Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-30-2003

A new method to select working sets for Decomposition Methods solving Support Vector Machines

D.Lai, N.Mani and M.Palaniswami

MONASH
UNIVERSITY

A new method to select working sets for Decomposition Methods solving Support Vector Machines

*D.Lai , +M.Palaniswami ,*N.Mani
*Dept. of Electrical and Computer Systems Engineering
Monash University, Clayton, Vic. 3168, Australia.
+Dept. of Electrical and Electronic Engineering,
The University of Melbourne, Vic. 3010, Australia.
{daniel.lai,n.mani@eng.monash.edu.au,swami@ee.mu.oz.au}

Abstract: In this report, we propose a iteration measure to determine a better working set choice for the decomposition method. The decomposition method generally solves a sequence of sub problems instead of the entire problem at each iterative step. This makes it an ideal optimization method for solving the Support Vector Machine classifier, which is usually trained on a large dataset using machines with limited processing memory. The rate of convergence of the decomposition algorithm depends largely on the order of the sub problems solved and has been shown to be linear in the worst case. Our iteration measure determines the amount of overstep if the update on an iterate causes it to exit the feasible region. We then choose a working set that minimizes this amount of overstep, there by also increasing the step towards the minimum of the objective function. We report some initial results, which show a good improvement to the rates of convergence for our benchmark datasets.

I. INTRODUCTION

The Support Vector Machines (SVM) developed by Vapnik[1] and co-workers has been shown to be a powerful supervised learning tool. The standard soft-margin Support Vector Machine is a binary classifier applied to classify a data set defined as,

$$\begin{aligned} &\Theta=\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)...(\mathbf{x}_l, y_l)\} \\ &\mathbf{x}_i \in \mathbb{R}^n \\ &y_i = \{1,-1\} \end{aligned} \tag{1}$$

The SVM formulation is essentially a regularized minimization problem leading to the use of Lagrange Theory and quadratic programming techniques. The formulation defines a boundary separating two classes in the form of a linear hyperplane in data space where the distance between the boundaries of the two classes and the hyperplane is known as the margin of the hyperplane. This idea is further extended for data that is not linearly separable; where it is first mapped via a nonlinear function to a higher dimension feature space. Maximizing the margin of the hyperplane in either space is equivalent to maximizing the distance between the class boundaries. Vapnik[1] suggests that the form of the hyperplane, $f(x) \in F$ be chosen from family of functions with sufficient capacity. In particular, F contains functions for the linearly and non-linearly separable hyperplanes;

$$f(x) = \sum_{i=1}^{l} w_i x_i + b \tag{2}$$

$$f(x) = \sum_{i=1}^{l} w_i \phi(x_i) + b \tag{3}$$

The weight vector, **w** in (3) is no longer the same expansion as in the linearly separable case (2). In fact, the non-linear mapping $\phi : \mathbf{x} \subset \Re^n \to \Re^m$ and $n, m \in [1, \infty)$ defines the mapping from data space to feature space. Hence the weights in feature space will have a one to one correspondence with the elements of $\phi(\mathbf{x})$. Now for separation in feature space, we would like to obtain the hyperplane with the following properties;

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{l} w_i \varphi(\mathbf{x}_i) + b \\ f(\mathbf{x}) &> 0 \quad \forall \quad i : y_i = +1 \\ f(\mathbf{x}) &< 0 \quad \forall \quad i : y_i = -1 \end{aligned} \tag{4}$$

The conditions in (4) can be described by a strict linear discriminant function, so that for each element pair in Θ, we require that;

$$y_i \left(\sum_{i=1}^{l} w_i \varphi(\mathbf{x}_i) + b \right) \geq 1 \tag{5}$$

The distance from the hyperplane to a support vector is $\frac{1}{\|\mathbf{w}\|}$ and the distance between the support vectors of one class to the other class is simply $\frac{2}{\|\mathbf{w}\|}$ by geometry. The soft-margin minimization problem relaxes the strict discriminant in (5) by introducing slack variables, ξ_i and is formulated as;

$$\text{P.1} \begin{cases} \text{minimize} \quad \Im(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{l} w_i^2 + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} \begin{cases} y_i \left(\sum_{i=1}^{l} w_i \varphi(\mathbf{x}_i) + b \right) \geq 1 + \xi_i \\ \forall i = 1..l \end{cases} \end{cases} \tag{6}$$

We now apply Lagrange Theory to solve (6) giving us the Lagrange Primal problem of the following form;

$$\text{P.2} \begin{cases} \text{minimize} \quad \Im(\mathrm{w}, \alpha) = \frac{1}{2} \sum_{i=1}^{l} w_i^2 + \sum_{i=1}^{l} \alpha_i \left(y_i \left(\sum_{j=1}^{l} w_j \varphi(x_j) + b \right) - 1 - \xi_i \right) + \sum_{i=1}^{l} \pi_i \xi_i \\ \text{subject to} \begin{cases} \nabla \left(\frac{1}{2} \sum_{i=1}^{l} w_i^2 \right) + \nabla \left(\sum_{i=1}^{l} \alpha_i \left(y_i \left(\sum_{j=1}^{l} w_j \varphi(x_j) + b \right) - 1 - \xi_i \right) \right) = 0 \\ \alpha_i, \pi_i \geq 0 \quad \forall \mathrm{i}=1..l \end{cases} \end{cases} \tag{7}$$

The following dual optimization of a cost function written in terms of Lagrange Multipliers alone is usually implemented by incorporating the gradient condition into the cost function of P.2 and minimizing the following Lagrange dual in terms of Lagrange multipliers, α alone;

$$\text{P.3} \begin{cases} \text{minimize} \quad \Im(\boldsymbol{\alpha}) = -\sum_{\mathrm{i}=1}^{l} \alpha_i + \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to} \quad \begin{cases} 0 \leq \alpha_{\mathrm{i}} \leq C \\ \sum_{i=1}^{l} \alpha_{\mathrm{i}} y_i = 0 \end{cases} \end{cases} \tag{8}$$

The separating hyperplane surface in (4) can now be written in terms of Lagrange Multipliers;

$$f(x) = sign \left(\sum_{i=1}^{l} \alpha_i y_i K(x, x_i) + b \right) \tag{9}$$

Recent work has shown that variations of the original formulation could be solved using linear programming[2, 3], but we shall focus on the quadratic program here. Several optimization algorithms are documented in [2] and they are often

modified versions of classical optimization methods which have been tailored to solve the Support Vector Machine problem. A standard fast iterative method is the Newton method that could be applied to solve the constrained optimization problem. Unfortunately, Newton-methods are not practical for large datasets since they require the computation of an inverse derivative, which needs to be stored in memory. Even with today's current memory standards, 512MB of RAM is only capable of storing 128 million floating-point numbers, which translates to approximately a 10000 x 10000 matrix.

The decomposition method is one possible solution for situations where processing memory is limited. Instead of solving the entire problem at each iteration (i.e. updating all n variables for an n-dimensioned problem), the method decomposes the problem into smaller sub problems and solves them sequentially. The advantage of this is that, we require only a fraction of the data to solve the sub problems at each iterative step making it possible to store the rest of the data on disk and to bring it into memory only when required. One such decomposition algorithm is the Sequential Minimal Optimization (SMO) introduced by Platt[4] which is one of the more efficient training and easy to implement algorithms around. SMO is a Newtonian decomposition type algorithm that solves sub-problems of two variables at any one time. However, it has been pointed out that such decomposition algorithms have somewhat slower rates of convergence compared to the quadratic convergence rates enjoyed by Newton methods[5-7]. Hsu [8], points out that this is due to the choice of sub-problems to solve which up to now has been rather arbitrary. If solving a sequence of sub problems makes poor improvement to the objective function, the rates of convergence naturally slow down. It is clear that the choice of sub problems to solve is central to the success of the decomposition method. Unfortunately, determining the optimal sub-problems is rather tricky since the algorithm only considers a subset of variables at any one time [8] and so far, the methods for solving SVMs do not apply any measure of iteration progress to evaluate the effectiveness of solving a particular sub problem. We believe however that some effort should be spent in finding a measure to determine the best sub problem to solve; one which when solved gives the greatest improvement to the objective function at each step. This can be done for a fixed sub problem size, by considering all the possible combinations of variables which is obviously going to be very computationally inefficient. Much of the current choices have been trial and error to find the better combination of variables to solve. We refer the reader to some empirical results and algorithm implementations of SVM decomposition type algorithms [4, 9-11].

In this report, we address the problem of choosing optimal sub problems to solve so that the rate of convergence of the decomposition method is increased. We first introduce our idea of iteration momentum which takes into account the distance which the iterates cross the constraint boundaries when the update rule takes them outside the feasible region. We then investigate the effect of minimizing momentum and show empirically that we can achieve faster convergence rates by doing so. We apply our idea to a Newtonian decomposition method which uses an SMO update rule. We note here that several modern implementations of SMO and SVMlight such as [10] and LibSVM[12] are currently much faster than the original pseudocodes. However, the newer implementations run a basic combination of the original update rules using more efficient coding. Our work here focuses on a method to choose optimal working set pairs that give the best decrease towards the minimum of the cost function. We could further apply the results here to the modern implementations to increase their convergences rates further.

The outline of this paper is as follows; in the next section, we review the Sequential Minimal Optimization algorithm and reformulate it in the familiar Newton update on a closed feasible region. In Section 3, we introduce the notion of momentum in the decomposition setting and investigate momentum minimization in the context of the Newton form. We show here that determining the actual momentum at each iterative step is computationally expensive, but we could predict choices of variables which would have near optimal momentum values. In our initial analysis here, we propose a heuristic set to approximate the momentum minimization idea. The remaining sections will be left for some initial results and further discussion.

II. The Newtonian Decomposition method for Support Vector Machines

A. *Overview of the general Newton Method*

We first review the Newton method briefly by defining the general minimization problem of a cost function subject to a set of constraints;

Definition II.1: General Minimization Problem with Constraints

$$\begin{aligned} &\text{Given a cost(objective) function } \Im : \alpha \subset \Re^n \to \Re \\ &\text{find } \alpha^* \in \mathbb{Q} \text{ so that } \Im(\alpha^*) = \inf_{\alpha \in \mathbb{Q}} \left(\Im(\alpha) \right) \end{aligned} \tag{10}$$

Geometrically, the set $\mathbb{Q}$ is a polytope formed by the constraint set on the minimization problem and usually referred to as the feasible region since the solution vector, α^* must lie in this region. The minimization problem can easily be converted to a maximization problem by optimizing the negative of the cost function. •

The parallel chord method[5] for a multivariable function is the following update rule;

$$\alpha^{t+1} = \alpha^t - M^{-1}\Im\left(\alpha^t\right) \tag{11}$$

The simplest matrix **M** is a diagonal matrix with elements, m_{ii} which actually solves n-functions in 1 dimension. Newton's method uses the derivative of the function $\Im$ as **M** in order to increase the step towards the minimum. We now have the well-known Newton method[13, 14] update written as,

$$\alpha^{t+1} = \alpha^t - \nabla\Im\left(\alpha^t\right)^{-1}\Im\left(\alpha^t\right) \tag{12}$$

We note however that the Newtonian method requires the calculation of the inverse derivative of the cost function, which requires a substantial amount of memory to store for large datasets. This has led to the use of decomposition methods, which break the main problem down into a series of smaller problems that can be loaded into memory as required. We give a more formal definition of the decomposition method and then review a Newtonian Decomposition algorithm.

Definition II.2: The Generalized Decomposition Method

Given the following problem;

$$\begin{aligned} \min/\max \quad & \Im(\mathbf{U}) \\ & \mathbf{U} \in \mathbb{Q}, \mathbb{Q} \in \Re^N \end{aligned} \tag{13}$$

The decomposition method solves the problem by decomposing the objective function $\Im(\mathbf{U})$ into m-sub-problems $\Im_1(\mathbf{u}_1), \Im_2(\mathbf{u}_2) \ldots \Im_m(\mathbf{u}_m)$ and solving each sub-problem independently. We refer to the subspaces $u_k \in \mathbb{Q}_k, \forall k = 1..m$ as the sub domains or working sets of $\Im(\mathbf{U})$ •

Definition II.2: The Newtonian Decomposition Method

Let the sequence of working sets; $\{\mathbf{u}_i\}$ be presented to the decomposition algorithm. Then the Newtonian Decomposition method applies the Newtonian update rule to each element of the sub domain $\mathbf{u}_i$ so that $\forall t > 0, i = 1..l$;

$$\mathbf{u}_i^{t+1} = \mathbf{u}_i^t - \frac{\Im(\mathbf{u})}{\nabla\Im(\mathbf{u})}$$

when solving the sequence of m-sub-problems $\Im_1(\mathbf{u}_1), \Im_2(\mathbf{u}_2) \ldots \Im_m(\mathbf{u}_m)$ •

The one main concern with the decomposition method is that solving sub problems might cause oscillations between locally optimal solutions thus making the convergence of a general decomposition method not easy to prove. Nevertheless, in the case of Support Vector Machines, several researchers have attempted to prove convergence for specific decomposition algorithms under several assumptions[15-18]. Perhaps this is easier to do since from (8), the feasible region for solutions, $\mathbb{Q}$ is in a convex polyhedral defined by;

$$\mathbb{Q} = \left\{ \alpha \left| \sum_{i=1}^{n} \alpha_i y_i, \quad 0 \le \alpha_i \le C \quad \forall i \right. \right\}.$$

In actual fact, we have shown that the optimal solution vector lies on a hyperplane bounded by a hypercube [19]. The convexity of the feasible region then guarantees a global unique solution and possibly reduces the chances of oscillations between local minima (**NB**: There is no local minima, but computational precision could cause "local" minima around the optimal solution and cause unnecessary iterations even though the optimal solution has already been

obtained). We believe that as long as the step is towards the minimum of the cost function, the SVM decomposition algorithms should converge.

B. SMO: A Newtonian Decomposition Method

We review the update rule used in SMO[4] which we show to be a hybrid element-wise Newton method in a decomposition algorithm setting. For the sake of simplicity, we will adopt the notations used by Platt in the original implementation for this summary. The working set size in SMO is fixed to two in order to enforce the equality constraint in (8) at every step of the iteration. We denote the working set as;

$$\alpha_W = \{\alpha_1, \alpha_2\} \quad \alpha_1, \alpha_2 \in \Re \tag{14}$$

Whenever we update the value of two multipliers, SMO maintains the equality constraint by ensuring;

$$\begin{aligned} &\alpha_1 y_1 + \alpha_2 y_2 = k \\ &\text{where} \quad \begin{cases} \alpha_1 + \alpha_2 = k & \text{if } y_1 = y_2 \\ \alpha_1 - \alpha_2 = k & \text{if } y_1 \neq y_2 \end{cases} \end{aligned} \tag{15}$$

We make use of (15) to express the following for some iterative step, t where $t > 0$;

$$\begin{aligned} &s = y_1 y_2 \\ &\alpha_1^{t+1} + s\alpha_2^{t+1} = \alpha_1^{\,t} + s\alpha_2^{\,t} = \gamma \end{aligned} \tag{16}$$

Platt defines the error of a training point as E_i where,

$$E_i = f(x_i) - y_i \quad \forall i = 1..l \tag{17}$$

The minimum of the cost function (8) subject to the equality constraint is first found in the direction of an arbitrary α_2 while the other multiplier α_1 is set to a value that maintains the equality constraint. We can derive the update rule for SMO by writing the cost function in terms of the working set members;

$$\Im(\alpha_1, \alpha_2) = -\alpha_1 - \alpha_2 + \frac{1}{2}K_{11}\alpha_1^{\,2} + \frac{1}{2}K_{22}\alpha_2^{\,2} + sK_{12}\alpha_1\alpha_2 + y_1\alpha_1 v_1 + y_2\alpha_2 v_2 + \Im_{const} \tag{18}$$

$$\text{where} \quad v_i = \sum_{j=3}^{l} y_j \alpha_j^{\,t} K_{ij} = f(x_1) - y_1\alpha_1^{\,t}K_{1i} - y_1\alpha_2^{\,t}K_{2i}$$

$$\Im_{const} = -\sum_{i=3}^{l}\alpha_i + \frac{1}{2}\sum_{i,j=3}^{l}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

Then, the objective function expressed in α_2 entirely is,

$$\begin{aligned} \Im(\alpha_2) = &-\gamma + s\alpha_2 - \alpha_2 + \frac{1}{2}K_{11}(\gamma - s\alpha_2)^2 + \frac{1}{2}K_{22}\alpha_2^{\,2} + sK_{12}(\gamma - s\alpha_2)\alpha_2 \\ &+ y_1(\gamma - s\alpha_2)v_1 + y_2\alpha_2 v_2 + \Im_{const} \end{aligned} \tag{19}$$

The minimum of (19) can be found with respect to α_2, by taking the derivative with respect to the multiplier;

$$\nabla\Im(\alpha_2) = sK_{11}(\gamma - s\alpha_2) - K_{22}\alpha_2 + K_{12}\alpha_2 - sK_{12}(\gamma - s\alpha_2) - y_2 v_2 + y_2 v_1 - s + 1 = 0 \tag{20}$$

Expanding and substituting for s and γ, we get

$$\alpha_2^{t+1}(K_{11} + K_{22} - 2K_{12}) = \alpha_2^{t}(K_{11} + K_{22} - 2K_{12}) + y_2(f(x_1) - f(x_2) + y_2 - y_1) \tag{21}$$

Finally, after some rearranging we obtain the update rule,

$$\alpha_2^{t+1} = \alpha_2^t + y_2 \frac{(E_1 - E_2)}{(K_{11} + K_{22} - 2K_{12})} \tag{22}$$

In order to ensure that $\alpha_1 \in \mathbb{Q}$, we enforce the following;

$$\alpha_2^{t+1} = \begin{cases} UB & if \ \alpha_2^{t+1} \geq UB \\ \alpha_2^{t+1} & if \ \ LB < \alpha_2^{t+1} < UB \\ LB & \alpha_2^{t+1} \leq LB \end{cases} \tag{23}$$

where;

$$\begin{aligned} UB &= \begin{cases} \min(C, \alpha_1 + \alpha_2) & \text{if } \ y_1 = y_2 \\ \min(C, C + \alpha_2 - \alpha_1) & \text{if } \ y_1 \neq y_2 \end{cases} \\ LB &= \begin{cases} \max(0, \alpha_1 + \alpha_2 - C) & \text{if } \ y_1 = y_2 \\ \max(0, \alpha_2 - \alpha_1) & \text{if } \ y_1 \neq y_2 \end{cases} \end{aligned} \tag{24}$$

The update for α_1 is then found by using (16);

$$\alpha_1^{t+1} = \alpha_1^t + s(\alpha_2^t - \alpha_2^{t+1,bounded}) \tag{25}$$

It can be seen here that the equality constraint is enforced at each step of the iteration through the use of (25). From (22), the Newton form is not immediately obvious, so we write the derivatives of the objective function (8) as;

$$\nabla \Im(\alpha_2) = y_2 (E_1 - E_2) \tag{26}$$

$$\nabla (\nabla \Im(\alpha_2)) = 2K_{12} - K_{22} - K_{11} \tag{27}$$

Then the update rule (22) can be written as;

$$\alpha_2^{t+1} = \alpha_2^t + y_2 \frac{(E_1 - E_2)}{(K_{11} + K_{22} - 2K_{12})} = \alpha_2^t - \frac{\nabla \Im(\alpha_2)}{\nabla (\nabla \Im(\alpha_2))} \tag{28}$$

Similarly, we find that the update rule for α_1 is also Newton;

$$\alpha_1^{t+1} = \alpha_1^t - \frac{\nabla \Im(\alpha_1)}{\nabla (\nabla \Im(\alpha_1))} \tag{29}$$

We note from (10) that $\alpha_1, \alpha_2 \in \mathbb{Q}$ implies that the multipliers are bounded at each iterative stage to ensure that they remain in the feasible region. Therefore, depending on the choice of $\mathbb{Q}$, the updates are optimal in the direction of the two multipliers on the feasible region. We recover (12) by setting $F(\alpha^t) = \nabla \Im(\alpha^t)$ and obtain the element-wise Newton method. We can easily see now from Definition II.1 that the SMO update rule actually solves the minimization of the derivative of the cost function (8) which is similar to Joachim[11]. The details of the derivation above are included in the appendix.

III. Iteration Momentum and The Potential Step Size

A. *The Idea of Momentum*

It can be seen that (22) and (29) leave the choice of the next working set variables arbitrary as before, that is we can pick any two variables to solve. The problem with this is that we are only assured the updates of the two variables are optimal in the Newtonian sense, but we do not know how optimal they were on the entire variable space. In the Newtonian case, the step is always an improvement as long as we start somewhere in the vicinity of the optimal solution. We would like some sense of measure to guide us during the iteration process so that at least our arbitrary choice of working set is "optimal" in some way. In this way, we could thus increase the rates of convergence of the decomposition method and reduce the training times for the Support Vector Machine.

We now introduce the quantity, ΔR to model the effect of (23) when restricting the update in (22) to the feasible region $\mathbb{Q}$. We loosely refer to quantity ΔR as the "momentum" of the iterative step but it is a direct measure of the amount of restriction imposed if the updates on the variables cause them to exit the region $\mathbb{Q}$. In doing so, we now have a measure of how "hard" an arbitrary iterative step causes the iterates to hit a constraint boundary. We first derive the necessary equations and then attempt to give a geometrical interpretation of our momentum quantity.

Let us first rewrite the update rules for two arbitrary Lagrange Multipliers α_1 and α_2 by taking into consideration our proposed iteration momentum.

For an arbitrary α_2 we have;

$$\alpha_2^{t+1} = \alpha_2^{t} - \frac{y_2(E_1 - E_2)}{(2K_{12} - K_{11} - K_{22})} - \Delta R \tag{30}$$

Then we have for α_1;

$$\begin{aligned} \alpha_1^{t+1} &= \alpha_1^{t} + s(\alpha_2^{t} - \alpha_2^{t+1}) \\ &= \alpha_1^{t} + s(\alpha_2^{t} - \left(\alpha_2^{t} - \frac{\nabla \Im(\alpha_2)}{\nabla\left(\nabla \Im(\alpha_2)\right)} - \Delta R \right)) \\ &= \alpha_1^{t} + s(\frac{\nabla \Im(\alpha_2)}{\nabla\left(\nabla \Im(\alpha_2)\right)} + \Delta R) = \alpha_1^{t} - \frac{y_1(E_2 - E_1)}{(2K_{12} - K_{11} - K_{22})} + \Delta R \end{aligned} \tag{31}$$

We have the following updates on the working set variables written in element-wise vector notation;

$$\alpha_w^{t+1} = \begin{pmatrix} \alpha_1^{t+1} \\ \alpha_2^{t+1} \end{pmatrix} = \begin{pmatrix} \alpha_1^{t} \\ \alpha_2^{t} \end{pmatrix} - \begin{pmatrix} \frac{y_1(E_2 - E_1)}{\eta} - \Delta R \\ \frac{y_2(E_1 - E_2)}{\eta} + \Delta R \end{pmatrix} \tag{32}$$

where;

$$\Delta R \begin{cases} \alpha_1^{t} - \frac{y_1(E_2 - E_1)}{\eta} - UB & \text{if } \alpha_2^{t+1} \geq UB \\ 0 & \text{if } LB < \alpha_2^{t+1} < UB \\ \alpha_1^{t} - \frac{y_1(E_2 - E_1)}{\eta} - LB & \text{if } \alpha_2^{t+1} \leq LB \end{cases} \tag{33}$$

$$\eta = 2K_{12} - K_{11} - K_{22}$$

We now investigate the effect of momentum by computing the change at each iterative step on the value of the objective function. Let us make the following substitutions to simplify the exposition.

$$A = \frac{y_1(E_2 - E_1)}{\eta} - \Delta R \quad , \quad B = \frac{y_2(E_1 - E_2)}{\eta} + \Delta R \tag{34}$$

We then have from (18) the updated cost function in terms of $\alpha_1^{t+1}, \alpha_2^{t+1}$;

$$\begin{aligned} \Im(\alpha_1, \alpha_2)^{t+1} = &-\alpha_1^{t} + A - \alpha_2^{t} + B + \frac{1}{2}K_{11}\left(\alpha_1^{t} - A\right)^2 + \frac{1}{2}K_{22}\left(\alpha_2^{t} - B\right)^2 + y_1 y_2 K_{12}\left(\alpha_1^{t} - A\right)\left(\alpha_2^{t} - B\right) \\ &+ y_1\left(\alpha_1^{t} - A\right)v_1 + y_2\left(\alpha_2^{t} - B\right)v_2 + \Im_{const} \end{aligned} \tag{35}$$

To simplify the exposition, we leave out the superscript *t* and all iterates are assumed to be for the iteration step t. We now proceed to simplify (35),

$$\rightarrow -\alpha_1 - \alpha_2 + \frac{1}{2}\left(K_{11}\alpha_1^2 + K_{22}\alpha_2^2 + 2y_1y_2K_{12}\alpha_1\alpha_2\right) + y_1\alpha_1v_1 + y_2\alpha_2v_2 + \Im_{const}$$
$$+ (A+B) + \frac{1}{2}\left[\left(-2\alpha_1A + A^2\right)K_{11} + \left(-2\alpha_2B + B^2\right)K_{22} + 2\left(-\alpha_1B - \alpha_2A + AB\right)y_1y_2K_{12}\right]$$
$$- Ay_1v_1 - By_2v_2$$
$$\rightarrow \Im\left(\alpha_1,\alpha_2\right) - Ay_1v_1 - By_2v_2 + (A+B)$$
$$+ \frac{1}{2}\left[A^2K_{11} + B^2K_{22} + 2ABy_1y_2K_{12}\right] + \frac{1}{2}\left[-2\alpha_1AK_{11} - 2\alpha_2BK_{22} + 2\alpha_1BK_{12} + 2\alpha_2AK_{12}\right]$$
$$\rightarrow \Im\left(\alpha_1,\alpha_2\right) - A\left(y_1v_1 - 1\right) - B\left(y_2v_2 - 1\right) + \frac{1}{2}\left[A^2y_1y_1K_{11} + B^2y_2y_2K_{22} + 2ABy_1y_2K_{12}\right]$$
$$-\alpha_1AK_{11} - \alpha_2BK_{22} - \alpha_1BK_{12} - \alpha_2AK_{12}$$
$$\rightarrow \Im\left(\alpha_1,\alpha_2\right) - A\left(y_1v_1 + \alpha_1y_1y_1K_{11} + \alpha_2y_2y_2K_{12} - 1\right) - B\left(y_2v_2 + \alpha_2y_2y_2K_{22} + \alpha_1y_1y_1K_{12} - 1\right)$$
$$+ \frac{1}{2}\left[Ay_1\phi\left(x_1\right) + By_2\phi\left(x_2\right)\right]^2$$
$$\rightarrow \Im\left(\alpha_1,\alpha_2\right) - Ay_1\left(f\left(x_1\right) - y_1\right) - By_2\left(f\left(x_2\right) - y_2\right) + \frac{1}{2}\left[Ay_1\phi\left(x_1\right) + By_2\phi\left(x_2\right)\right]^2$$
$$\rightarrow \Im\left(\alpha_1,\alpha_2\right) - Ay_1E_1 - By_2E_2 + \frac{1}{2}\left[Ay_1\phi\left(x_1\right) + By_2\phi\left(x_2\right)\right]^2$$

$$\therefore \Im\left(\alpha_1,\alpha_2\right)^{t+1} = \Im\left(\alpha_1,\alpha_2\right)^t - Ay_1E_1 - By_2E_2 + \frac{1}{2}\left[Ay_1\phi\left(x_1\right) + By_2\phi\left(x_2\right)\right]^2 \quad (36)$$

The step size or step length denotes the improvement to the cost function with respect to the updates at each iterative step. We now have step size for this Newtonian Decomposition rule by writing (36) in the following form;

$$\Im\left(\alpha_1,\alpha_2\right)^{t+1} = \Im\left(\alpha_1,\alpha_2\right)^t + \Delta M_{y_1,y_2} \quad (37)$$

Here, $\Delta M_{y_1,y_2}$ is the step size, which we write as,

$$\Delta M_{y_1,y_2} = -Ay_1E_1 - By_2E_2 + \frac{1}{2}\left[Ay_1\phi\left(x_1\right) + By_2\phi\left(x_2\right)\right]^2 \quad (38)$$

We note here that $\Delta M_{y_1,y_2}$ is derived for the minimization of the cost function $\Im$ and can be similarly derived for the maximization of the cost function $\Im$. Instead of defining norms on the domain space, we can now investigate the improvement at each iteration step in greater detail based on the step size $\Delta M_{y_1,y_2}$.

We proceed with our analysis by further substituting for A and B in (38) and to make it easier to follow, we make the substitution;

$$C = \frac{y_1(E_2 - E_1)}{\eta}, \qquad \mathrm{D} = \frac{y_2(E_1 - E_2)}{\eta}$$

So that;

$$
\begin{aligned}
&\Delta M_{y_1,y_2} \\
&\rightarrow -\left(\frac{y_1(E_2-E_1)}{\eta}-\Delta R\right)y_1E_1-\left(\frac{y_2(E_1-E_2)}{\eta}+\Delta R\right)y_2E_2+\frac{1}{2}\left[\left(C-\Delta R\right)y_1\phi\left(x_1\right)+\left(D+\Delta R\right)y_2\phi\left(x_2\right)\right]^2 \\
&\rightarrow -\left(\frac{(E_2-E_1)E_1}{\eta}+\frac{(E_1-E_2)E_2}{\eta}\right)-\Delta R\left(y_2E_2-y_1E_1\right) \\
&\quad+\frac{1}{2}\left[\left(C-\Delta R\right)^2\phi\left(x_1\right)^2+\left(D+\Delta R\right)^2\phi\left(x_2\right)^2+2\left(C-\Delta R\right)\left(D+\Delta R\right)y_1y_2\phi\left(x_1\right)\phi\left(x_2\right)\right] \\
&\rightarrow \frac{(E_2-E_1)^2}{\eta}-\Delta R\left(y_2E_2-y_1E_1\right)+\frac{1}{2}\left(C^2K_{11}+D^2K_{22}+2CDy_1y_2K_{12}\right) \\
&\quad+\frac{\Delta R^2}{2}\left(K_{11}+K_{22}-2y_1y_2K_{12}\right)+\Delta R\left(C\left(y_1y_2K_{12}-K_{11}\right)+D\left(K_{22}-y_1y_2K_{12}\right)\right) \\
&\rightarrow \frac{(E_2-E_1)^2}{\eta}-\Delta R\left(y_2E_2-y_1E_1\right)-\frac{(E_2-E_1)^2}{2\eta}+\frac{\Delta R^2}{2}\left(y_1\phi\left(x_1\right)-y_2\phi\left(x_2\right)\right)^2 \\
&\quad-\Delta R\frac{(E_2-E_1)}{\eta}\left(y_1K_{11}+y_2K_{12}-\left(y_2K_{22}+y_1K_{12}\right)\right) \\
&\rightarrow \frac{(E_2-E_1)^2}{2\eta}-\Delta R\left(y_2E_2-y_1E_1\right)+\frac{\Delta R^2}{2}\left(y_1\phi\left(x_1\right)-y_2\phi\left(x_2\right)\right)^2-\frac{\Delta R(E_2-E_1)}{\eta}\left(y_2K_{22}+y_1K_{11}-\left(y_2+y_1\right)K_{12}\right)
\end{aligned}
$$

We now have;

$$
\begin{aligned}
\Delta M_{y_1,y_2} &= \frac{(E_2-E_1)^2}{2\eta}-\Delta R\left(y_2E_2-y_1E_1\right)+\frac{\Delta R^2}{2}\left(y_1\phi\left(x_1\right)-y_2\phi\left(x_2\right)\right)^2 \\
&\quad-\frac{\Delta R(E_2-E_1)}{\eta}\left(y_1K_{11}+y_2K_{22}-\left(y_2+y_1\right)K_{12}\right)
\end{aligned}
\qquad (39)
$$

We now have the step size function $\Delta M_{y_1,y_2}$ in terms of momentum, ΔR which models how hard the iterates would hit the constraint boundaries when they are updated using a Newton step. The larger the value of ΔR, the larger the momentum of the iteration or the harder the iterates hit the constraint facet. We note that this general form applies only to an arbitrary working set α_w of size 2. It is also clear that $\Delta M_{y_1,y_2}$ depends on the labels y_1, y_2, hence the choice of our notation for the step size function.

If we set $\Delta R = 0$, we immediately disregard the existence of boundaries i.e. the region $\mathbb{Q}$ is open and unbounded, we have;

$$
\Delta M_p = \frac{(E_2-E_1)^2}{2\eta} \qquad (40)
$$

We refer to the function ΔM_p as the *potential step size* or simply the potential function that actually depicts the true Newtonian step that would be taken if the solution vector lies in the entire $\Re^l$ space. In other words, it is the potential improvement of the objective function if they were no constraints on the variable vector $\boldsymbol{\alpha}$.

Observation 1: Largest Potential Step

We note that for a positive (semi-positive) definite kernel matrix, $\eta < 0$ $(\eta \le 0)$ the potential step is always towards the minimum of the objective function (8) for any arbitrary working set $\boldsymbol{\alpha}_w$. In the case of a constrained region, the largest actual step is approximated in [4] by using the maximal violating pair $\left|E_2 - E_1\right|$ for an arbitrary $\boldsymbol{\alpha}_w$. This has empirically been shown to work well by Platt in the case of the constrained SVM dual in (8). However, in the case of decomposition for a general optimization problem, we do not know yet whether the largest potential step would actually guarantee the best improvement to the objective function in a constrained region. The necessary and sufficient conditions to guarantee this still remains an interesting research topic for us to study further •

In this work, we characterize the step size function (39), according to the labels y_1, y_2 which correspond directly to choice of α_1, α_2 in the working set, α_w. We first list several relationships that would be useful for the following derivations.

$$\eta = 2K_{12} - K_{11} - K_{22} = -\left\langle \left(\phi(x_1) - \phi(x_2)\right), \left(\phi(x_1) - \phi(x_2)\right)\right\rangle = -\left(\phi(x_1) - \phi(x_2)\right)^2 \quad (41)$$

$$K_{22} - K_{11} = \left\langle \left(\phi(x_2) - \phi(x_1)\right), \left(\phi(x_2) + \phi(x_1)\right)\right\rangle = \left(\phi(x_2) - \phi(x_1)\right)\left(\phi(x_2) + \phi(x_1)\right) \quad (42)$$

Case 1: If $y_1 = y_2 = 1$, we then have from (39),

$$\begin{aligned}\Delta M_{1,1} &= \frac{(E_2 - E_1)^2}{2\eta} - \Delta R\left(E_2 - E_1\right) + \frac{\Delta R^2}{2}\left(\phi(x_1) - \phi(x_2)\right)^2 - \frac{\Delta R(E_2 - E_1)}{\eta}\left(K_{22} + K_{11} - 2K_{12}\right) \\ &= \frac{(E_2 - E_1)^2}{2\eta} - \Delta R\left(E_2 - E_1\right) + \frac{\Delta R^2}{2}\left(\phi(x_1) - \phi(x_2)\right)^2 - \frac{\Delta R(E_2 - E_1)}{\eta}\left(-\eta\right) \\ &= \frac{(E_2 - E_1)^2}{2\eta} + \frac{\Delta R^2}{2}\left(\phi(x_1) - \phi(x_2)\right)^2\end{aligned} \quad (43)$$

Case 2: If $y_1 = y_2 = -1$, we then have from (39),

$$\begin{aligned}\Delta M_{-1,-1} &= \frac{(E_2 - E_1)^2}{2\eta} - \Delta R\left(-E_2 + E_1\right) + \frac{\Delta R^2}{2}\left(-\phi(x_1) - \phi(x_2)\right)^2 - \frac{\Delta R(E_2 - E_1)}{\eta}\left(-K_{11} - K_{22} + 2K_{12}\right) \\ &= \frac{(E_2 - E_1)^2}{2\eta} + \Delta R\left(E_2 - E_1\right) + \frac{\Delta R^2}{2}\left(\phi(x_1) + \phi(x_2)\right)^2 - \frac{\Delta R(E_2 - E_1)}{\eta}\left(\eta\right) \\ &= \frac{(E_2 - E_1)^2}{2\eta} + \frac{\Delta R^2}{2}\left(\phi(x_1) + \phi(x_2)\right)^2\end{aligned} \quad (44)$$

Case 3: If $y_1 \neq y_2, y_1 = -1, y_2 = 1$, we then have,

$$\begin{aligned}\Delta M_{-1,1} &= \frac{(E_2 - E_1)^2}{2\eta} - \Delta R\left(E_2 + E_1\right) + \frac{\Delta R^2}{2}\left(-\phi(x_1) - \phi(x_2)\right)^2 - \frac{\Delta R(E_2 - E_1)}{\eta}\left(K_{11} - K_{22}\right) \\ &= \frac{(E_2 - E_1)^2}{2\eta} - \Delta R\left(E_2 + E_1\right) - \frac{\Delta R(E_2 - E_1)}{\eta}\left(K_{22} - K_{11}\right) + \frac{\Delta R^2}{2}\left(\phi(x_1) + \phi(x_2)\right)^2\end{aligned} \quad (45)$$

Case 4: If $y_1 \neq y_2, y_1 = 1, y_2 = -1$, we then have the symmetry,

$$\begin{aligned}\Delta M_{1,-1} &= \frac{(E_2 - E_1)^2}{2\eta} - \Delta R\left(-E_2 - E_1\right) + \frac{\Delta R^2}{2}\left(\phi(x_1) + \phi(x_2)\right)^2 - \frac{\Delta R(E_2 - E_1)}{\eta}\left(K_{11} - K_{22}\right) \\ &= \frac{(E_2 - E_1)^2}{2\eta} + \Delta R\left(E_2 + E_1\right) + \frac{\Delta R(E_2 - E_1)}{\eta}\left(K_{22} - K_{11}\right) + \frac{\Delta R^2}{2}\left(\phi(x_1) + \phi(x_2)\right)^2\end{aligned} \quad (46)$$

If we generalize the cases further, we can see that $y_1 = y_2$ and $y_1 \neq y_2$ give different step sizes and have different momentum values. This seems to prove the existence of an "optimal" choice of working set to solve in order to give the best improvement to the objective function. It also seems to suggest that we could predict which cases would most likely give a better step during the iteration process. For now, we give a geometrical discussion of our momentum idea in the following section.

B. Iteration Momentum in a bounded region

Consider the problem of minimizing an objective function subject to a set of constraints on the problem variables as given in Definition II.1. We can interpret the set of constraints as boundary planes[20] which form a region $\mathbb{Q}$ lying in the general $\Re^l$ space. So, if the problem has *n*-constraints, the region $\mathbb{Q}$ then has *n* sides or facets. The solution to the problem which is a vector of variables $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2 ... \alpha_l\}$ is constrained to lie in this "feasible" region $\mathbb{Q}$. In the case of our Support Vector dual problem (8), each training example contributes a constraint and so we can think of the region $\mathbb{Q}$ as a hyperplane lying in a hypercube for the case where $l > 3$. Now, let us consider a simpler problem where the region $\mathbb{Q}$ has the following shape as in Fig 1. For illustration purposes, the solution vector α^* to this minimization problem is located towards the "bottom" of the feasible region. Suppose we start the algorithm off with an initial guess, I_0 then in the ideal case, we would like the optimization algorithm to follow the path of the thick line (indicated by I_1) so that we would have obtained the solution to the problem in the quickest way. Unfortunately the location of α^* is unknown to the optimization algorithm and a good rule of thumb so far is to follow the strictest path of descent that we have shown as a series of dashed lines.

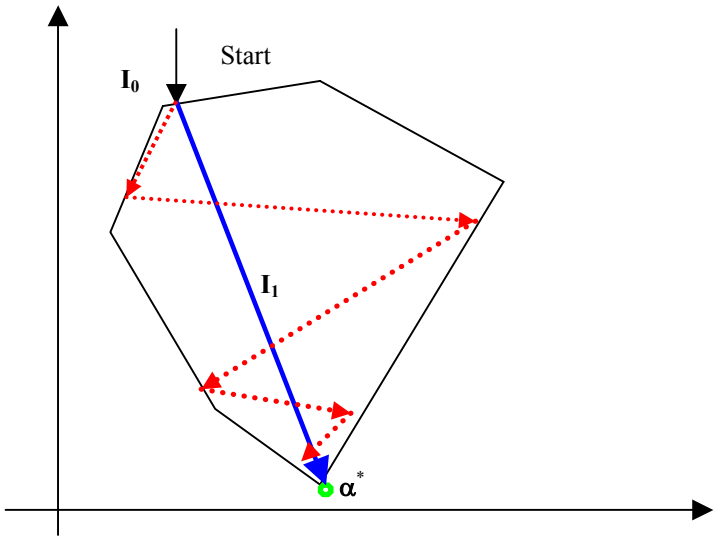

Fig 1: An example of a feasible region Q on the α space for a minimization problem where the solution vector α^* is denoted by a circle.

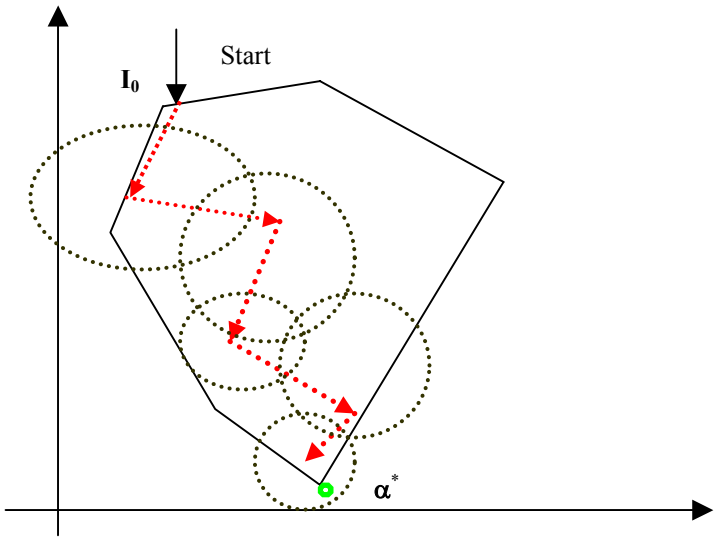

Fig 2: An example of the regions where the potential vector α could lie when only a subset of it is updated through the decomposition method. The figure depicts the case for 5 iterative steps.

In the case of the decomposition method in Fig 2, only a subset of variables (the working set) is updated at each step and not the entire variable vector. Since the choice of working set is arbitrary at each iterative step, we denote the possible locations of the variable vector as circular regions that we refer to as potential regions. The radius of this region is the largest possible step towards the solution vector if the optimal working set was solved. The dashed lines now represent the path taken by the algorithm that has chosen an arbitrary sequence of working sets. We can see now that in the case of the decomposition method, the largest potential step (denoted by the radius of the region) does not guarantee the quickest convergence. From the example diagram, it is possible that there may be more than one choice of working set, which gives the largest step, and so the vector $\boldsymbol{\alpha}$ could possibly lie on the boundary of a potential region and the next update could send us away from the optimal solution. It is clear that the choice of working sets affects how quickly we arrive at the solution vector. A bad choice could result in the algorithm zigzagging in the feasible region Q while a good choice would quickly lead us to the optimal solution vector.

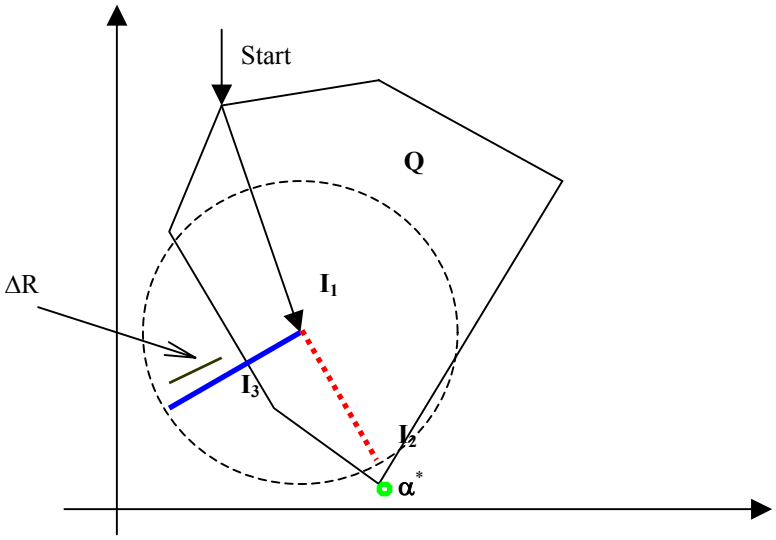

Fig 3: Feasible region, Q with minimum point α^* after iteration I_1 , the possible positions I_2 and I_3 for the choice of two arbitrary working sets.

In Fig 3, we show our momentum quantity as the amount of overstep, if the update on an arbitrary working set causes the vector α to exit the feasible region $\mathbb{Q}$. In SMO, the algorithm always maintains the feasibility of the solution by binding the variables to the bounds of the constraints as in (23). The amount of restriction is never considered further. However, from Fig 3, we can see that if the amount of momentum is large then our vector α is too close to a boundary facet and the choice of working set members may cause minimal improvement to the objective function. A solution to this is to choose a working set that does not cause the vector α to exit the region, which can be done by minimizing the amount of overstep. This leads us to the idea of momentum minimization that is investigated further in the following section.

IV. Momentum Minimization

In any iterative algorithm, the idea is always to take the best step towards the minimum of the objective function for the quickest convergence. This can be done by following the direction of steepest descent indicated by the gradient of the problem. However in the case of decomposition, we do not have the entire gradient space available to us. Instead, we are only limited to subsets of the gradient space and if the solution vector lies in a constrained region, the direction of steepest descent may not necessarily be the best[18]. This fact has also been geometrically demonstrated in the previous section.

In the case of our working set, the best that is done so far is to pick the first variable, α_1 arbitrarily and then choose the second variable such that the step size is maximized[4]. We now propose to select a working set, which minimizes the amount of momentum ΔR at every iterative step. In doing so, the actual step (39) towards the minimum of the objective function is also maximized. In a geometric sense, we want to choose a working set such that the updates α_1, α_2 not only stay within the feasible region $\mathbb{Q}$ but also such that if they hit a constraint boundary they do so as "softly" as possible. We can see this clearly from Fig.1 where the minimum momentum in I_2 carries the solution vector closer to the minimum α^* compared to I_3. We believe that this would greatly increase the rates of convergence for the decomposition method when applied to constrained problems.

Unfortunately, the actual momentum ΔR is only known after the step is taken or during the computation of the step. We could explicitly compute it beforehand by considering all the possible combinations of working set pairs at each iterative step. For n examples, this amounts to roughly n^2 combinations, which is obviously computationally expensive! Instead, we propose to use the theoretical optimal value of momentum, ΔR^* to predict the best choice of α_2 which would generate momentum close to the optimal value when they are successfully updated.

A. Theoretical Momentum Minimization

We can then derive ΔR^* i.e. the optimal ΔR to obtain the minimum $\Delta M_{y_1,y_2}$ at every step. The stationary point of $\Delta M_{y_1,y_2}$ with respect to ΔR is found by taking the gradient of (39).

$$\frac{\partial \Delta M_{y_1,y_2}}{\partial \Delta R} = -\left(y_2E_2 - y_1E_1\right) + \Delta R\left(y_1\phi\left(x_1\right) - y_2\phi\left(x_2\right)\right)^2 - \frac{(E_2 - E_1)}{\eta}\left(y_1K_{11} + y_2K_{22} - \left(y_2 + y_1\right)K_{12}\right) = 0$$

The optimum momentum which minimizes the step size function is then,

$$\Delta R^* = \frac{1}{\left(y_1\phi\left(x_1\right) - y_2\phi\left(x_2\right)\right)^2}\left[\left(y_2E_2 - y_1E_1\right) + \frac{(E_2 - E_1)}{\eta}\left(y_1K_{11} + y_2K_{22} - \left(y_2 + y_1\right)K_{12}\right)\right] \tag{47}$$

We now derive mathematically the optimal value of $\Delta M_{y_1,y_2}$ by examining the cases that could arise with an arbitrary choice of working set pairs.

Case 1: $y_1 = y_2 = 1$

$$\begin{aligned}\Delta R^* &= \frac{1}{\left(\phi\left(x_1\right) - \phi\left(x_2\right)\right)^2}\left[\left(E_2 - E_1\right) + \frac{(E_2 - E_1)}{\eta}\left(K_{11} + K_{22} - 2K_{12}\right)\right] \\ &= \frac{1}{\left(\phi\left(x_1\right) - \phi\left(x_2\right)\right)^2}\left[\left(E_2 - E_1\right) + \frac{(E_2 - E_1)}{\eta}\left(-\eta\right)\right] = 0\end{aligned} \tag{48}$$

Substituting (48) into (43) gives;

$$\Delta M^*_{1,1} = \frac{(E_2 - E_1)^2}{2\eta} \tag{49}$$

Case 2: $y_1 = y_2 = -1$

$$\begin{aligned}\Delta R^* &= \frac{1}{\left(-\phi\left(x_1\right) + \phi\left(x_2\right)\right)^2}\left[\left(-E_2 + E_1\right) + \frac{(E_2 - E_1)}{\eta}\left(-K_{11} - K_{22} + 2K_{12}\right)\right] \\ &= \frac{1}{\left(\phi\left(x_2\right) - \phi\left(x_1\right)\right)^2}\left[-\left(E_2 - E_1\right) + \frac{(E_2 - E_1)}{\eta}\left(\eta\right)\right] = 0\end{aligned} \tag{50}$$

Substituting (50) into (44) gives;

$$\Delta M^*_{-1,-1} = \frac{(E_2 - E_1)^2}{2\eta} \tag{51}$$

Case 3: $y_1 \neq y_2$

For $y_1 = 1, y_2 = -1$

$$\Delta R^* = \frac{1}{\left(\phi(x_1)+\phi(x_2)\right)^2}\left[\left(E_2+E_1\right)+\frac{(E_2-E_1)}{\eta}\left(K_{22}-K_{11}\right)\right] \tag{52}$$

$$\begin{aligned}
\Delta M_{-1,1}{}^* &= \frac{(E_2-E_1)^2}{2\eta} - \Delta R\left(E_2+E_1\right) - \frac{\Delta R(E_2-E_1)}{\eta}\left(K_{22}-K_{11}\right) + \frac{\Delta R^2}{2}\left(\phi(x_1)+\phi(x_2)\right)^2 \\
&= \frac{(E_2-E_1)^2}{2\eta} - \frac{1}{\left(\phi(x_1)+\phi(x_2)\right)^2}\left[\left(E_2+E_1\right)^2+\frac{(E_2^2-E_1^2)}{\eta}\left(K_{22}-K_{11}\right)\right] \\
&\quad - \frac{(E_2^2-E_1^2)}{\left(\phi(x_1)+\phi(x_2)\right)^2\eta}\left(K_{22}-K_{11}\right) + \frac{(E_2-E_1)^2}{\left(\phi(x_1)+\phi(x_2)\right)^2\eta^2}\left(K_{22}-K_{11}\right)^2 \\
&\quad + \frac{\left(\phi(x_1)-\phi(x_2)\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^4}\left[\left(E_2+E_1\right)+\frac{(E_2-E_1)}{\eta}\left(K_{22}-K_{11}\right)\right]^2 \\
&= \frac{(E_2-E_1)^2}{2\eta} - \frac{\left(E_2+E_1\right)^2}{\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{(E_2^2-E_1^2)\left(K_{22}-K_{11}\right)}{\eta\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{(E_2^2-E_1^2)\left(K_{22}-K_{11}\right)}{\eta\left(\phi(x_1)+\phi(x_2)\right)^2} \\
&\quad + \frac{(E_2-E_1)^2\left(K_{22}-K_{11}\right)^2}{\left(\phi(x_1)+\phi(x_2)\right)^2\eta^2} + \frac{\left(E_2+E_1\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^2} + \frac{(E_2-E_1)^2\left(K_{22}-K_{11}\right)^2}{2\eta^2\left(\phi(x_1)+\phi(x_2)\right)^2} + \frac{(E_2^2-E_1^2)\left(K_{22}-K_{11}\right)}{\left(\phi(x_1)+\phi(x_2)\right)^2\eta}
\end{aligned}$$

$$\begin{aligned}
&= \frac{(E_2-E_1)^2}{2\eta} - \frac{(E_2^2-E_1^2)\left(K_{22}-K_{11}\right)}{\eta\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{\left(E_2+E_1\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{(E_2-E_1)^2\left(K_{22}-K_{11}\right)^2}{\left(\phi(x_1)+\phi(x_2)\right)^2\eta^2} + \frac{(E_2-E_1)^2\left(K_{22}-K_{11}\right)^2}{2\eta^2\left(\phi(x_1)+\phi(x_2)\right)^2} \\
&= \frac{(E_2-E_1)^2}{2\eta} - \frac{(E_2-E_1)^2\left(K_{22}-K_{11}\right)}{2\eta^2\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{\left(E_2+E_1\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{(E_2^2-E_1^2)\left(K_{22}-K_{11}\right)}{\eta\left(\phi(x_1)+\phi(x_2)\right)^2} \\
&= \frac{(E_2-E_1)^2}{\eta}\left(\frac{1}{2}+\frac{1}{2}\right) - \frac{\left(E_2+E_1\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^2} + \frac{(E_2^2-E_1^2)\left(\phi(x_2)-\phi(x_1)\right)\left(\phi(x_2)+\phi(x_1)\right)}{\left(\phi(x_1)-\phi(x_2)\right)^2\left(\phi(x_1)+\phi(x_2)\right)^2}
\end{aligned}$$

$$= \frac{(E_2-E_1)^2}{\eta} - \frac{\left(E_2+E_1\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{(E_2^2-E_1^2)}{\left(K_{22}-K_{11}\right)} \tag{53}$$

Case 4: If $y_1 \neq y_2, y_1 = 1, y_2 = -1$, we then have the symmetry,

$$\Delta R^* = -\frac{1}{\left(\phi(x_1)+\phi(x_2)\right)^2}\left[\left(E_2+E_1\right)+\frac{(E_2-E_1)}{\eta}\left(K_{22}-K_{11}\right)\right] \tag{54}$$

$$\begin{aligned}
\Delta M_{1,-1}{}^* &= \frac{(E_2-E_1)^2}{2\eta} + \Delta R\left(E_2+E_1\right) + \frac{\Delta R(E_2-E_1)}{\eta}\left(K_{22}-K_{11}\right) + \frac{\Delta R^2}{2}\left(\phi(x_1)+\phi(x_2)\right)^2 \\
&= \frac{(E_2-E_1)^2}{2\eta} - \frac{1}{\left(\phi(x_1)+\phi(x_2)\right)^2}\left[\left(E_2+E_1\right)^2+\frac{(E_2^2-E_1^2)}{\eta}\left(K_{22}-K_{11}\right)\right] - \frac{(E_2^2-E_1^2)}{\left(\phi(x_1)+\phi(x_2)\right)^2\eta}\left(K_{22}-K_{11}\right) \\
&\quad + \frac{(E_2-E_1)^2}{\left(\phi(x_1)+\phi(x_2)\right)^2\eta^2}\left(K_{22}-K_{11}\right)^2 + \frac{1}{2\left(\phi(x_1)+\phi(x_2)\right)^2}\left[\left(E_2+E_1\right)+\frac{(E_2-E_1)}{\eta}\left(K_{22}-K_{11}\right)\right]^2
\end{aligned}$$

$$= \frac{(E_2 - E_1)^2}{\eta} - \frac{\left(E_2 + E_1\right)^2}{2\left(\phi(x_1) + \phi(x_2)\right)^2} - \frac{(E_2^2 - E_1^2)}{(K_{22} - K_{11})} \qquad (55)$$

Generally, our idea of minimizing the momentum when considering a sub set of variables is meant to choose a working set that when updated, gives the best possible improvement to the objective function value. If we translate this to the SVM notion, we want to adjust the hyperplane in the quickest possible way such that the margin is maximized and the number of misclassified points is minimized. The variables or the Lagrange Multipliers of the working set directly determine the position of the hyperplane and they have a one to one correspondence with the data examples. This allows us to give in the following some geometrical interpretation of the iteration process with momentum minimization.

From (48) and (50), we see that if the working set consists of multipliers corresponding to examples of the same class label, i.e. $y_1 = y_2$, the theoretical optimal momentum is zero. This shows that the best step size would be obtained if they were not allowed to exit the constraint boundaries at all. In geometric terms, we are maximizing the margin of the hyperplane by changing two multipliers that belong to examples of the same class. Intuitively, when doing so we definitely want to push the hyperplane towards the opposite half space as much as possible, which means that, the updated multipliers should ensure that the corresponding points remain in the correct half space.

In the example of Fig 4, the working set α_w chosen for the next update after I_0 is depicted with a circle. The initial hyperplane position is labeled I_0 which in iteration terms corresponds to the initial state at t=0. The next iterative step is shown by the new hyperplane position labeled I_1 and the arrows depict the direction of motion for the hyperplane. We can see that momentum minimization not only results in the updated hyperplane correctly classifying the working set elements of I_0, but forces X_1 that was previously misclassified (α_1=C or it is at the boundary) to become a non-bound Support Vector as seen when the updated hyperplane is at I_1. Ideally, X_2 could also be a non-bound Support Vector but since it was correctly classified at I_0 and I_1 (α_2=0) then it still has minimum momentum. i.e. The multiplier α_2 remained unchanged at the constraint boundary.

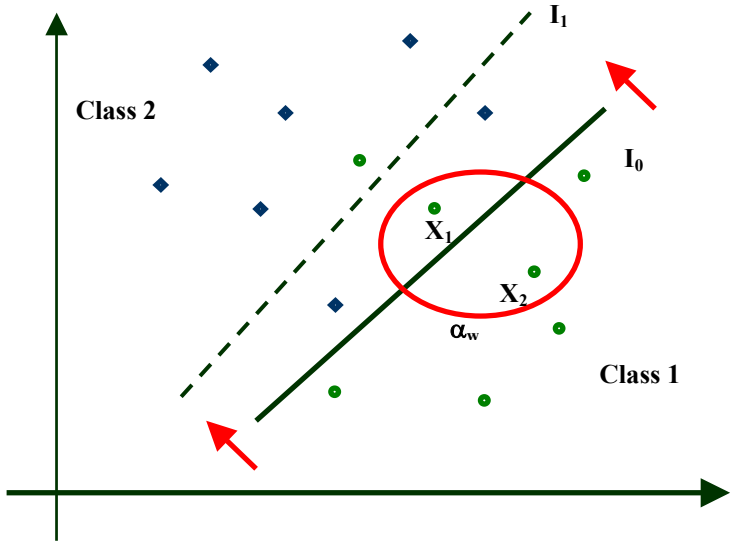

Fig 4: Effect of minimizing momentum for two examples of the same class (y_1=y_2)

However, when the working set pair consists of opposite class examples, i.e. $y_1 \neq y_2$, the optimal momentum becomes more complex. We now have to compromise between finding the maximal margin of separation and the minimization of misclassified examples between the working set pair. Currently, we believe that the theoretical optimal momentum for (52) and (54) captures this compromise but have yet been unable to clearly prove it mathematically. The equations seem to indicate that for the generic case of $y_1 \neq y_2$ we can at best allow the updated multipliers to hit the boundaries and not just touch them. We try to get an intuition for this in Fig 5, where the working set variables correspond to the violators X_1 and X_2. For the sake of illustration, let us assume that $\alpha_1, \alpha_2 = C$ at I_0 which we show in Fig 5. The updated position of the hyperplane, I_1 is shown to correctly classify the two points if the minimum momentum was zero as with the previous case. However, we can see that at I_1 the point X_3 that was previously correctly classified by I_0

becomes misclassified! It would seem that a better position would be the hyperplane at I_2 where now X_3 remains correctly classified and X_2 is also in the right half-space. However X_1 still remains misclassified ($\alpha_1 = C$) and would have some amount of momentum since it previously was at the boundary and the update would have carried it outside the feasible region. We are hence led to believe that the theoretical equations put a bound on the minimum amount of momentum that not only gives a better hyperplane position, but also the better step size towards the minimum of the objective function.

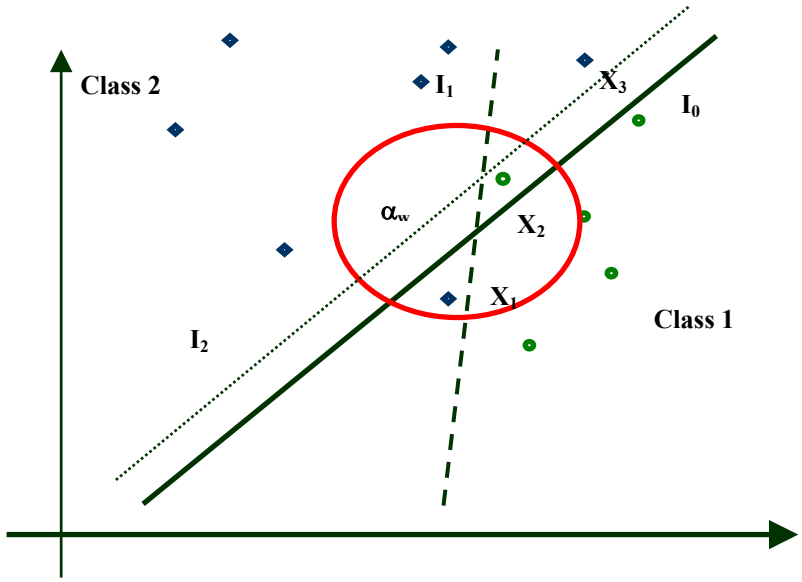

Fig 5: Effect of minimizing momentum for two examples of the opposite class ($y_1 \neq y_1$)

B. Approximate Momentum Minimization through heuristic implementation

In the previous section, we computed the theoretical minimized momentum values, ΔR^* for the two generic cases of $y_1 = y_2$ and $y_1 \neq y_2$. Some would point out that these are functions of the unknowns α_1, α_2 and depend very much on their choice beforehand. This seems to have left us in a quagmire since we cannot determine the working set choice that exactly minimizes momentum without trying all the possible working set combinations as before. However, this is not what momentum minimization attempts to do. Recall that in our decomposition setting, we are trying to select the second variable α_2 such that the step size is maximized and the resulting momentum for the pair is also minimized. We can approximate the minimum momentum by considering the generalized combinations of working sets α_w and then explicitly computing the conditions on the second variable so that the updated pair has minimal momentum. In the following, we examine more thoroughly the situations that arise for the generalized cases i.e. $y_1 = y_2$ and $y_1 \neq y_2$.

1) Working set of examples from the same class

Let us examine first the situations that occur at the Upper Bound (UB) of (33), where $UB = \min\left(C, \alpha_1 + \alpha_2\right)$.

If $\alpha_1 = 0$ and $\alpha_2 = C$, then $UB = \alpha_1 + \alpha_2$, then from (33),

$$\begin{aligned}\Delta R &= \alpha_1^{\,t} - \frac{y_1(E_2 - E_1)}{\eta} - UB \\ &= \alpha_1^{\,t} - \frac{y_1(E_2 - E_1)}{\eta} - \left(\alpha_1 + \alpha_2\right)^t \\ &= -C - \frac{y_1(E_2 - E_1)}{\eta} = 0 \quad \text{only if } E_2 = E_1 - \frac{C\eta}{y_1}\end{aligned}$$

If $\alpha_2 \le C$, we have then the condition $E_2 \le E_1 - \frac{C\eta}{y_1} < E_1$ where we now can form our first heuristic rule.

H1: If $\alpha_1 = 0$ choose $\alpha_2 = C$ such that $E_2 \le E_1$

We can check that the converse is true since the working set is arbitrary. If $\alpha_1 = C$ and $\alpha_2 = 0$, then $UB = \alpha_1 + \alpha_2$,

$$\begin{aligned}\Delta R &= \alpha_1^t - \frac{y_1(E_2 - E_1)}{\eta} - UB \\ &= \alpha_1^t - \frac{y_1(E_2 - E_1)}{\eta} - (\alpha_1 + \alpha_2)^t \\ &= -\frac{y_1(E_2 - E_1)}{\eta} = 0 \quad \text{only if } E_2 = E_1\end{aligned}$$

If $\alpha_1 \le C$, we have $E_2 \ge E_1$ from which the following heuristic follows as a result.

H2: If $\alpha_1 = C$ choose $\alpha_2 = 0$ such that $E_2 \ge E_1$ and $y_1 = y_2$

Now, suppose that $0 < \alpha_1 < C$, then $UB = \alpha_1 + \alpha_2$ iff $\alpha_1 + \alpha_2$ or $\alpha_2 < C - \alpha_1$. Then;

$$\begin{aligned}\Delta R &= \alpha_1^t - \frac{y_1(E_2 - E_1)}{\eta} - UB \\ &= \alpha_1^t - \frac{y_1(E_2 - E_1)}{\eta} - (\alpha_1 + \alpha_2)^t \\ &= -\alpha_2 - \frac{y_1(E_2 - E_1)}{\eta} = 0 \quad \text{only if } E_2 = E_1 - \frac{\alpha_2 \eta}{y_1}\end{aligned}$$

This can be further generalized to the condition in the following heuristic;

H3: If $0 < \alpha_1 < C$ choose $\alpha_2 \le C$ such that $E_2 \le E_1$

We can show further that for the case when $UB = C$, a similar heuristic set follows due to the arbitrary choice of α_1, α_2. Note however that the optimal value of momentum from (48) and (50) is $\Delta R^* = 0$. If we examine(33), we can see this likely occurs when both α_1, α_2 become non-bound Support Vectors. We could then approximate the working set which mostly likely gives minimum momentum to consist of $0< \alpha_1, \alpha_2 < C$ since in the later stages of the iteration only the non-bound Support Vectors have yet to be correctly determined. However, we need to compute the value of the objective function at each step to estimate when the algorithm is in its final stage, so this sort of heuristic is very rough and is not really computationally efficient.

2) Working set of examples from the opposite class

For the case where $y_1 \ne y_2$, let us first examine the Upper Bound (UB) in (33). Let $\alpha_L = \max\{\alpha_1, \alpha_2\}$ and suppose $\alpha_1 > \alpha_2$ and $y_1 = 1$, we then have from (33);

$$\Delta R = \alpha_1^t - C \ - \frac{(E_1 - E_2)}{\eta} \tag{56}$$

From Case 3, we have

$$\begin{aligned}\Delta R^* &= \frac{1}{\left(\phi(x_1) + \phi(x_2)\right)^2}\left[(E_2 + E_1) + \frac{(E_2 - E_1)}{\eta}(K_{22} - K_{11})\right] \\ &= \frac{(E_2 + E_1)}{\left(\phi(x_1) + \phi(x_2)\right)^2} + \frac{(E_2 - E_1)(K_{22} - K_{11})}{\eta\left(\phi(x_1) + \phi(x_2)\right)^2} \\ &\le \frac{(E_2 + E_1)}{\left(\phi(x_1) + \phi(x_2)\right)^2} + \frac{(E_2 - E_1)}{\eta} \\ &\le \frac{(E_2 + E_1)}{\left(\phi(x_1) + \phi(x_2)\right)^2} - \frac{(E_1 - E_2)}{\eta}\end{aligned} \tag{57}$$

since $\frac{(K_{22}-K_{11})}{(\phi(x_1)+\phi(x_2))^2} \le 1$ and equality exists when $\phi_1(x)=0$.

By equating (56) and (57), we get;

$$\alpha_1^t - C \le \frac{(E_2+E_1)}{(\phi(x_1)+\phi(x_2))^2}$$

$$\alpha_1^t \le C + \frac{(E_2+E_1)}{(\phi(x_1)+\phi(x_2))^2}$$

$$\rightarrow \alpha_1^t > 0$$

Now since $\alpha_1 = \alpha_L$, then the choice of α_2 becomes $\alpha_2 \le C$ with the maximum value of $\alpha_2 = C$. In the case that $\alpha_1 = 0$, we could immediately try $\alpha_2 = 0$ so that there is a better chance of minimizing ΔR. We summarize this in the following two heuristics.

H4: If $\alpha_1 = 0$, choose $\alpha_2 = 0$ if $y_1 \ne y_2$.
H5: If $\alpha_1 > 0$, choose $\alpha_2 = C$ if $y_1 \ne y_2$.

Conversely, we could also examine the Lower Bound (LB), but due to the arbitrary selection of the working sets, this would yield similar counterparts to the heuristics **H4** and **H5**. We now list these several additional heuristics in addition to the requirement in Observation 1 that is we want to choose the working set that minimizes momentum and also gives the maximal update.

Momentum Minimization Heuristic Set

If $y_1 = y_2$, we use the following heuristics to approximately minimize the momentum and such that $|E_2 - E_1|$ is maximal,
H1: If $\alpha_2 = C$ choose $\alpha_1 = 0$ such that $E_2 \le E_1$
H2: If $\alpha_2 = 0$ choose $\alpha_1 = C$ such that $E_2 \ge E_1$
H3: If $0 < \alpha_2 < C$ choose $\alpha_1 \le C$ such that $E_1 \le E_2$

If $y_1 \ne y_2$, we then use the following heuristics to approximately minimize the momentum and such that $|E_2 - E_1|$ is maximal,
H4: If $\alpha_2 = 0$, choose $\alpha_1 = 0$
H5: If $\alpha_2 > 0$, choose $\alpha_1 = C$

Final: If updating any α_1 is still unsuccessful, try any α_2

The last heuristic, simply allows the algorithm to check all possibilities to ensure that the solution is truly optimal. We note that this is a rough heuristic set that attempts to perform "approximate" momentum minimization. The heuristic set however imposes a stricter rule for the choice of working sets and no longer contains any random elements as in Platt's original implementation of SMO. We note that further calculations would yield more stringent conditions on the bounds of E_1 and E_2 which are currently quite loose and based on several approximations.

V. EXPERIMENTAL RESULTS

In order to investigate the performance of momentum minimization, we implemented the basic SMO update rule together with our heuristic set for the choice of working set members. We refer to our algorithm loosely as Momentum Minimization Optimization (MMO) that is an attempt to investigate our notion of momentum. We then ran a series of benchmarks on some well-known datasets to compare with the original SMO results. All experiments were carried out on a Pentium IV, 1.5 GHz computer with 256MB RAM. The SMO program and MMO were implemented on Visual ++

6.0. We recorded the number of iterations and epochs required till convergence was established. Here iterations refer to the number of successful updates on the working sets and epochs refer to the number of loops which the algorithm goes through in the main program. MMO uses an SMO update rule with our heuristics for choosing working sets. Hence, when comparing these two Newtonian decomposition methods, we can compare the number of iterations and epochs to determine the improvement in performance. We have assumed that they are proportional to other performance measures such as CPU time, kernel computations or flop counts. We omit these measures here because our machine is different from Platt's.

We ran our MMO on the UCI adult data set and the web dataset[21] using the Gaussian kernels with kernel width of 10. We also compare the performance on our fish data set using different values of C for a Gaussian and Polynomial kernel. Our fish problem[22] is highly inseparable and the task is to classify 5 species of fish given 1400 examples with 10 attributes each. For this experiment, we trained a binary SVM classifier to classify species 1 against the rest of the species. All experiments were carried out using a tolerance of 0.001. We note here that our implementation of Platt's original pseudocode seems to give very much slower results than what was reported in [4] but nevertheless we compare our results against that as reported by Platt. For illustrative purposes, we compare the behavior of the negative objective function (8) in for the UCI 1 dataset.

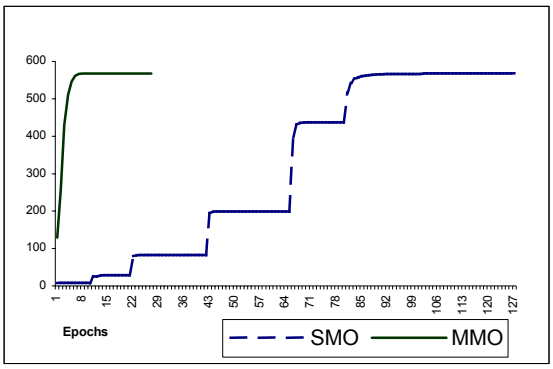

Fig 6: MMO and SMO: Comparison of objective function behavior for UCI adult 1.

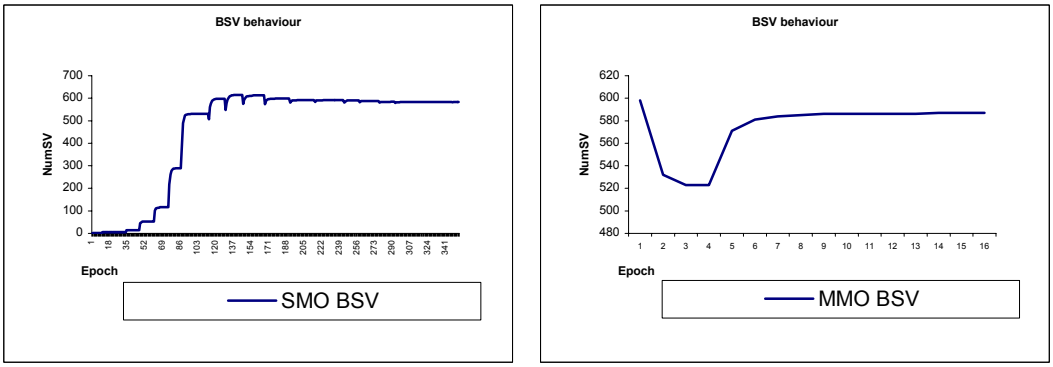

Fig 7: SMO and MMO: Comparison of bound Support Vector behavior for UCI adult 1.

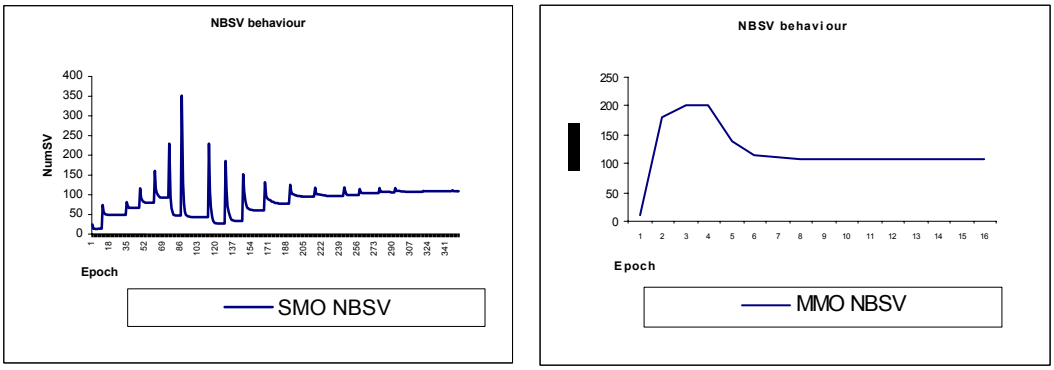

Fig 8: SMO and MMO: Comparison of non-bound Support Vector behaviour for UCI adult 1.

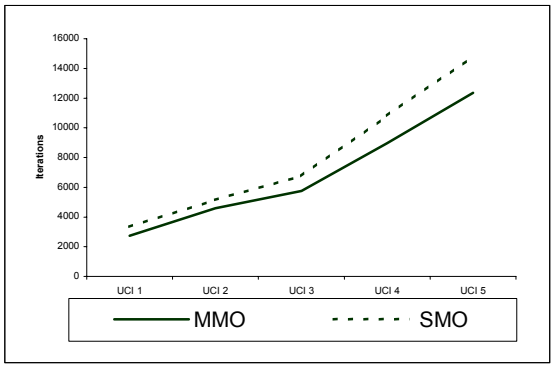

Fig 9: Performance of original SMO against MMO across the UCI adult datasets for Gaussian kernel with C=1.

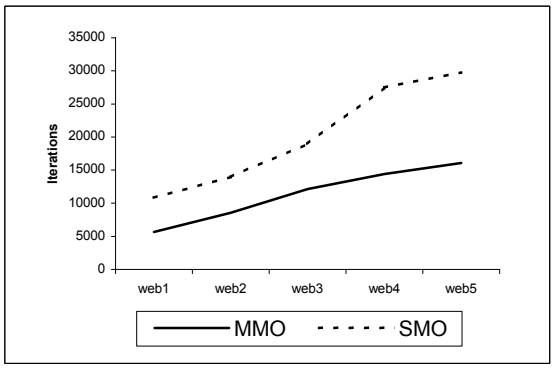

Fig 10: Performance of original SMO against MMO across the UCI Web datasets for Gaussian kernel with C=5.

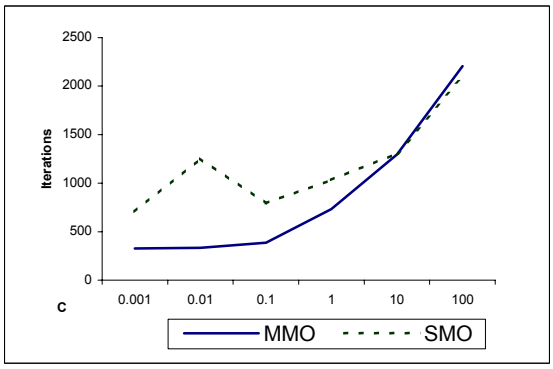

Fig 11: Performance of original SMO against MMO for fish data set for Gaussian kernel over a range of C.

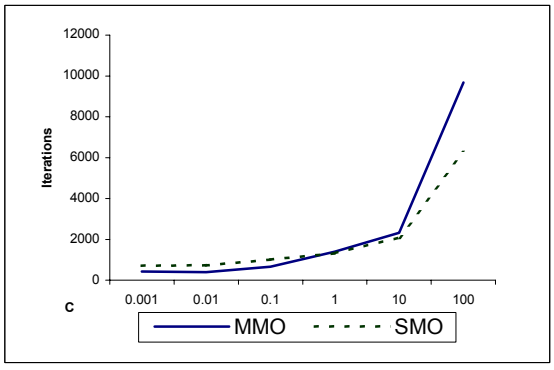

Fig 12: Performance of original SMO against MMO for fish data set for Linear kernel over a range of C.

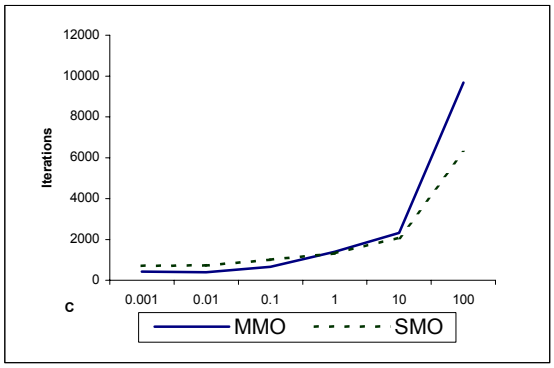

Fig 13: Performance of original SMO against MMO for fish data set for Polynomial kernel over a range of C.

VI. DISCUSSION

In Fig 6, we observe that the SMO algorithm does not improve the objective function significantly when iterating over non-bound Support Vectors. This heuristic is employed in the second loop of Platt's algorithm and it causes step-like behavior that slows down the rate of convergence of SMO. This demonstrates that the wrong choice of working sets does affect the convergence rates of the decomposition method. However in MMO, the objective function approaches the optimum value quickly and there is a strict increase in the Wolfe dual at all times. The reason for this can be seen from Fig 7 and Fig 8, where SMO finds the bounded and non-bounded Support Vectors rather slowly through the update of a random working set choice. In MMO, the necessary Support Vectors are discovered quickly through the proper choice of working set variables. Note also that the number of epochs required for convergence is much less in MMO than in SMO, indicating also that momentum minimization could help speed up the decomposition method.

In the case of the UCI data sets, Fig 9 and Fig 10 we record an improvement of 20%-40% improvement to the number of iterations required till convergence when using MMO. This seems to empirically indicate that using an approximation to our momentum minimization idea does allow a somewhat better choice of working sets to be made during the iteration. However, we observe in the fish classification experiments, Fig 11-Fig 13 that for larger values of C, the speed of MMO begins to become comparable to the normal SMO. We can explain this from a geometric perspective by observing that momentum minimization tries to provide the largest step of improvement to the objective function by ensuring that updates are not wasted trying to move variables between boundary values. However, when the optimal solution has many variables not at bound, momentum minimization does not influence the choice of working sets much. This is because most of the variables move inside the feasible region and do not hit the constraint boundaries, which results in an average momentum close to zero. We can see this in Fig 12 where a linear kernel does not separate our fish dataset well. It is also possible that a better approximation of optimal working set choice could be made and this problem may not be so apparent. Another way around it is to stop the iteration once the variables at bound have been determined and possibly apply a heuristic to check the variables not at bounds. The details of this are presently being worked on.

Platt inadvertently proposed a decomposition algorithm with a Newton iterate rule and solved the problem of applying a Newton rule to large datasets without the memory constraint problem. SMO actually ends up solving the minimization of the gradients of the Lagrange Dual which is related to the problem solved by Joachim[11]. However, we postulate that the rates of convergence of SMO are no longer quadratic due to the decomposition aspect of the problem. The enforcement of the equality constraint in the threshold is not necessary for formulations that do not explicitly minimize the threshold b e.g.[22]. We have shown that the update rule of SMO is Newtonian and the algorithm which was pointed out to be related to Bergman methods[4] is also simply just a bounded form of Newtonian iteration with a set of heuristics derived empirically. We have gone further and derived the theoretical relationship that relates the solution of the sub problem to the main problem. The results show that a proper choice of sub problems selected through momentum minimization has been shown to increase the rates of convergence. However, in some cases this may not be desirable if the sub problem that gives the optimal step is expensive to compute. We note that the rough heuristic set used here is just an approximation of the optimal momentum minimization. We could obtain a better approximation to the optimal minimal momentum by considering search methods and better prediction techniques. Our initial results so far seem to suggest that the momentum minimization idea could greatly improve the convergence when the solution for an optimization problem exists in a constrained region.

Remark 1:
It can be seen from results above that the norm-reduction property of the decomposition method is dependant on the value of the step $\Delta M_{y_1,y_2}$. We can see that $\Delta M_{y_1,y_2}$ is a function of deterministic variables y_1, y_2 and also the value of ΔR. For every iterative step to be norm reducing, we require $\forall t > 0$;

$$\Delta M_{y_1,y_2} < 0 \qquad (58)$$

We can see that in all cases computed, ΔM has two common terms namely $\frac{(E_2 - E_1)^2}{\eta}$ and $\frac{\Delta R^2}{2}\left(\phi(x_1) \pm \phi(x_2)\right)^2$. We note that the right hand most term is always positive regardless and minimizing this positive term requires an exhaustive search, which is computationally expensive. Platt's first heuristic actually minimizes (largest negative) the first term since we have assumed $\eta < 0$ and the term is negative. It is clear now that minimizing the term $\frac{(E_2 - E_1)^2}{\eta}$ corresponds directly to choosing the maximal violating pair $|E_2 - E_1|$ for an arbitrary α_2.

If we choose the maximal violator pair for update, we have the following for all violators;

$$y_i f(x_i) \to y_i\left(f(x_i) - y_i\right) \to y_i E_i < 0$$

which implies when $y_i > 0$ then $E_i < 0$ and vice-versa. We then see that if $E_2 < 0$ the largest difference $|E_2 - E_1|$ would most likely be found by looking for the largest $E_1 > 0$ and vice versa. We then note that here, $y_1 \neq y_2$ become the more likely choice of candidates for the algorithm to update using the main heuristic. If this is the case, then one immediate improvement to the heuristic set is to further investigate a better estimate for minimizing the momentum in the selection of variables where $y_1 \neq y_2$.

Remark 2:
From observation 1, we choose maximal violators to minimize the negative terms but we have yet to show that the step size function, $\Delta M_{y_1,y_2}$ is always negative or towards the minimum of the objective function. In fact, we have the following necessary conditions from (43) and (45);

a) *If* $y_1 = y_2$, $\Delta M_{y_1,y_2} < 0$ *if and only if for some* α_1, α_2 *;*

$$\frac{(E_2 - E_1)^2}{2\eta} < \frac{\Delta R^2}{2}\left(\phi(x_1) - \phi(x_2)\right)^2$$

b) *If* $y_1 \neq y_2$, $\Delta M_{y_1,y_2} < 0$ *if and only if for some* α_1, α_2 *;*

$$\frac{(E_2 - E_1)^2}{2\eta} < \frac{\Delta R^2}{2}\left(\phi(x_1) - \phi(x_2)\right)^2 - \Delta R\left(E_2 + E_1\right) - \frac{\Delta R(E_2 - E_1)}{\eta}\left(K_{22} - K_{11}\right)$$

However, if we choose α_1, α_2 such that $\Delta R = \Delta R^*$ then we are assured that $\Delta M_{y_1,y_2} < 0$ holds always due to observation 1. Case 1 and Case 2 are obvious because the optimal step size is the potential step (40) which is always negative since $\eta < 0$. We show this for Case 3 and Case 4 where we have;

$$
\begin{aligned}
\Delta M^{*}_{y_1,y_2} &= \frac{(E_2-E_1)^2}{\eta} - \frac{\left(E_2+E_1\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{(E_2^2-E_1^2)}{\left(K_{22}-K_{11}\right)} \\
&= -\frac{(E_2-E_1)^2}{\left(\phi(x_2)-\phi(x_1)\right)^2} - \frac{\left(E_2+E_1\right)^2}{2\left(\phi(x_1)+\phi(x_2)\right)^2} - \frac{(E_2^2-E_1^2)}{\left(\phi(x_2)-\phi(x_1)\right)\left(\phi(x_2)+\phi(x_1)\right)} \\
&= -\frac{1}{2}\left[\frac{2(E_2-E_1)^2}{\left(\phi(x_2)-\phi(x_1)\right)^2} + \frac{\left(E_2+E_1\right)^2}{\left(\phi(x_1)+\phi(x_2)\right)^2} + \frac{2(E_2^2-E_1^2)}{\left(\phi(x_2)-\phi(x_1)\right)\left(\phi(x_2)+\phi(x_1)\right)}\right] \\
&= -\frac{1}{2}\left[\left(\frac{(E_2-E_1)}{\left(\phi(x_2)-\phi(x_1)\right)} + \frac{\left(E_2+E_1\right)}{\left(\phi(x_1)+\phi(x_2)\right)}\right)^2 + \frac{(E_2-E_1)^2}{\left(\phi(x_2)-\phi(x_1)\right)^2}\right] \\
&\le 0
\end{aligned}
\qquad (59)
$$

Thus, we have shown that minimizing momentum not only increases the rate of convergence of the decomposition method, but also ensures that the actual step size is towards the minimum of the objective function for an arbitrary working set.

Remark 3:
We note that (39) is a convex quadratic function in terms of momentum ΔR. This leads us to believe that we could attach some notion of direction to the iteration process as with actual momentum in physics. At this stage, it still remains a subject of further investigation for us.

VII. Conclusion

We have presented an iteration measure that gives some indication for the choice of optimal variables to be included in a working set for the decomposition technique. We have demonstrated that minimizing our proposed iteration momentum improves the rates of convergence greatly for the decomposition method. Future work will concentrate on improving the prediction techniques to obtain near optimal momentum minimization. We are currently investigating the effect of momentum minimization for more complex boundary geometries and also the effect of applying extrapolation while minimizing momentum.

VIII. Appendix

A. *SMO update is a Newton method*

We expand the update rule (22);

$$
\alpha_2^{\,t+1} = \alpha_2^{\,t} + y_2\frac{(E_1-E_2)}{(K_{11}+K_{22}-2K_{12})} = \alpha_2^{\,t} + \frac{(y_2 f(x_1) - y_2 y_1 - y_2 f(x_2) + 1)}{(K_{11}+K_{22}-2K_{12})} \qquad \text{(A1)}
$$

Now, we simplify (26),

$$
\begin{aligned}
&\nabla\Im(\alpha_2) \\
&= sK_{11}(\gamma - s\alpha_2) - K_{22}\alpha_2 + K_{12}\alpha_2 - sK_{12}(\gamma - s\alpha_2) - y_2v_2 + y_2v_1 - s + 1 \\
&= sK_{11}(\alpha_1 + s\alpha_2 - s\alpha_2) - K_{22}\alpha_2 + K_{12}\alpha_2 - sK_{12}(\alpha_1 + s\alpha_2 - s\alpha_2) - y_2\left(f(x_2) + b - y_1\alpha_1K_{12} - y_2\alpha_2K_{22}\right) \\
&\quad + y_2\left(f(x_1) + b - y_1\alpha_1K_{11} - y_1\alpha_2K_{21}\right) - s + 1 \\
&= s\alpha_1K_{11} - K_{22}\alpha_2 + K_{12}\alpha_2 - s\alpha_1K_{12} - y_2f(x_2) + s\alpha_1K_{12} + \alpha_2K_{22} + y_2f(x_1) - s\alpha_1K_{11} + \alpha_2K_{21} - s + 1 \\
&= y_2f(x_1) - y_2f(x_2) - s + 1 \\
&= y_2f(x_1) - y_1y_2 - y_2f(x_2) + 1
\end{aligned}
$$

Then using (27) we get the required relationship;

$$\begin{aligned}\alpha_2^{t+1} &= \alpha_2^{t} + \frac{(y_2 f(x_1) - y_2 y_1 - y_2 f(x_2) + 1)}{(K_{11} + K_{22} - 2K_{12})} \\ &\rightarrow \alpha_2^{t} - \frac{\nabla \Im(\alpha_2)}{\nabla(\nabla \Im(\alpha_2))}\end{aligned} \tag{A2}$$

Since α_2 is arbitrary, we get the following relationship for α_1;

$$\begin{aligned}\nabla \Im(\alpha_2) &= y_2(E_1 - E_2) \\ &= -y_1 y_1 y_2 (E_2 - E_1) \\ &= -sy_1(E_2 - E_1) \\ &= -s\nabla \Im(\alpha_1)\end{aligned} \tag{A3}$$

We note that (27) implies $\nabla(\nabla \Im(\alpha_2)) = \nabla(\nabla \Im(\alpha_1))$ since α_2 is arbitrary. Then, we have from (25) the following result;

$$\begin{aligned}\alpha_1^{t+1} &= \alpha_1^{t} + s(\alpha_2^{t} - \alpha_2^{t+1,clipped}) \\ &= \alpha_1^{t} + s(\alpha_2^{t} - \alpha_2^{t} + \frac{\nabla \Im(\alpha_2)}{\nabla(\nabla \Im(\alpha_2))}) \\ &= \alpha_1^{t} + s\frac{\nabla \Im(\alpha_2)}{\nabla(\nabla \Im(\alpha_2))} \\ &= \alpha_1^{t} - s^2 \frac{\nabla \Im(\alpha_1)}{\nabla(\nabla \Im(\alpha_1))} = \alpha_1^{t} - \frac{\nabla \Im(\alpha_1)}{\nabla(\nabla \Im(\alpha_1))} \rightarrow \alpha_1^{t+1} \in \mathbb{Q}\end{aligned} \tag{A4}$$

B. Gateux-differentiable functions

Definition VIII.1 (Ortega[5])

A mapping $F : D \subset R^n \rightarrow R^m$ is G-differentiable at an interior point of D if there is some linear operator, $A \in L(R^n, R^m)$ so that for any $h \in R^n$,

$$\lim_{t \rightarrow 0} \frac{1}{t}(F(x + th) - Fx - tAh) = 0 \qquad \bullet$$

REFERENCES

[1] V. N. VAPNIK, *THE NATURE OF STATISTICAL LEARNING THEORY*, 2ND ED. NEW YORK: SPRINGER, 2000.

[2] B. SCHÖLKOPF AND A. J. SMOLA, *LEARNING WITH KERNELS : SUPPORT VECTOR MACHINES, REGULARIZATION, OPTIMIZATION, AND BEYOND*. CAMBRIDGE, MASS.: MIT PRESS, 2002.

[3] A. J. SMOLA AND B. SCHÖLKOPF, "A TUTORIAL ON SUPPORT VECTOR REGRESSION," NEUROCOLT2 TECHNICAL REPORT NC2-TR-1998-030, 1998.

[4] J. PLATT, "FAST TRAINING OF SUPPORT VECTOR MACHINES USING SEQUENTIAL MINIMAL OPTIMIZATION," IN *ADVANCES IN KERNEL METHODS-SUPPORT VECTOR LEARNING*, B. SCHÖLKOPF, C. J. C. BURGES, AND A. J. SMOLA, EDS.: CAMBRIDGE MIT PRESS, 1998, PP. 185-208.

[5] J. M. ORTEGA AND W. C. RHEINBOLDT, *ITERATIVE SOLUTION OF NONLINEAR EQUATIONS IN SEVERAL VARIABLES*. NEW YORK,: ACADEMIC PRESS, 1970.

[6] B. SCHÖLKOPF, C. J. C. BURGES, AND A. J. SMOLA, *ADVANCES IN KERNEL METHODS : SUPPORT VECTOR LEARNING*. CAMBRIDGE, MASS.: MIT PRESS, 1999.

[7] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *PRACTICAL OPTIMIZATION*. LONDON ; NEW YORK: ACADEMIC PRESS, 1981.

[8] C.-W. H. A. C.-J. LIN, "A SIMPLE DECOMPOSITION METHOD FOR SUPPORT VECTOR MACHINES.," DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION ENGINEERING, 1999.

[9] E. F. OSUNA, R.; GIROSIT, F., "TRAINING SUPPORT VECTOR MACHINES: AN APPLICATION TO FACE DETECTION," PRESENTED AT COMPUTER VISION AND PATTERN RECOGNITION, 1997. PROCEEDINGS., 1997 IEEE COMPUTER SOCIETY CONFERENCE ON, 1997.

[10] S. K. S. S.S. KEERTHI, C. BHATTACHARYYA AND K.R.K. MURTHY, "IMPROVEMENTS TO PLATT'S SMO ALGORITHM FOR SVM CLASSIFIER DESIGN,," CONTROL DIVISION, DEPT. OF MECHANICAL ENGINEERING,NATIONAL UNIVERSITY OF SINGAPORE CD-99-14, 1999.

[11] T. JOACHIMS, "MAKING LARGE SCALE SUPPORT VECTOR MACHINE LEARNING PRACTICAL," IN *ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING*, B. SCHÖLKOPF, C. J. C. BURGES, AND A. J. SMOLA, EDS.: CAMBRIDGE , MIT PRESS, 1998, PP. 169-184.

[12] C.-J. LIN, "LIBSVM," HTTP://WWW.CSIE.NTU.EDU.TW/~CJLIN/.

[13] R. FLETCHER, *PRACTICAL METHODS OF OPTIMIZATION*. CHICHESTER [ENG.] ; NEW YORK: J. WILEY, 1981.

[14] N. CRISTIANINI AND J. SHAWE-TAYLOR, *AN INTRODUCTION TO SUPPORT VECTOR MACHINES : AND OTHER KERNEL-BASED LEARNING METHODS*. NEW YORK: CAMBRIDGE UNIVERSITY PRESS, 2000.

[15] C.-J. LIN, "ON THE CONVERGENCE OF THE DECOMPOSITION METHOD FOR SUPPORT VECTOR MACHINES," *NEURAL NETWORKS, IEEE TRANSACTIONS ON*, VOL. 12, PP. 1288-1298, 2001.

[16] E. G. G. S.S. KEERTHI, "CONVERGENCE OF A GENERALIZED SMO ALGORITHM FOR SVM CLASSIFIER DESIGN," CONTROL DIVISION, DEPT. OF MECHANICAL ENGINEERING,NATIONAL UNIVERSITY OF SINGAPORE 2001.

[17] C.-J. LIN, "LINEAR CONVERGENCE FOR A DECOMPOSITION METHOD FOR SUPPORT VECTOR MACHINES," NOVEMBER 2001.

[18] C.-C. CHANG, C.-W. HSU, AND C.-J. LIN, "THE ANALYSIS OF DECOMPOSITION METHODS FOR SUPPORT VECTOR MACHINES," *NEURAL NETWORKS, IEEE TRANSACTIONS ON*, VOL. 11, PP. 1003-1008, 2000.

[19] D. LAI AND N. MANI, "MATRIX FORMULATION FOR THE SUPPORT VECTOR CLASSIFIER," MONASH UNIVERSITY MECE-7-2003, 2003.

[20] L. R. FOULDS, *OPTIMIZATION TECHNIQUES : AN INTRODUCTION*. NEW YORK: SPRINGER-VERLAG, 1981.

[21] C. L. M. BLAKE, C.J., "UCI REPOSITORY OF MACHINE LEARNING DATABASES IRVINE, CA," UNIVERSITY OF CALIFORNIA, DEPARTMENT OF INFORMATION AND COMPUTER SCIENCE, 1998.

[22] D. LAI, M. PALANISWAMI, AND N. MANI, "FAST LINEAR STATIONARY METHODS FOR AUTOMATICALLY BIASED SUPPORT VECTOR MACHINES," PRESENTED AT NEURAL NETWORKS, 2003. PROCEEDINGS OF THE INTERNATIONAL JOINT CONFERENCE ON, 2003.