2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)

An Active Set Method for Solving Certain Support Vector Machine Problems

Amal Al-Saket
Department of Mathematics
Al-Zaytoonah University
Amman, Jordan
amal_saket@zuj.edu.jo

Dua' Arman
Computer Engineering
Jordanian Engineers Association
Amman, Jordan
du3tame@gmail.com

***Abstract*—We propose an active set method to solve the dual of the convex quadratic programming problem which is the core of the support vector machine (SVM) training. The method stems from the more general method developed by the first author. By using the special handling of certain quadratic programming problems where the Hessian matrix in the objective function is given as a product of a matrix and its transpose, and by simplifying the solution of the linear system arising at each iteration of the method, we were able to produce an implementation for certain SVMs. The results of an experiment using MATLAB are reported.**

Keywords—quadratic programming (QP); Karush–Kuhn–Tucker (KKT) conditions; active set methods; orthogonal factorization; support vector machines

I. INTRODUCTION

In this paper, we derive an implementation targeted to linearly binary separable SVMs based on the algorithm described in Al-Saket [1]. The goal of the binary SVM classifier is to find a separating hyperplane, given a set of n data patterns $X_j \in R^m$ and corresponding binary labels $y_j \in \{1,-1\}$ that partitions the labelled data points X_j while maximizing the margin between the hyperplane and the closest data point X_j to the hyperplane. The hyperplane can be described by the linear equation $w^T X + b = 0$ where w is normal to the hyperplane and is called the weights-vector and the scalar b is called the bias. The predicted label of X then becomes +1 or −1 according to the sign of $(w^T X + b)$ positive or negative, respectively. Finding a separating hyperplane with maximal margin width can be formulated as a QP problem as

$$\min_{w\in\Re^m, b\in\Re} \tfrac{1}{2}\|w\|^2 \tag{1}$$

$$\text{subject to} \quad y_j(w^T X_j + b) \geq 1 \quad , \quad \text{for} \quad j = 1,\ldots,n$$

Here, we assume that the data points are linearly separable. That is a hyperplane that perfectly classifies data points can be found. This case is called a hard- margin formulation of the SVM. We leave it for future work, which is more practical in fact, the soft- margin case where some data points violate the constraints. We refer the reader to [2] and [3] for tutorials on the subject of SVM. See also, [4] and [5].

Let $P_j = y_j X_j$, for $j = 1,\ldots,n$, $P = [P_1, P_2 \cdots P_n]$, $Y = [y_1 \ y_2 \ \cdots \ y_n]$, $\alpha = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_n]^T$ and $e = [1 \ 1 \ \cdots 1]^T$, then the dual of (1) is the QP problem

$$\min_{\alpha\in\Re^n} \tfrac{1}{2}\|P\alpha\|^2 - e^T\alpha = \tfrac{1}{2}\alpha^T P^T P\alpha - e^T\alpha \tag{2}$$

$$\text{subject to} \quad Y\alpha = 0, \quad \alpha \geq 0$$

Where, $\|\cdot\|$ denotes the Euclidean norm of a vector.

Problem (2) is, of course, a problem of positive semi-definite quadratic programming.
In this article we develop an active set method to solve (2) that exploits the factored form $P^T P$ of the Hessian matrix, where our method works on the part P without having to compute the dot products $P_i^T P_j$ to form the entries of $P^T P$. Gill and Murray [6] suggest that quadratic programming problems, for which the Hessian matrix in the objective function is given as a product of a matrix and its transpose should be tackled using specialized techniques and, in this case, active set methods will be more efficient than other methods. Active set methods are among typical approaches to solving quadratic programming problems. For general discussions of quadratic programming and formulation of the dual problem see [7], [8] and [9]. See also, [10] and [11].

The algorithm given here generates a finite sequence of reduced dimension sub-problems without non-negativity constraints. The solution of a sub-problem involves solving a linear system where we use an orthogonal factorization of the coefficient matrix which is a submatrix of the joint matrix $\begin{bmatrix} Y \\ P \end{bmatrix}$. The matrix of the new sub-problem is defined by augmenting a single column or by an "exchange" step where one column replaces another.

The rest of the paper is organized as follows. In the next section we introduce the active set method to solve (2) and then present the whole algorithm which includes our method as well as the recovering of w and b from the optimal solution to (2). We also establish the finite convergence of our method. In Section III, we describe the simplification of the linear system, the updating procedures of the orthogonal factorization as well as an initialization procedure. In section IV, we present the results obtained when our method was applied to one famous data set. Finally, we have a conclusion section.

We would like to note that, for the sake of brevity, the proofs of the lemmas and some results were omitted.

II. BASIC THEORY AND FORMULATION OF THE METHOD

For α to be an optimal solution to (2) it is both necessary and sufficient (because of the objective convexity and constraint linearity) that α and some number μ (the Lagrange multiplier for the constraint $Y\alpha = 0$) satisfy the (KKT) conditions

$$Y\alpha = 0, \quad \alpha \geq 0$$
$$\mu Y^T + P^T P\alpha - e \geq 0$$
$$\mu y_j + P_j^T P\alpha - 1 = 0 \quad \text{or} \quad \alpha_j = 0 \quad \text{for each } j = 1, \ldots, n$$

Throughout, $J \subseteq \{1,2,\cdots,n\}$ denotes a nonempty set of column indices corresponding to a submatrix $\begin{bmatrix} Z \\ S \end{bmatrix}$ of columns of $\begin{bmatrix} Y \\ P \end{bmatrix}$. The algorithm revises J until it is the set of indices corresponding to variables α_j that take on positive values at the optimal solution. Hence the complement of J corresponds to the non-negativity constraints that are active at the solution. Each submatrix generated by the algorithm satisfies the condition that

$\begin{bmatrix} Z \\ S \end{bmatrix}$ has full column rank. (3)

For each such matrix we require a solution (β, μ) to the linear system

$$Z\beta = 0$$
$$\mu Z^T + S^T S\beta = e \tag{4}$$

where e is a column vector of ones of appropriate size.

Lemma 1: If assumption (3) holds then system (4) has a solution (β, μ). Furthermore, β is unique. ■

Lemma 2: (β, μ) solves (4) if and only if β minimizes $\frac{1}{2}\|S\beta\|^2 - e^T\beta$ s.t. $Z\beta = 0$. In this case $\frac{1}{2}\|S\beta\|^2 - e^T\beta = -\frac{1}{2}(e^T\beta)$. ■

Lemma 3: (Optimality Criterion)

Suppose that (β, μ) solves (4) and J corresponds to $\begin{bmatrix} Z \\ S \end{bmatrix}$. Define the corresponding n-vector α by

$$\alpha_j = \beta_j \text{ if } j \in J \text{ and } \alpha_j = 0 \text{ if } j \notin J.$$

If $\alpha_j \geq 0$ and $y_j\mu + P_j S\beta - 1 \geq 0$ for all $j \notin J$ (5)

the (KKT) optimality conditions

■

and, hence, α is an optimal solution to (2).

If α does not solve (2) i.e.

$$y_h\mu + P_h^T S\beta - 1 < 0 \quad \text{for some } h \notin J \tag{6}$$

then the constraint $\alpha_h = 0$ should be made inactive by appending index h to J.

Lemma 4: Suppose that assumption (3) holds and (β, μ) solves system (4). If (6) holds then either

$\begin{bmatrix} Z & y_h \\ S & P_h \end{bmatrix}$ has full column rank (7)

or there exists a vector $\bar{r}$ such that

$\begin{bmatrix} Z \\ S \end{bmatrix}\bar{r} = \begin{bmatrix} y_h \\ P_h \end{bmatrix}$ and $e^T\bar{r} - 1 = y_h\mu + P_h^T S\beta - 1 < 0$. ■ (8)

In Section III we give a procedure for testing (7) and determining $\bar{r}$ when (7) does not hold.

Lemma 5: (Augmentation)

Suppose that assumption (3) holds, (β, μ) solves system (4) associated with the matrix $\begin{bmatrix} Z \\ S \end{bmatrix}$ and (6) and (7) hold. Then the augmented system (4) associated with the matrix $\begin{bmatrix} Z & y_h \\ S & P_h \end{bmatrix}$ has a solution $(\beta^+, \beta_h^+, \mu^+)$ such that $\beta_h^+ > 0$ and

$$\frac{1}{2}\left\|\begin{bmatrix} S & P_h \end{bmatrix}\begin{bmatrix} \beta^+ \\ \beta_h \end{bmatrix}\right\|^2 - e^T\begin{bmatrix} \beta^+ \\ \beta_h^+ \end{bmatrix} - \left(\frac{1}{2}\|S\beta\|^2 - e^T\beta\right) =$$

$$= \frac{1}{2}\beta_h^+\left(y_h\mu + P_h^T S\beta - 1\right) < 0$$

Furthermore, if $\beta > 0$, if $\beta^+ \not> 0$, if $\bar{t}$ is the largest value of $t \in [0,1]$ such that $t\begin{bmatrix} \beta^+ \\ \beta_h^+ \end{bmatrix} + (1-t)\begin{bmatrix} \beta \\ 0 \end{bmatrix} \geq 0$ and if the column vector of dimension $|J|+1$, γ is defined by $\bar{t}\begin{bmatrix} \beta^+ \\ \beta_h^+ \end{bmatrix} + (1-\bar{t})\begin{bmatrix} \beta \\ 0 \end{bmatrix}$ then $\bar{t} > 0$, $\gamma_h > 0$, $\gamma \geq 0$, $\begin{bmatrix} Z & y_h \end{bmatrix}\gamma = 0$, $\gamma_k = 0$ for some $k \in J$ and $\frac{1}{2}\|\begin{bmatrix} S & P_h \end{bmatrix}\gamma\|^2 - e^T\gamma < \frac{1}{2}\|S\beta\|^2 - e^T\beta$. ■

Lemma 6: (Unboudedness/Exchange)
Suppose that assumption (3) holds and (β, μ) solves (4) with $\beta > 0$ and (6) holds but (7) does not hold, i.e., there exists a vector $\bar{r}$ satisfying (8). If $\bar{r} \leq 0$ then problem (2) is unbounded. Otherwise, let $k \in J$ be an index such that

$$\frac{\beta_k}{\bar{r}_k} = \hat{t} = \min\left\{\frac{\beta_j}{\bar{r}_j} \middle| j \in J, \bar{r}_j > 0\right\} \tag{9}$$

Let $\widetilde{J} = J \setminus \{k\}$, $\widetilde{Z}$ and $\widetilde{S}$ be the corresponding submatrices of Z and S, respectively. Let $\hat{\beta}$ be defined by $\hat{\beta}_j = \beta_j - \hat{t}\bar{r}_j$ if $j \in \widetilde{J}$ and $\hat{\beta}_h = \hat{t}$. Then $\hat{\beta} \geq 0$, $\left[\widetilde{Z} \;\; y_h\right]\hat{\beta} = 0$ and $\frac{1}{2}\left\|\left[\widetilde{S} \;\; P_h\right]\hat{\beta}\right\|^2 - e^T\hat{\beta} < \frac{1}{2}\|S\beta\|^2 - e^T\beta$.

Also, the system (4) associated with $\widetilde{J} \cup \{h\}$ has a solution $(\widetilde{\beta}, \widetilde{\mu})$ such that $\frac{1}{2}\left\|\left[\widetilde{S} \;\; P_h\right]\widetilde{\beta}\right\|^2 - e^T\widetilde{\beta} < \frac{1}{2}\|S\beta\|^2 - e^T\beta$. Furthermore, if $\widetilde{\beta} \not> 0$, if $\bar{t}$ is the largest value of $t \in [0,1]$ such that $t\widetilde{\beta} + (1-t)\hat{\beta} \geq 0$ and if the $\left|\widetilde{J}\right| + 1$ vector γ is defined by $\gamma = \bar{t}\widetilde{\beta} + (1-\bar{t})\hat{\beta}$, then $\gamma \geq 0$, $\left[\widetilde{Z} \;\; y_h\right]\gamma = 0$, $\frac{1}{2}\left\|\left[\widetilde{S} \;\; P_h\right]\gamma\right\|^2 - e^T\gamma < \frac{1}{2}\|S\beta\|^2 - e^T\beta$, $\gamma_h > 0$ and $\gamma_k = 0$ for some $k \in \widetilde{J}$. ■

Algorithm 1: The outline of the algorithm is the following:

1) Part 1: Solving the dual problem

We assume that we have an initial index J, corresponding matrices Z and S satisfying the assumption (3) such that system (4) has a solution $\beta > 0$. In Section III, we give an initialization phase.

Step 1:
Solve system (4) for (β, μ). If $\beta > 0$ go to Step 2, otherwise, go to Step 3.

Step 2:
Test the validity of (5). If (5) holds then "OPTIMALITY" STOP. Otherwise, (6) holds, test the validity of (7). If (7) holds go to Step 2.1 and otherwise go to Step 2.2.

2.1: Set $\hat{\beta}_j = \beta_j$ for each $j \in J$ and $\hat{\beta}_h = 0$, append index h to J, column $\begin{bmatrix} y_h \\ P_h \end{bmatrix}$ to $\begin{bmatrix} Z \\ S \end{bmatrix}$ and go to Step 1.

2.2: Calculate $\bar{r}$. If $\bar{r} \leq 0$ then "UNBOUNDEDNESS" STOP. Otherwise, compute $\hat{t}$ as in (9); delete index k from J, column $\begin{bmatrix} y_k \\ P_k \end{bmatrix}$ from $\begin{bmatrix} Z \\ S \end{bmatrix}$; append index h to J, column $\begin{bmatrix} y_h \\ P_h \end{bmatrix}$ to $\begin{bmatrix} Z \\ S \end{bmatrix}$; set $\hat{\beta}_j = \beta_j - \hat{t}\bar{r}_j$ for each $j \in J$, $j \neq h$ and $\hat{\beta}_h = \hat{t}$; and go to Step 1.

Step 3:
Compute $\bar{t}$, the maximum value of $t \in [0,1]$ such that $t\beta + (1-t)\hat{\beta} \geq 0$ and let $\gamma = \bar{t}\beta + (1-\bar{t})\hat{\beta}$. For each j such that $\gamma_j > 0$, set $\hat{\beta}_j = \gamma_j$ and for each k such that $\gamma_k = 0$, delete index k from J, column $\begin{bmatrix} y_k \\ P_k \end{bmatrix}$ from $\begin{bmatrix} Z \\ S \end{bmatrix}$. Then go to Step 1.

2) Part 2: Complete solving the SVM problem

Compute $w = \sum_{j \in J} \beta_j \, y_j \, X_j$ and $b = y_j - w^T X_j$ for any $j \in J$.

Lemma 7: Suppose that Step 3 is entered with $\begin{bmatrix} Z \\ S \end{bmatrix}$ satisfying assumption (3) and that γ determined at this execution of Step 3 satisfies

$$\gamma \geq 0$$
$$Z\gamma = 0$$
$$\gamma_{h_0} > 0, \; \gamma_k = 0 \text{ for some } k \neq h_\circ \tag{10}$$
$$\text{and } \frac{1}{2}\|S\gamma\|^2 - e^T\gamma < \frac{1}{2}\left\|S^\circ\beta^\circ\right\|^2 - e^T\beta^\circ$$

Where, at the most recent execution of step 2, J°, Z°, S° and β° are the entering values of J, Z, S, β and $h_\circ$ was the index added to J°. Let the new values of J, Z, S defined at Step 3, be denoted by J^-, Z^- and S^-, respectively. Then the new system (4) has a solution $\beta = \beta^-$ such that either

$$\beta^- > 0 \text{ and } \frac{1}{2}\left\|S^-\beta^-\right\|^2 - e^T\beta^- < \frac{1}{2}\left\|S^\circ\beta^\circ\right\|^2 - e^T\beta^\circ \tag{11}$$

or $\beta^- \not> 0$ and the above hypotheses (10) are satisfied with $\gamma = \gamma^-$, $Z = Z^-$ and $S = S^-$, where γ^- is the value of γ determined at the next execution of Step 3. ■

In the following theorem we establish the finite convergence of *Part 1* of the algorithm.

Theorem 1: Suppose that the initial condition holds. Then after a finite number of executions of steps 1, 2 and 3, *Part 1* of the algorithm terminates at Step 2 with either an optimal solution to (2) or an indication of unboundedness.

Proof: To each J considered at Step 1 there corresponds a unique β solution of (4) and a corresponding objective value $\frac{1}{2}\|S\beta\|^2 - e^T\beta = \frac{1}{2}\|P\alpha\|^2 - e^T\alpha$, Where α is the vector defined by appending zeros to β as in Lemma 3. We now show that each entry to Step 2 has an associated objective value that is strictly lower than the one at previous entry to Step 2, and, furthermore, that the number of consecutive executions of Step 3 between Step 2 executions is finite.

By Lemma 5 (in case Step 2.1 is taken) and Lemma 6 (in case Step 2.2 is taken), the above statement is clearly true if there are no intermediate Step 3 executions. So, suppose Step 3 is entered after some execution of Step 2 which adds an index h_0 to J . By Lemma 5 and Lemma 6 we have at this first entry to Step 3 that the hypotheses (10) of Lemma 7 are satisfied. From Lemma 7, for all subsequent consecutive executions of Step 3 we have $\gamma_{h_0} > 0$ and some index $k \neq h_0$ is removed from J .

Therefore, J never becomes empty, the number of such consecutive Step 3 executions is finite, and furthermore, by Lemma 7 this sequence must terminate with (11) being satisfied. Now (11) implies a return to Step 2 with strictly improved objective value. Now, since the number of possible index sets J is finite and all such sets corresponding to entries of Step 2 must be distinct (due to their different corresponding objective values), the algorithm is finite. ■

III. SIMPLIFYING THE LINEAR SYSTEM, UPDATING, AND INITIALIZATION

A. Simplifying the Linear System

Suppose that assumption (3) holds so, there exists a vector $\hat{\beta}$ such that $Z\hat{\beta} = 0$ and suppose that $\begin{bmatrix} Z \\ S \end{bmatrix}$ has d columns. Let Q be an $(m+1)\times d$ matrix with orthonormal columns and R be a $d \times d$ upper triangular matrix such that
$\begin{bmatrix} Z \\ S \end{bmatrix} = QR$. This orthogonal factorization (see, e.g., Watkins [12]) is possible because of (3). Partition Q to form
$Q = \begin{bmatrix} Q_z \\ Q_s \end{bmatrix}$. Since Q has orthonormal columns, we have $Z^TZ + S^TS = R^TR$. Define the vector g by $g = R^{-T}e$ and the scalars f_1 and f_2 by $f_1 = Q_z g$ and $f_2 = Q_z Q_z^T$. Then by some algebra, we can show that

$$\mu = \frac{f_1}{f_2} \quad \text{and} \quad R\beta = g - \mu Q_z^T \tag{12}$$

So, when g , f_1 and f_2 are known, μ can be computed as in (12) and β can be obtained by solving the upper triangular system in (12).

B. Dependency Test and Calculation of $\bar{r}$

Lemma 8: Suppose that assumption (3) holds and the factors Q and R of $\begin{bmatrix} Z \\ S \end{bmatrix}$ are known. Let $\begin{bmatrix} y_h \\ P_h \end{bmatrix}$ be a column to be augmented to $\begin{bmatrix} Z \\ S \end{bmatrix}$. Let $u = \begin{bmatrix} y_h \\ P_h \end{bmatrix} - QQ^T\begin{bmatrix} y_h \\ P_h \end{bmatrix}$ and $r = Q^T\begin{bmatrix} y_h \\ P_h \end{bmatrix}$. Let $\bar{r}$ be the solution to the triangular system $R\bar{r} = r$. Then $\|u\| = 0$ if and only if $\begin{bmatrix} Z \\ S \end{bmatrix}\bar{r} = \begin{bmatrix} y_h \\ P_h \end{bmatrix}$ ■

C. Updating

In the augmentation case, when $\begin{bmatrix} Z \\ S \end{bmatrix}$ is replaced at Step 2 of the algorithm by $\begin{bmatrix} Z & y_h \\ S & P_h \end{bmatrix}$, having u and r as defined in Lemma 8, we replace R by $\begin{bmatrix} R & r \\ 0 & \rho \end{bmatrix}$ and Q by $[Q \;\; q] = \begin{bmatrix} Q_z & q_z \\ Q_s & q_s \end{bmatrix}$, where $\rho = \|u\|$ and $q = \frac{u}{\rho}$.

Observe that

$$\begin{bmatrix} R & r \\ 0 & \rho \end{bmatrix}^{-T} e = \begin{bmatrix} R^{-T} & 0 \\ -r^T R^{-T}/\rho & 1/\rho \end{bmatrix} e = \begin{bmatrix} R^{-T} \\ (1 - rR^{-T}e)/\rho \end{bmatrix} e \; .$$

So, replace g by $\begin{bmatrix} g \\ g^+ \end{bmatrix}$, where $g^+ = (1 - r^T g)/\rho$.

Since $[Q_z \;\; q_z]\begin{bmatrix} g \\ g^+ \end{bmatrix} = Q_z g + q_z g^+ = f_1 + q_z g^+$, replace f_1 by $f_1 + q_z g^+$.

Since $[Q_z \;\; q_z]\begin{bmatrix} Q_z^T \\ q_z \end{bmatrix} = Q_z Q_z^T + q_z^2 = f_2 + q_z^2$, replace f_2 by $f_2 + q_z^2$.

Now, suppose that column k is deleted from $\begin{bmatrix} Z \\ S \end{bmatrix}$ at Step 2.2 or Step 3 of part 1 of the algorithm. Let $\hat{R}$ be the submatrix of R obtained by deleting R 's kth column, R_k , and let $\hat{Z}$ be the corresponding submatrix of Z .
Let H be a permutation matrix that reorders the columns of R so that $RH = [\hat{R} \;\; R_k]$. Let G be the product of Givens matrices required to reduce the upper Hessenberg matrix $\hat{R}$ to an upper triangular matrix $\bar{R}$ over an $(d-1)$ - dimensional zero vector (see, e.g., Watkins [12]).

Let

$$G\left[\hat{R} \quad R_k \quad Q_z^T \quad Q_s^T \quad g\right]=\begin{bmatrix} \overline{R} & r & \overline{Q}_z^T & \overline{Q}_s^T & \overline{g} \\ 0 & \rho & \overline{q}_z^T & \overline{q}_s^T & g^- \end{bmatrix}.$$

Replace R by $\overline{R}$, Q by $\overline{Q}=\begin{bmatrix} \overline{Q}_z \\ \overline{Q}_s \end{bmatrix}$, g by $\overline{g}$, f_1 by $f_1-\overline{q}_z g^-$ and f_2 by $f_2-\overline{q}_z^2$.

D. *Initialization*

Lemma 9: If $\begin{bmatrix} y_1 & y_h \\ P_1 & P_h \end{bmatrix}$ does not have full column rank for some $h\in\{2,3,\cdots,n\}$ and for which $y_h=-y_1$ then (2) is unbounded from below and hence (1) is infeasible. ■

In view of the above Lemma, a starting matrix is given by $\begin{bmatrix} y_1 & y_h \\ P_1 & P_h \end{bmatrix}$, where $\begin{bmatrix} y_h \\ P_h \end{bmatrix}$, $h\in\{2,3,\cdots,n\}$ is the first column that satisfies $y_h=-y_1$. For such a matrix let $R=\begin{bmatrix} \eta & r \\ 0 & \rho \end{bmatrix}$ and $Q=[q \quad u/\rho]$ where $\eta=\left\|\begin{bmatrix} y_1 \\ P_1 \end{bmatrix}\right\|$,

$$q=\frac{1}{\eta}\begin{bmatrix} y_1 \\ P_1 \end{bmatrix},\ r=q^T\begin{bmatrix} y_h \\ P_h \end{bmatrix},\ u=\begin{bmatrix} y_h \\ P_h \end{bmatrix}-qq^T\begin{bmatrix} y_h \\ P_h \end{bmatrix}$$

and $\rho=\|u\|$. It can be easily shown that the solution β of the corresponding linear system satisfies $\beta_1=\beta_h>0$.

IV. A COMPUTATIONAL EXPERIMENT

In order to test the validity of our algorithm, we have written a prototype MATLAB code. For our run, we have used the well-known Iris flower data set (see, e.g., [13]). The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica, and Iris versicolor). Four features were measured for each sample which are the lengths and the widths of the sepal and petal. The first class is linearly separable from the other two. The latter two classes are not linearly separable from each other. To create a binary linearly separable classification problem, we have chosen the two classes setosa and versicolour. The experiment reveals that our algorithm finds the optimal solution quite accurately and has worked successfully in both the training and the testing stages. Also, when we picked the two non-separable classes versicolour and virginica, the message "unbounded" was given where this message indicates that the classification problem is not separable.

In order to obtain a visualization of a 2-dimentional classification problem, we have picked from the same data set the two features sepal width and petal width. Fig.1 shows a plot of the two classes along with the optimal separating line. And in order to obtain a visualization in 3-dimentions, we have picked the three features sepal length, sepal width and petal width. Fig. 2 shows a plot of the two classes as well as the optimal separating plane. Finally, Fig. 3 shows a plot in 2-dimentions of the two non-separable classes versicolor and virginica where we have picked the two features sepal length and petal length.

V. CONCLUSION

We have presented an algorithm for solving linearly separable SVMs by taking advantage of the factorized form of the Hessian matrix in the dual problem as a product of a matrix and its transpose so this does not require the user to compute n^2 dot products to form P^TP. Also, since the most difficult step is in the linear system solution at each iteration, our method relies on orthogonal factorizations which lead to solving upper triangular systems. We have also directed our attention to simplifying the upper triangular system as well as simplifying the updating of the orthogonal factorizations. We have conducted an experiment where we have implemented our method in MATLAB just to test the validity of our algorithm.

A natural extension of the algorithm is to the case of a soft margin where values of a regularization parameter C are introduced so, our dual problem will be box constrained ($\alpha\geq 0$ will be replaced with $0\leq\alpha_j\leq C$ for $j=1,...,n$). A computational efficiency, as

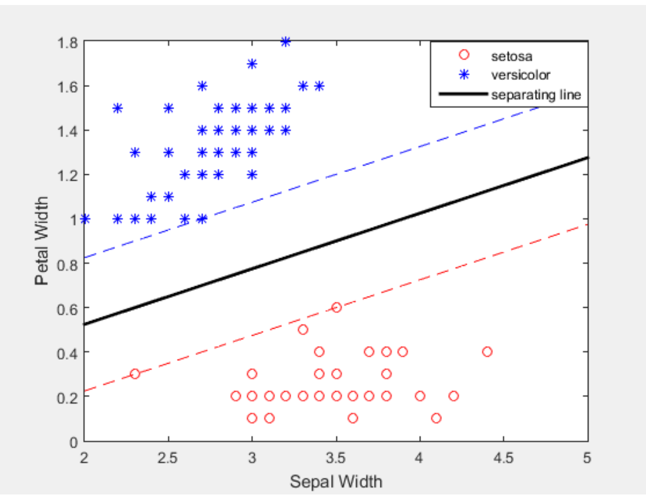

Fig. 1. Separating Line for the Iris Flower Data Set

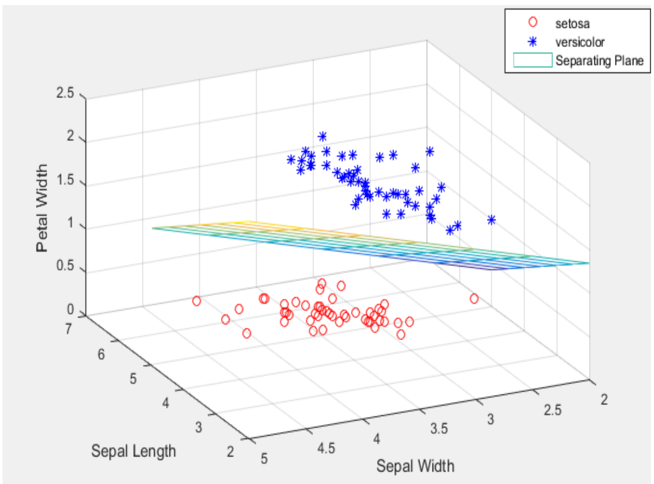

Fig. 2. Separating Plane for the Iris Flower Data Set

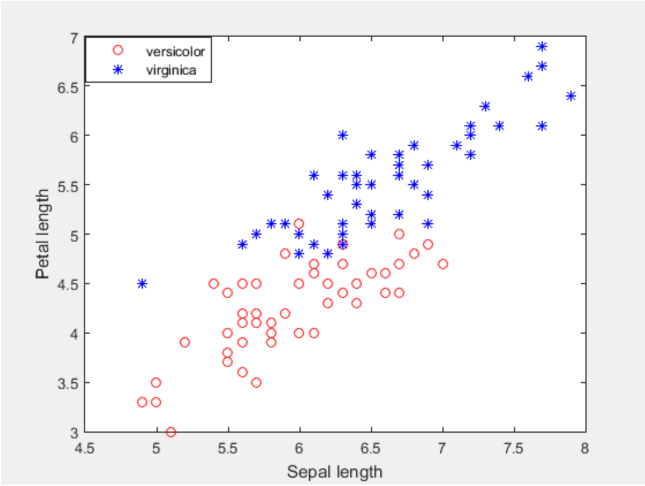

Fig. 3. Non-Separable Classes

we anticipate, will be gained first by not having to compute the whole matrix $P^T P$, and secondly by simplifying the upper triangular system to be solved in each iteration. This will be tested in the future when we extend our method and compare it with other existing algorithms.

We conjecture that continuing the development of the algorithm will lead to a highly competitive SVM solver. This may be a promising direction of future work.

Remark: We would like to note again that for the sake of brevity some of the proofs and results were omitted. We refer the reader to [1] for similar proofs and results.

REFERENCES

[1] A. Al-Saket, An Algorithm with Degeneracy Resolution for Solving Certain Quadratic Programming Problems, Ph.D. thesis, Washington State University, United States, 1985.

[2] C. J. C. Burges, A tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121-167, 1998.

[3] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-based Learning Methods, Cambridge University Press, 2000.

[4] A. Shilton, M. Palaniswami, D. Ralph and A. C. Tosoi, Incremental Training of Support Vector Machines, IEEE Transactions on Neural Networks, vol. 16, no. 1, pp. 114-131, 2005.

[5] C. J. Org, S. Shao and J. Yang, An Improved Algorithm for the Solution of Regulization Path of Support Vector Machine, IEEE Transactions on Neural Networks, vol. 21, no. 3, pp. 451-462, 2010.

[6] P. E. Gill and W. Murray, Numerically stable methods for quadratic programming, mathematical Programming, vol. 14, pp. 349-372, 1978.

[7] J. Nocedal and S. J. Wright, Nunerical Optimization, 2nd Ed., Springer, 2006.

[8] D. Luenberger and Y. Ye, Linear and Nonlinear Programming, 4th Ed., Springer, 2016.

[9] J. A. Snyman and D. N. Wilke, Practical Mathematical Optimization, Springer, 2018.

[10] A. Bemporad, A Quadratic Programming Algorithm Based on Nonnegative Least Squares with Applications to Embedded Model Predictive Control, IEEE Trans. Automat. Control, vol. 61, no. 4, pp. 1111–1116, 2016.

[11] A. Forsgren, P. Gill and E. Wong, Primal and dual active-set methods for convex quadratic programming, mathematical Programming , vol. 159, pp.469-508, 2016.

[12] D. Watkins, Fundamentals of Matrix Computations, 3rd Ed., John Wiley, 2010.

[13] I. A. Taha and J. Ghosh, Symbolic Interpretation of Artificial Neural Networks, IEEE Trans. Knowl. Data Eng., vol. 11, no. 3, pp. 448-463, 1999.