, and handwritten character recognition. The number of variables in SVM is so huge that traditional optimization methods cannot be directly applied. Some decomposition method [1–4] with its subproblem being a special case of (1) is the main approach for large-scale SVM problems. The solution of the subproblem by generalized variable projection method (GVPM) and projected gradient method (PGM) is introduced in [5, 6], respectively. Moreover, Zanni, and so forth, also proposes the parallel decomposition algorithms based on these two methods in [7, 8]. For more general large-scale model, some of the parallelization pieces of literature in [9, 10] recently are proposed, but these methods cannot be applied specifically to SVM so far.

In this work, we will give an accelerated proximal gradient algorithm for (1). First, we consider the following nonsmooth convex optimization problem:

$$\min_{x \in \mathbb{R}^n} F(x) = h(x) + g(x),$$

(3)

where $g : \mathbb{R}^n \to (-\infty, \infty]$ is a proper, lower semicontinuous (lsc), convex function and $h$ is convex smooth (i.e., continuously differentiable) on an open subset of $\mathbb{R}^n$ containing
dom \( g = \{ x \mid g(x) < \infty \} \). \( \nabla h \) is Lipschitz continuous on dom \( g \). That is,
\[
\| \nabla h (x) - \nabla h (y) \| \leq L \| x - y \| \quad \forall x, y \in \text{dom } g \tag{4}
\]
for some \( L > 0 \), where, and in what follows, \( \| \cdot \| \) denotes the spectral norm. Obviously, problem (1) is a special case of (3) with \( h(x) = f(x) \) and \( g(x) \) being the indicator function for the feasible region \( \Omega \) defined by
\[
g(x) = \begin{cases} 
0 & \text{if } x \in \Omega, \\
+\infty & \text{else}.
\end{cases} \tag{5}
\]

Recently, great attention has been paid to the solution of (3). Nesterov and Nemirovski [11, 12] study the accelerated proximal gradient method for (1) with an attractive iteration complexity of \( O(1/\sqrt{\varepsilon}) \) for achieving \( \varepsilon \)-optimality. Almost at the same time, Beck and Teboulle [13] give a fast iterative shrinkage-thresholding algorithm (FISTA) which achieves the same convergence rate. After that, Tseng [14] summarizes shrinkage-thresholding algorithms (FISTA) which achieve the same time, Beck and Teboulle[13] give a fast iterative
\begin{align*}
\text{Algorithm 3.} & \quad \text{In what follows, } X^* \text{ and } f^* \text{ denote the set of optimal solutions and optimal value of (1), respectively.}
\end{align*}
Theorem 4. Let \( \{x_k\} \) and \( \{f(x_k)\} \) be the sequences generated by Algorithm 3 with \( L \geq \lambda_{\max}(G) \); then, for \( k \geq 1 \), the following results hold:

(a) \( \{f(x_k)\} \) is nonincreasing
\[
    f(x_k) - f(x_{k+1}) \geq \frac{L}{2} \|x_k - x_{k+1}\|^2; 
\]
\( (16) \)

(b) \( f(x_k) - f(x^*) \leq L\|x_0 - x^*\|^2/2k \), for all \( x^* \in X^* \);

(c) if \( \{x_k\} \) is a convergent subsequence of \( \{x_k\} \) and \( \lim_{j \to \infty} x_{kj} = x^* \) then \( x^* \in X^* \);

(d) if \( G \) is positive definite, then
\[
    \|x_k - x^*\| \leq \sqrt{\frac{L}{2\lambda_{\min}(G)}} \|x_0 - x^*\|. 
\]
\( (17) \)

Proof. (a) Let \( z = x_k, x_{k+1} = P_L(z) \); then from (11) and the optimality of \( P_L(z) \) we have
\[
    f(x_{k+1}) \leq Q(x_{k+1}, x_k) = f(x_k). 
\]

Furthermore, take \( x = x_k \) in (12); we get
\[
    f(x_k) - f(x_{k+1}) \geq \frac{L}{2} \|x_k - x_{k+1}\|^2. 
\]
\( (19) \)

(b) We have
\[
    k(f(x_k) - f(x^*)) \\
    = kf(x_k) - \sum_{n=0}^{k-1} f(x_{n+1}) + \sum_{n=0}^{k-1} f(x_{n+1}) - kf(x^*) \\
    \leq \sum_{n=0}^{k-1} (f(x_{n+1}) - f(x^*)) \\
    \leq -\sum_{n=0}^{k-1} \left( \frac{L}{2} \|x_{n+1} - x_n\|^2 - L(x_n - x^*, x_{n+1} - x_n) \right) \\
    = -\frac{L}{2} \left( \sum_{n=0}^{k-1} (\|x^* - x_{n+1}\|^2 - \|x^* - x_n\|^2) \right) \\
    = -\frac{L}{2} (\|x^* - x_k\|^2 - \|x^* - x_0\|^2) \\
    \leq \frac{L}{2} \|x_0 - x^*\|^2, 
\]
where the first inequality uses the fact that \( \{f(x_k)\} \) is nonincreasing and the second inequality is established by taking \( x = x^*, z = x_k \) in (12). Dividing both sides by \( k \) in the last inequality, we get
\[
    f(x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2k}. 
\]
\( (21) \)

(c) It follows from (b) that, for all \( x^* \in X^* \),
\[
    f(x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2k}. 
\]
\( (22) \)

Let \( j \to \infty \) in the above inequality; we have \( f(x^{**}) \leq f(x^*) \). On the other hand, by the definition of \( x^* \), we have \( f(x^{**}) \geq f(x^*) \). Hence, \( f(x^{**}) = f(x^*) \), which implies that \( x^{**} \in X^* \).

(d) It follows from the definition of \( f(x) \) that, for all \( x^* \in X^* \),
\[
    f(x_k) - f(x^*) \\
    = \langle f(x^*) \rangle (x_k - x^*) + (x_k - x^*)^T G (x_k - x^*) \\
    \geq (x_k - x^*)^T G (x_k - x^*) \\
    \geq \lambda_{\min}(G) \|x_k - x^*\|^2, 
\]
where the first inequality is established by the definition of \( x^* \) and the second inequality, with \( \lambda_{\min}(G) \) denoting the minimum eigenvalue of \( G \), is obtained from the positive definite property of \( G \).

From (23) and (b), we get
\[
    \|x_k - x^*\| \leq \sqrt{\frac{L}{2\lambda_{\min}(G)}} k \|x_0 - x^*\|. 
\]
\( (24) \)

Theorem 4 shows that Algorithm 3 has a result complexity of \( O(1/k) \). Now we introduce a new algorithm with an improved complexity result of \( O(1/k^2) \).

Algorithm 5. We have the following steps.

Step 1. Choose \( z_1 = x_0 \in R^n \), set \( k = 1, t_1 = 1 \).

Step 2. While \( (x_{k-1}, x_k) \) does not satisfy the terminal conditions
\[
    x_k = P_L(z_k), 
\]
\( (25) \)
\[
    t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, 
\]
\( (26) \)
\[
    z_{k+1} = x_k + \left( \frac{t_k - 1}{t_{k+1}} \right)(x_k - x_{k-1}). 
\]
\( (27) \)

Remark 6. In Algorithm 5, the operator \( P_L(\cdot) \) is employed on the point \( x_N \) which is a specific linear combination of the previous two points \( (x_{k-1}, x_{k-2}) \). However, in Algorithm 3, the operator only uses the previous point \( x_{k-1} \). On the other hand, the computational effort of these two algorithms is almost the same except for the computation of (26) in Algorithm 5 which is negligible.

Now we give the promising improved complexity result for Algorithm 5.

Theorem 7. Let \( \{x_k\} \) and \( \{f(x_k)\} \) be the sequences generated by Algorithm 5 with \( L \geq \lambda_{\max}(G) \). Then, for \( k \geq 1 \), the following results hold:

(a) \( f(x_k) - f(x^*) \leq 2L\|x_0 - x^*\|^2/(k+1)^2 \), for all \( x^* \in X^* \);

(b) if \( \{x_k\} \) is a convergent subsequence of \( \{x_k\} \) and \( \lim_{j \to \infty} x_{kj} = x^{**} \), then \( x^{**} \in X^* \);
(c) if $G$ is positive definite, then
\[
\|x_k - x^*\| \leq \sqrt{\frac{L}{2\lambda_{\min}(G)(k+1)^2}}\|x_0 - x^*\|. \tag{28}
\]

**Proof.** (a) Using Lemma 2, the proof is similar to the proof of Theorem 4.4 in [13]. (b) and (c) can be proved in the same way as that of (c) and (d) in Theorem 4, respectively. \qed

### 3. How to Solve $P_L(\cdot)$

In this section, we extend the method studied by Dai and Fletcher [6] to solve $P_L(\cdot)$. In order to obtain $P_L(\cdot)$ in (8), we need to solve the following problem:

\[
\begin{align*}
& \min \frac{L}{2}x^T - q^T x \\
& \text{s.t. } c^T x = d, \quad l \leq x \leq u,
\end{align*} \tag{29}
\]

where $q = Lz + b - Gz$. For a given value of $\lambda$, we consider the following box constrained QP problem:

\[
\begin{align*}
& \min \frac{L}{2}x^T - q^T x - \lambda \left(c^T x - d\right) \\
& \text{s.t. } l \leq x \leq u,
\end{align*} \tag{30}
\]

and denote the minimizer of (30) as $x(\lambda)$. Then,

\[
x(\lambda) = \text{mid}(l, \alpha, u), \tag{31}
\]

where $\alpha = \alpha(\lambda)$ has components $\alpha_i = (q_i + \lambda c_i)/L$ ($i = 1, 2, \ldots, n$) and $\text{mid}(l, \alpha, u)$ is the componentwise operation that supplies the median of its three arguments. That is,

\[
[x(\lambda)]_i = \begin{cases} l_i + \alpha_i - u_i - \max(l_i, \alpha_i, u_i) \\ - \min(l_i, \alpha_i, u_i) \end{cases} \quad (i = 1, 2, \ldots, n). \tag{32}
\]

Define

\[
r(\lambda) = c^T x(\lambda) - d; \tag{33}
\]

then from [6], we know that $r(\lambda)$ is a piecewise linear continuous and monotonically increasing function of $\lambda$. Furthermore, $x(\lambda^*)$ is the optimal of (29) if $\lambda^*$ is located such that $r(\lambda^*) = 0$. Hence, the main task of solving (29) is to find a $\lambda^*$ such that $r(\lambda^*) = 0$. For this purpose, we adopt the algorithm introduced in [6] which consists of a bracketing phase and a secant phase. Numerical experiments in the next section show that this algorithm achieves high efficiency.

### 4. Numerical Experiments

As we know, one of the important applications of SVM is classification. Given a training set

\[
D = \{(z_i, y_i), i = 1, \ldots, n, z_i \in R^n, y_i \in \{-1, 1\}\}, \tag{34}
\]

we need to find a hyperplane to separate the two classes of points. This can be done by solving the following convex quadratic programming problem:

\[
\begin{align*}
& \min \frac{1}{2}x^T Qx - e^T x \\
& \text{s.t. } y^T x = 0, \quad 0 \leq x \leq Ce,
\end{align*} \tag{35}
\]

where $y = [y_1, y_2, \ldots, y_n]^T, e \in R^n$ is the vector of all ones, and $C$ is a positive scalar. $Q$ is an $n \times n$ symmetric and positive semidefinite matrix with entries $Q_{ij} = y_i y_j K(x_i, x_j), i, j = 1, 2, \ldots, n$, where $K : R^n \times R^n \rightarrow R$ is some kernel function.

In this section, we illustrate the performance of Algorithms 3 and 5 with Matlab7.10 on a Windows XP professional computer (Pentium R., 2.79 GHZ). We conduct a test on SVM classification problems with the random data sets and the real world data sets.

The generation of the random data sets is based on four parameters $n, m, \beta$, and $\eta$, where $n$ is the number of samples and $m$ is the dimension of $z_i$ ($i = 1, \ldots, n$). Each element of $z_i$ is randomly generated in $[\beta, \eta]$ ($i = 1, \ldots, n$), and $-1$ or $1$ is randomly generated in the $i$th element of $y$ ($i = 1, \ldots, n$).

We have generated three random data sets with $\beta = -1$, $\eta = 1, m = 5$, and $n = 200, 600, 1000$, respectively. The two real world data sets are the UCI adult data set and the heart data set. For the UCI adult, the versions with $n = 1605, 2265$, and $3185$, are considered. The Gauss radial basis function

\[
K(z_i, z_j) = \exp\left(-\frac{|z_i - z_j|^2}{\sigma^2}\right) \tag{36}
\]

is used in our tests. The parameters $C$ in (35) and $\sigma$ in (36) are set to $(10, \sqrt{40}), (1, \sqrt{3})$, and $(1, 1000)$ for random data sets, heart, and UCI adult data sets respectively.

For all test problems, the initial point is chosen as $x_0 = (0, \ldots, 0)^T$ and the const $L = \sum_{i=1}^{n} Q_{ii}$. The terminal rules for solving $P_L(\cdot)$ are $|r(\lambda)| \leq 10^{-8}$.

For the test problem with random data set ($n = 200$) and heart data set, the running steps are $8000$ and $1000$, respectively. In order to make a visual comparison between Algorithms 3 and 5, we plot the values of $f(x_k)$ with iterations in Figure 1. Furthermore, we compute

\[
CR(j) = \sum_{k=1}^{M} cr(M \times (j-1) + k) \quad (j = 1, 2, \ldots, J), \tag{37}
\]

where $cr(M \times (j-1) + k)$ denotes the number of calculating $r(\lambda)$ in step $M \times (j-1) + k$. In Figure 2, we plot the values of $CR(j) (j = 1, 2, \ldots, J)$ with $M = 100, J = 80$ and $M = 20, J = 50$, respectively.

For the test problems with the rest data sets, the terminal condition is based on the fulfillment of the KKT conditions within $0.001$ (see [15]). The numerical results are shown in
Tables 1 and 2, where the meaning of the indexes are as follows:

(i) \( n \): the scale of the data sets,
(ii) sec: the computing time (in seconds),
(iii) \( k \): the number of iterations,
(iv) raver: the average number of calculating \( r(\lambda) \) in each step.

From Figure 2, we can see that although the total number of calculating \( r(\lambda) \) every 100 (27) steps in Algorithm 3 is less than that in Algorithm 5. However, Algorithm 5 uses much less iterations and time to achieve the approximate solution, compared with Algorithm 3 from Figure 1 and Tables 1 and 2. For the subproblem, less than 4 iterations are used to achieve a very accurate solution on average, which shows that \( P_L(\cdot) \) can be solved efficiently. Moreover, from the column raver in Tables 1 and 2, we can also see that the average
Table 1: Numerical results for Algorithm 3.

<table>
<thead>
<tr>
<th>Problems</th>
<th>n</th>
<th>Sec</th>
<th>k</th>
<th>Raver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>600</td>
<td>19.97</td>
<td>19524</td>
<td>2.98</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>78.23</td>
<td>32153</td>
<td>2.99</td>
</tr>
<tr>
<td>Adult</td>
<td>1605</td>
<td>6.81</td>
<td>1064</td>
<td>1.34</td>
</tr>
<tr>
<td></td>
<td>2265</td>
<td>20.45</td>
<td>1515</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>3185</td>
<td>49.50</td>
<td>2103</td>
<td>1.50</td>
</tr>
</tbody>
</table>

Table 2: Numerical results for Algorithm 5.

<table>
<thead>
<tr>
<th>Problems</th>
<th>n</th>
<th>Sec</th>
<th>k</th>
<th>Raver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>600</td>
<td>5.33</td>
<td>5021</td>
<td>3.68</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>19.05</td>
<td>7095</td>
<td>3.30</td>
</tr>
<tr>
<td>Adult</td>
<td>1605</td>
<td>1.41</td>
<td>88</td>
<td>2.82</td>
</tr>
<tr>
<td></td>
<td>2265</td>
<td>2.84</td>
<td>106</td>
<td>2.97</td>
</tr>
<tr>
<td></td>
<td>3185</td>
<td>6.30</td>
<td>125</td>
<td>2.98</td>
</tr>
</tbody>
</table>

number of calculating $r(\lambda)$ with real world data sets is slightly less than that with random data sets.

5. Conclusion and Future Work

We have extended the accelerated proximal point method to the solution of singly linearly constrained quadratic programming with box constraints. The new algorithm is proved to be globally convergent. Numerical results also show that the new algorithm performs well on medium-scale quadratic programs. On the other hand, solving the subproblem by searching a root of a piecewise linear continuous function is very cheap. Considering the good performance of the new Algorithm 5, we can apply it to the solution of the subproblems in decomposition methods for large-scale SVM problems. Moreover, a parallel version of this algorithm combined with the theory in [9, 10] is also a direction of our future research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (11101420, 71271204) and the Excellent Young Scientist Foundation of Shandong Province (2010BSE06047).

References
