Stability Analysis of the Decomposition Method for solving Support Vector Machines

D.Lai *, **A.Shilton**†, **N.Mani***, **M.Palaniswami**†
* Department of Electrical and Computer Systems Engineering,
Monash University, Clayton, Vic. 3168, Australia.
† Department of Electrical and Electronic Engineering,
The University of Melbourne Vic. 3010, Australia.
daniel.lai@eng.monash.edu.au

***Abstract*— In situations where processing memory is limited, the Support Vector Machine quadratic program can be decomposed into smaller sub-problems and solved sequentially. The convergence of this method has been proven previously through the use of a counting method. In this initial investigation, we approach the convergence analysis by treating the decomposed sub-problems as subsystems of a general system. The gradients of the sub-problems and the inequality constraints are explicitly modelled as system variables. The change in these variables during optimization form a dynamic system modelled by vector differential equations. We show that the change in the objective function can be written as the energy in the system. This makes it a natural Lyapunov function which has an asymptotically stable point at the origin. The asymptotic stability of the whole system then follows under certain assumptions.**

I. Introduction

The Support Vector Machines (SVM) developed by Vapnik [1] has been shown to be a powerful supervised learning tool for pattern recognition problems. The data to be classified is usually written as:

$$\begin{aligned}&\mathbf{\Theta} = \{(\mathbf{x_1}, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n)\}\\ &\mathbf{x}_i \in \Re^m\\ &y_i \in \{-1, 1\}\end{aligned} \tag{1}$$

The SVM formulation is essentially a regularized minimization problem leading to the use of Lagrangian 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. The idea is further extended for data that is not linearly separable; where it is first mapped via a nonlinear function to a possibly higher dimension feature space. The nonlinear function usually defined as $\phi(\mathbf{x}) : \mathbf{x} \subset \Re^n \rightarrow \Re^m, \quad n << m$ is never explicitly used in the calculation. We note that maximizing the margin of the hyperplane in either space is equivalent to maximizing the distance between the class boundaries.

The following dual problem expressed solely in terms of Lagrange multipliers, α_i is usually solved:

$$\begin{aligned}\min_{\boldsymbol{\alpha}\in\mathbf{D}} \Im(\boldsymbol{\alpha}) =& \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{e}\\ \mathbf{D} =& \{\boldsymbol{\alpha} | 0 \leq \alpha_{\mathrm{i}} \leq C, \boldsymbol{\alpha}^T\mathbf{y} = 0\}\end{aligned} \tag{2}$$

where

$$\begin{aligned}G_{ij} &= y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\\ \mathbf{e} &= \{1 \ldots 1\}\end{aligned}$$

The explicit definition of the nonlinear function $\phi(.)$, has been circumvented by the used of a kernel function, defined formally as the dot products of the nonlinear functions;

$$K(\mathbf{x}_i, \mathbf{x}_j) = < \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) > \tag{3}$$

The trained classifier then has the following form:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{4}$$

The decomposition method is generally applied to situations where computing memory is limited and hence not all problem variables can be considered simultaneously during optimization. The method decomposes the main problem into a series of sub-problems which are then solved sequentially where a sub-problem is identified by the corresponding set of variables known as *working sets* in SVM literature. Osuna [2] is usually credited as the earliest to apply a form of this method to the SVM problem which he called *chunking*. Later algorithms such as SMO [3] and SVMlight [4] selected working sets based on the steepest search direction. It has been recognized by Lin, Hsu [5] and Chang [6] that the choice of working sets is central to the speed of the decomposition method. Lin [7], [8] also showed that working sets chosen in this manner resulted in a linear convergence rate. This was empirically confirmed by Laskov [9] who further showed that decomposition was sometimes faster than optimization on the entire problem space. The convergence of the problem under a SMO type algorithm has been proven by Keerthi and Gilbert [10] using a counting method. A general

assumption is that the rate of convergence is proportional to the rate of improvement to the objective function [9].

In this initial investigation, we examine the convergence of the decomposition method using stability analysis. We first model the optimization process as a dynamic system governed by a vector differential equation. The gradients of the sub-problem and the potential of overstep form the variables of this second order system. The potential of overstep explicitly measures the amount of constraint violation when the updated variable is constrained to the feasible region. We show that this system is non-conservative and the energy of the system turns out to be the change in the objective function. Next using Lyapunov's direct method we show that the dynamic system has an asymptotically stable point at the origin which results in zero energy loss. We begin by formally defining some notations which will be used in the analysis and proceed to describe the decomposition technique. In Section III, we model the optimization process as a dynamic system and then apply stability analysis to the resulting system. We use the standard notation where scalars are denoted by italics, column vectors by boldface small letters and matrices by boldface capitals. Due to notation complexity, we take $\mathbf{x}(i)$ to mean the *i*-th element of $\mathbf{x}$.

II. The Decomposition Method for Support Vector Machines

In the decomposition technique, the main optimization problem is broken down into a series of sub-problems. The solution of a sub-problem results in the update of the working set variables while the variables excluded from the working set are untouched during the optimization step. We use the subscript *p* to indicate the working set variables that are updated during an iteration and *s* to indicate variables that do not change i.e. static during the optimization step. The subscripts are also used to indicate the size of the vectors and the matrices e.g. if $\boldsymbol{\alpha}_p \in \Re^m$ then $\mathbf{G}_p \in \Re^{m \times m}$.

The decomposed sub-problem of the Lagrangian dual (2) can be written in matrix form as follows:

$$\Im\left(\boldsymbol{\alpha}_p\right) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha}_p \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{G}_p & \mathbf{G}_* \\ \mathbf{G}_*^T & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_p \\ \boldsymbol{\alpha}_s\end{bmatrix} - \begin{bmatrix}\boldsymbol{\alpha}_p \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{e}_p \\ \mathbf{e}_s\end{bmatrix} \quad (5)$$

subject to:

$$\mathbf{D}_p = \{\boldsymbol{\alpha}_p | 0 \leq \boldsymbol{\alpha}_p \leq C\mathbf{1}, \boldsymbol{\alpha}_p^T\ \mathbf{y}_p = 0\}$$

To simplify further, we can incorporate the equality constraint into the objective function by treating the scalar b, as a Lagrangian multiplier and write (5) as:

$$\Im\left(\boldsymbol{\alpha}_p, b\right) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha}_p \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{G}_p & \mathbf{G}_* \\ \mathbf{G}_*^T & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_p \\ \boldsymbol{\alpha}_s\end{bmatrix} - \begin{bmatrix}\boldsymbol{\alpha}_p \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{e}_p \\ \mathbf{e}_s\end{bmatrix} + b\begin{bmatrix}\boldsymbol{\alpha}_p \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{y}_p \\ \mathbf{y}_s\end{bmatrix} \quad (6)$$

subject to:

$$\mathbf{D}_p = \{\boldsymbol{\alpha}_p | 0 \leq \boldsymbol{\alpha}_p \leq C\mathbf{1}\}$$

This general form of the decomposed sub-problem has previously been proposed in [11] and has been shown to give a similar result to (5).

Direct minimization of (6) on the feasible region defined by the sub-space $\mathbf{D}_p$ can be done by finding the stationary gradients. This naturally results in the Newton method. However, one can employ other update methods as long as one is careful to ensure the updated variables remain in $\mathbf{D}_p$ e.g. [12]. The nature of our problem simplifies this considerably because we have m-inequality constraints which define $\mathbf{D}_p$. We first write the decomposed sub-problem as:

$$\Im\left(\boldsymbol{\alpha}_p, b\right) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha}_p \\ b \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{G}_p & \mathbf{y}_p & \mathbf{G}_* \\ \mathbf{y}_p^T & 0 & \mathbf{y}_s^T \\ \mathbf{G}_*^T & \mathbf{y}_s & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_p \\ b \\ \boldsymbol{\alpha}_s\end{bmatrix} - \begin{bmatrix}\boldsymbol{\alpha}_p \\ b \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{e}_p \\ 0 \\ \mathbf{e}_s\end{bmatrix}$$

or more compactly in augmented vector form as:

$$\Im\left(\boldsymbol{\alpha}_p'\right) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha}_p' \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{H}_p & \mathbf{H}_* \\ \mathbf{H}_*^T & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_p' \\ \boldsymbol{\alpha}_s\end{bmatrix} - \begin{bmatrix}\boldsymbol{\alpha}_p' \\ \boldsymbol{\alpha}_s\end{bmatrix}^T \begin{bmatrix}\mathbf{e}_p' \\ \mathbf{e}_s\end{bmatrix} \quad (7)$$

subject to:

$$\mathbf{D}_p = \{\boldsymbol{\alpha}_p | 0 \leq \boldsymbol{\alpha}_p \leq C\mathbf{1}\}$$

where the augmented vectors corresponding to the working set is:

$$\boldsymbol{\alpha}_p' = \begin{bmatrix}\boldsymbol{\alpha}_p \\ b\end{bmatrix}$$
$$\mathbf{H}_p = \begin{bmatrix}\mathbf{G}_p & \mathbf{y}_p \\ \mathbf{y}_p^T & 0\end{bmatrix}$$
$$\mathbf{H}_* = \begin{bmatrix}\mathbf{G}_* \\ \mathbf{y}_s^T\end{bmatrix}$$
$$\mathbf{e}_p' = \begin{bmatrix}\mathbf{e}_p \\ 0\end{bmatrix}$$

Since α_s is treated as "static" the general gradient vector is:

$$\underset{\boldsymbol{\alpha}_p'}{\Im'}\left(\boldsymbol{\alpha}_p'\right) = \begin{bmatrix}\mathbf{H}_p & \mathbf{H}_* \\ \mathbf{O}_*^T & \mathbf{O}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_p' \\ \boldsymbol{\alpha}_s\end{bmatrix} - \begin{bmatrix}\mathbf{e}_p' \\ \mathbf{0}_s\end{bmatrix} \quad (8)$$

The elements of the gradient vector corresponding to the "static" variables are zero as expected. The Hessian matrix is then:

$$\underset{\boldsymbol{\alpha}_p'}{\Im''}\left(\boldsymbol{\alpha}_p'\right) = \begin{bmatrix}\mathbf{H}_p & \mathbf{O}_* \\ \mathbf{O}_*^T & \mathbf{O}_s^T\end{bmatrix} \quad (9)$$

and $\mathbf{O}$ represents a matrix of zeroes of appropriate size. The update rule for the variables is then found by setting the gradients to zero as follows:

$$\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_* \\ \mathbf{O}_*^T & \mathbf{O}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_p' \\ \boldsymbol{\alpha}_s\end{bmatrix} - \begin{bmatrix}\mathbf{e}_p' \\ \mathbf{0}_s\end{bmatrix} = \begin{bmatrix}\mathbf{0}_p' \\ \mathbf{0}_s\end{bmatrix}$$

Let t denote an arbitrary iteration step then:

$$\begin{bmatrix}\boldsymbol{\alpha}'_p\\ \boldsymbol{\alpha}_s\end{bmatrix}^{t+1}=\begin{bmatrix}\boldsymbol{\alpha}'_p\\ \boldsymbol{\alpha}_s\end{bmatrix}^{t}-\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{O}_*^T & \mathbf{O}_s\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{O}_*^T & \mathbf{O}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}'_p\\ \boldsymbol{\alpha}_s\end{bmatrix}+\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{O}_*^T & \mathbf{O}_s\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{e}'_p\\ \mathbf{0}_s\end{bmatrix} \quad (10)$$

This form is meant to show that $\boldsymbol{\alpha}_s$ remains unchanged but is difficult to compute directly since the composite matrix

$$\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{O}_*^T & \mathbf{O}_s\end{bmatrix}$$

is singular. However, if we remove the rows and columns corresponding to $\boldsymbol{\alpha}_s$ in (8) and (9) we obtain respectively the augmented gradient vector corresponding to the working set variables:

$$\begin{aligned}\mathbf{v}'_p =&\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}'_p\\ \boldsymbol{\alpha}_s\end{bmatrix}-\mathbf{e}'_p\\ =&\mathbf{H}_p\boldsymbol{\alpha}'_p+\mathbf{H}_*\boldsymbol{\alpha}_s-\mathbf{e}'_p\end{aligned} \quad (11)$$

The unconstrained update rule is then:

$$\boldsymbol{\alpha'}_p^{t+1}=\boldsymbol{\alpha'}_p^{t}-\mathbf{H}_p^{-1}\mathbf{v}'_p \quad (12)$$

We assume that $\mathbf{H}_p$ is non-singular which should hold as long as the kernel function is positive definite. The unconstrained update rule (12) is clearly Newtonian. All that remains is to ensure that the updated variables remain in $\mathbf{D}_p$ which can be done simply by restricting them to the bounds of the inequality constraints should they overstep the bounds. The constrained updated multiplier $\forall i=1...m$ is then:

$$\underset{restricted}{\boldsymbol{\alpha}_p^{t+1}}(i)=\begin{cases}C & \text{if } \boldsymbol{\alpha}_p^{t+1}(i)>C\\ 0 & \text{if } \boldsymbol{\alpha}_p^{t+1}(i)<0\\ \boldsymbol{\alpha}_p^{t+1}(i) & \text{otherwise}\end{cases} \quad (13)$$

In [13] we proposed the variable $\boldsymbol{\tau}_p(i)$ to account for *potential overstep* i.e. if the iteration step causes the i-th updated variable to exit the bounds. We also define the quantity $\boldsymbol{d}_p(i)$ as the distance of the i-th variable to either the Upper Bound (UB) or Lower Bound (LB) depending on the direction of update. We then have the following augmented vector relationship:

$$\boldsymbol{\tau}'_p=-\mathbf{H}_p^{-1}\mathbf{v}'_p-\boldsymbol{d}'_p \quad (14)$$

where the augmented potential overstep vector is

$$\boldsymbol{\tau}'_p=\begin{bmatrix}\boldsymbol{\tau}_p\\ \tau_b\end{bmatrix}$$

and the augmented vector of distances is

$$\boldsymbol{d}'_p=\begin{bmatrix}\boldsymbol{d}_p\\ d_b\end{bmatrix}$$

The scalars τ_b and d_b are introduced since we treat b as a variable. We believe that they could be interpreted loosely as the potential of violating the equality constraint in (2) and the distance to the supremum of b respectively. When solving the problem in [3], these scalars are not required in the computation since the inequality bounds are explicitly adjusted to account for them. We now write the constrained version of (12) as:

$$\boldsymbol{\alpha'}_p^{t+1}=\boldsymbol{\alpha'}_p^{t}-\mathbf{H}_p^{-1}\mathbf{v}'_p-\boldsymbol{\tau}'_p \quad (15)$$

which ensures that $\boldsymbol{\alpha}_p^{t+1}\in\mathbf{D}_p$ and b minimizes the corresponding term in (6). The change in objective function during an iteration can be computed as follows:

***Proposition** 2.1 (Change in Objective Function):* Let the kernel function $K(\mathbf{x}_i,\mathbf{x}_j)$ be positive definite and suppose we solve (7) using (15). The change in objective function value for any $\boldsymbol{\alpha}'_p\subset\boldsymbol{\alpha}'$ and $\forall p=1...p'$ is

$$M(\mathbf{v}'_p,\boldsymbol{\tau}'_p)=\frac{1}{2}\left[\boldsymbol{\tau'}_p^T\mathbf{H}_p\boldsymbol{\tau'}_p-\mathbf{v}_p^{'T}\mathbf{H}_p^{-1}\mathbf{v}'_p\right] \quad (16)$$

where the Hermitian matrix $\mathbf{H}_p$ is the Hessian of the subproblem, $\mathbf{v}'_p$ is the augmented gradient vector and $\boldsymbol{\tau'}_p$ is the vector of potential overstep.

Proof: The proof is through direct calculation where the change in objective function is defined for an arbitrary step t as:

$$\triangle\Im=\Im(\boldsymbol{\alpha}_p^{t+1})-\Im(\boldsymbol{\alpha}_p^{t})$$

Using (7) and recalling that $\boldsymbol{\alpha}_s$ is static during the iteration we have:

$$\begin{aligned}\triangle\Im=&\frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t+1}\\ \boldsymbol{\alpha}_s\end{bmatrix}^T\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{H}_*^T & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t+1}\\ \boldsymbol{\alpha}_s\end{bmatrix}\\ &-\begin{bmatrix}\boldsymbol{\alpha'}_p^{t+1}\\ \boldsymbol{\alpha}_s\end{bmatrix}^T\begin{bmatrix}\mathbf{e}'_p\\ \mathbf{e}_s\end{bmatrix}+\begin{bmatrix}\boldsymbol{\alpha'}_p^{t}\\ \boldsymbol{\alpha}_s\end{bmatrix}^T\begin{bmatrix}\mathbf{e}'_p\\ \mathbf{e}_s\end{bmatrix}-\\ &\frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t}\\ \boldsymbol{\alpha}_s\end{bmatrix}^T\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{H}_*^T & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t}\\ \boldsymbol{\alpha}_s\end{bmatrix}\end{aligned}$$

which can be simplified as follows:

$$\begin{aligned}\triangle\Im=&\frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t+1}\\ \boldsymbol{\alpha}_s\end{bmatrix}^T\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{H}_*^T & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t+1}\\ \boldsymbol{\alpha}_s\end{bmatrix}\\ &-\frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t}\\ \boldsymbol{\alpha}_s\end{bmatrix}^T\begin{bmatrix}\mathbf{H}_p & \mathbf{H}_*\\ \mathbf{H}_*^T & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha'}_p^{t}\\ \boldsymbol{\alpha}_s\end{bmatrix}\\ &-\begin{bmatrix}\boldsymbol{\alpha'}_p^{t+1}-\boldsymbol{\alpha'}_p^{t}\\ \boldsymbol{\alpha}_s-\boldsymbol{\alpha}_s\end{bmatrix}^T\begin{bmatrix}\mathbf{e}'_p\\ \mathbf{e}_s\end{bmatrix}\\ =&\frac{1}{2}\left[\boldsymbol{\alpha'_p}^{t+1^T}\mathbf{H}_p\boldsymbol{\alpha'_p}^{t+1}-\boldsymbol{\alpha'_p}^{t^T}\mathbf{H}_p\boldsymbol{\alpha'_p}^{t}\right]\\ &+\boldsymbol{\alpha}_s^T\mathbf{H}_*^T(\boldsymbol{\alpha'}_p^{t+1}-\boldsymbol{\alpha'}_p^{t})\\ &-(\boldsymbol{\alpha'}_p^{t+1}-\boldsymbol{\alpha'}_p^{t})^T\mathbf{e}'_p\\ =&\frac{1}{2}\Delta\boldsymbol{\alpha'_p}^T\mathbf{H}_p\Delta\boldsymbol{\alpha'_p}+\Delta\boldsymbol{\alpha'_p}^T\mathbf{v}'_p\end{aligned}$$

where

$$\Delta\boldsymbol{\alpha'_p}=\boldsymbol{\alpha'}_p^{t+1}-\boldsymbol{\alpha'}_p^{t}$$

Now using (11) and (15) we then write the change in objective function as a function of $\mathbf{v}'_p$ and $\boldsymbol{\tau}'_p$ as follows:

$$\begin{aligned}\triangle\Im =&M(\mathbf{v}'_p, \boldsymbol{\tau}'_p)\\ =&\frac{1}{2}\left[(\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau'}_p)^T\mathbf{H}_p(\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau'}_p)\right]\\ &-(\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau'}_p)^T\mathbf{v}'_p\end{aligned}$$

After further algebra we retrieve the required result:

$$M(\mathbf{v}'_p, \boldsymbol{\tau}'_p)= \frac{1}{2}\left[\boldsymbol{\tau'}_p^T\mathbf{H}_p\boldsymbol{\tau'}_p - \mathbf{v}_p^{'T}\mathbf{H}_p^{-1}\mathbf{v}'_p\right]$$

■

III. Stability Analysis of the Decomposition Method

In order to analyse the stability of the decomposition method, we examine the change in the gradients, $\mathbf{v}'_p$ and the potential overstep, $\boldsymbol{\tau'}_p$ of an arbitrary sub-problem. There are several reasons for this choice; the first is due to the fact that the gradients form the KKT conditions of the constrained minimization problem and the analysis becomes mathematically nice in terms of these variables.

A. Modelling decomposition as a dynamic system

We attempt to investigate the change in gradients when a sub-problem is solved through a number of iterations. We denote T as the duration of the iteration process or *horizon* of the dynamic system in control theory. The change in gradients is defined as:

$$\delta\mathbf{v}'_p = \mathbf{v}'^{\,t+1}_p - \mathbf{v}'^{\,t}_p$$

Using (11) this then becomes

$$\begin{aligned}\delta\mathbf{v}'_p =&\mathbf{H}_p(\boldsymbol{\alpha'}^{t+1}_p - \boldsymbol{\alpha'}^{t}_p)\\ =&\mathbf{H}_p(-\mathbf{H}_p^{-1}\mathbf{v}'_p - \boldsymbol{\tau'}_p)\\ =&-\mathbf{v}'_p - \mathbf{H}_p\boldsymbol{\tau'}_p\end{aligned} \tag{17}$$

Dividing both sides by an iteration step δt , we get:

$$\frac{\delta\mathbf{v}'_p}{\delta t} = \frac{-\mathbf{v}'_p - \mathbf{H}_p\boldsymbol{\tau'}_p}{\delta t} \tag{18}$$

This is a discretized vector differential equation where we take $\delta t = 1$. Now let us assume that the horizon of the iteration is long or T is very large compared to an iteration step so we have:

$$\lim_{T\to\infty,\delta t\to 0}\frac{\delta\mathbf{v}'_p}{\delta t} = \frac{d\mathbf{v}'_p}{dt} = -\mathbf{v}'_p - \mathbf{H}_p\boldsymbol{\tau'}_p$$

The optimization of a sub-problem has now been modelled as a first order dynamic system governed by the following vector differential equation:

$$\frac{d\mathbf{v}'_p}{dt} + \mathbf{v}'_p + \mathbf{H}_p\boldsymbol{\tau'}_p = 0 \tag{19}$$

The second order dynamic system can be found simply by differentiating with respect to time, t giving us:

$$\mathbf{H}_p^{-1}\frac{d^2\mathbf{v}'_p}{dt^2} + \mathbf{H}_p^{-1}\frac{d\mathbf{v}'_p}{dt} + \frac{d\boldsymbol{\tau'}_p}{dt} = 0 \tag{20}$$

The second order dynamic system (20) is actually similar in form to the general damped mass-spring system [14], [15]. However instead of scalar functions, we have the damping vector function $h(\mathbf{v}'_p) : \Re^m \to \Re^m$ given by:

$$\mathbf{h}(\mathbf{v}'_p) = \mathbf{H}_p^{-1}\frac{d\mathbf{v}'_p}{dt} \tag{21}$$

and the potential energy vector function $\mathbf{p}(\mathbf{s}) : \Re^m \to \Re^m$ written in terms of the potential of overstep:

$$\mathbf{p}(\mathbf{s}) = -\int_0^{\mathbf{v}'_p}\frac{d\boldsymbol{\tau'}_p}{dt}d\mathbf{s}$$

The potential energy is then the norm or the length of the potential energy vector defined as:

$$\begin{aligned}P(\mathbf{s})=&\left\|\mathbf{p}(\mathbf{s})\right\|\\ =&-\int_0^{\mathbf{v}'_p}\frac{d\boldsymbol{\tau'}_p}{dt}^T d\mathbf{s}\end{aligned} \tag{22}$$

We now define the energy of the system, $E(\mathbf{v}'_p, \frac{d\mathbf{v}'_p}{dt})$ as the sum of kinetic and potential energy as follows:

$$E(\mathbf{v}'_p, \frac{d\mathbf{v}'_p}{dt}) = \frac{1}{2}\left\|\frac{d\mathbf{v}'_p}{dt}\right\|^2_{\mathbf{H}_p^{-1}} + \left\|\mathbf{p}(\mathbf{s})\right\| \tag{23}$$

Using the definition in (14) and (19) this simplifies as follows:

$$\begin{aligned}E(\mathbf{v}'_p, \frac{d\mathbf{v}'_p}{dt}) =&\frac{1}{2}\frac{d\mathbf{v}'_p}{dt}^T\mathbf{H}_p^{-1}\frac{d\mathbf{v}'_p}{dt} + \left\|-\int_0^{\mathbf{v}'_p}\frac{d\boldsymbol{\tau'}_p}{dt}d\mathbf{s}\right\|\\ =&\frac{1}{2}(-\mathbf{v}'_p - \mathbf{H}_p\boldsymbol{\tau'}_p)^T\mathbf{H}_p^{-1}(-\mathbf{v}'_p - \mathbf{H}_p\boldsymbol{\tau'}_p)\\ &+\left\|\int_0^{\mathbf{v}'_p}\mathbf{H}_p^{-1}\frac{d\mathbf{s}}{dt}d\mathbf{s}\right\|\\ =&\frac{1}{2}\left[\mathbf{v}'^{\,T}_p\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau'}^T_p\mathbf{H}_p\boldsymbol{\tau'}_p\right] + \mathbf{v}'^{\,T}_p\boldsymbol{\tau'}_p\\ &+\int_0^{\mathbf{v}'_p}\left[\mathbf{H}_p^{-1}(-\mathbf{s} - \mathbf{H}_p\boldsymbol{\tau'}_p)\right]^T d\mathbf{s}\\ =&\frac{1}{2}\left[\mathbf{v}'^{\,T}_p\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau'}^T_p\mathbf{H}_p\boldsymbol{\tau'}_p\right] + \mathbf{v}'^{\,T}_p\boldsymbol{\tau'}_p\\ &-\frac{1}{2}\mathbf{v}'^{\,T}_p\mathbf{H}_p^{-1}\mathbf{v}'_p - \int_0^{\mathbf{v}'_p}(-\mathbf{H}_p^{-1}\mathbf{s} - \boldsymbol{d}'_p)^T d\mathbf{s}\\ =&\frac{1}{2}\left[\mathbf{v}'^{\,T}_p\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau'}^T_p\mathbf{H}_p\boldsymbol{\tau'}_p\right] + \mathbf{v}'^{\,T}_p\boldsymbol{\tau'}_p\\ &-\mathbf{v}'^{\,T}_p\mathbf{H}_p^{-1}\mathbf{v}'_p - \mathbf{v}'^{\,T}_p\boldsymbol{\tau'}_p\\ =&\frac{1}{2}\left[-\mathbf{v}'^{\,T}_p\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau'}^T_p\mathbf{H}_p\boldsymbol{\tau'}_p\right]\\ =&M(\mathbf{v}'_p, \boldsymbol{\tau}'_p)\end{aligned} \tag{24}$$

This shows that the energy of our dynamic system is equivalent to the change in the objective function during optimization and further analysis on the system will have a direct correspondence to the iteration process itself.

B. Lyapunov's direct method for stability

Lyapunov's direct method is generally applied to investigate the asymptotic stability of equilibrium points in a dynamic system. We review without proof Lyapunov's direct method in the following theorem which can be found in various forms e.g. [14]–[16]. Detailed proofs can be found in the references.

***Theorem** 3.1 (Lyapunov's Direct Method):* Let $\mathbf{D}$ be an open subset of $\Re^n$ containing an equilibrium point $\mathbf{x}_0$ for a function $f \in C'(\mathbf{D})$ i.e. $f(\mathbf{x})$ is from a family of first-order differentiable functions with $f(\mathbf{x}_0) = 0$. Suppose we can find a function $L \in C'(\mathbf{D})$ such that $L(\mathbf{x}_0) = 0$ and $L(\mathbf{x}) > 0$ if $\mathbf{x} \neq \mathbf{x}_0$ then,

i. If $\dot{L}(\mathbf{x}) \leq 0$ for all points $\mathbf{x} \in D$ then $\mathbf{x}_0$ is a stable equilibrium point.
ii. If $\dot{L}(\mathbf{x}) < 0$ for all points $\mathbf{x} \in D$ and in a neighbourhood of $\mathbf{x}_0$ then $\mathbf{x}_0$ is an asymptotically stable equilibrium point.
iii. If $\dot{L}(\mathbf{x}) > 0$ for all points $\mathbf{x} \in D$ and in a neighbourhood of $\mathbf{x}_0$ then $\mathbf{x}_0$ is an unstable equilibrium point.

Here $\dot{L}(\mathbf{x})$ is the time derivative of the Lyapunov function $L(\mathbf{x})$. In order to extend the results of the theorem above to our dynamic system of vector differential equations let us suppose we have the system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$$

Let us select a Lyapunov function having the matrix quadratic form:

$$L(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}\mathbf{x}$$

where $\mathbf{x} \in \Re^{2n}$ and $\mathbf{Q}$ is a $2n \times 2n$ matrix. Differentiating with respect to time, we get the following:

$$\begin{aligned}\dot{L}(\mathbf{x}) =& \mathbf{x}^T\mathbf{Q}\,\dot{\mathbf{x}} + \dot{\mathbf{x}}^T\mathbf{Q}\mathbf{x} \\ =& \mathbf{x}^T\mathbf{Q}\mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{x}^T\mathbf{Q}\mathbf{x} \\ =& \mathbf{x}^T\mathbf{B}\mathbf{x}\end{aligned}$$

where

$$\mathbf{B} = \mathbf{Q}\mathbf{A} + \mathbf{A}^T\mathbf{Q}$$

Now to apply Theorem 3.1 we need only to determine the definiteness of the matrix $\mathbf{B}$. For the dynamic system (20) a natural choice for the Lyapunov equation is the energy function derived previously. However as noted before, the dynamic system is a damped system so this choice of Lyapunov equation ends up always negative since it signifies energy loss. Since the hypothesis of Theorem 3.1 requires $L(\mathbf{x}) > 0$, let us select the negative of the energy function instead as the Lyapunov equation. Conversely we could have written Theorem 3.1 in the negative sense and reverse the order of the inequalities. Nevertheless, we write our choice in the matrix quadratic form as follows:

$$\begin{aligned}L(\mathbf{x}) =& -\frac{1}{2}\left[-\mathbf{v}'_p{}^T\mathbf{H}_p^{-1}\mathbf{v}'_p + \boldsymbol{\tau}'{}_p^T\mathbf{H}_p\boldsymbol{\tau}'_p\right] \\ =& \frac{1}{2}\begin{bmatrix}\mathbf{v}'_p \\ \boldsymbol{\tau}'_p\end{bmatrix}^T\begin{bmatrix}\mathbf{H}_p{}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_p\end{bmatrix}\begin{bmatrix}\mathbf{v}'_p \\ \boldsymbol{\tau}'_p\end{bmatrix} \\ =& \mathbf{x}^T\mathbf{Q}\mathbf{x}\end{aligned}$$

where

$$\begin{aligned}\mathbf{x} =& \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{v}'_p \\ \boldsymbol{\tau}'_p\end{bmatrix} \\ \mathbf{Q} =& \begin{bmatrix}\mathbf{H}_p{}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_p\end{bmatrix}\end{aligned} \tag{25}$$

The derivative of $\mathbf{x}$ with respect to time, t is found using (19) and (14) to give:

$$\begin{aligned}\dot{\mathbf{x}} =& \frac{1}{\sqrt{2}}\begin{bmatrix}\frac{d\mathbf{v}'_p}{dt} \\ \frac{d\boldsymbol{\tau}'_p}{dt}\end{bmatrix} \\ =& \frac{1}{\sqrt{2}}\begin{bmatrix}-\mathbf{v}'_p - \mathbf{H}_p\boldsymbol{\tau}'_p \\ -\mathbf{H}_p^{-1}(-\mathbf{v}'_p - \mathbf{H}_p\boldsymbol{\tau}'_p)\end{bmatrix} \\ =& \frac{1}{\sqrt{2}}\begin{bmatrix}-\mathbf{I}_p & -\mathbf{H}_p \\ \mathbf{H}_p^{-1} & \mathbf{I}_p\end{bmatrix}\begin{bmatrix}\mathbf{v}'_p \\ \boldsymbol{\tau}'_p\end{bmatrix} = \mathbf{A}\mathbf{x}\end{aligned}$$

where $\mathbf{I}_p$ is the identity matrix of appropriate dimensions. We now compute $\mathbf{B}$ as follows:

$$\begin{aligned}\mathbf{B} =& \mathbf{Q}\mathbf{A} + \mathbf{A}^T\mathbf{Q} \\ =& \begin{bmatrix}\mathbf{H}_p{}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_p\end{bmatrix}\begin{bmatrix}-\mathbf{I}_p & -\mathbf{H}_p \\ \mathbf{H}_p^{-1} & \mathbf{I}_p\end{bmatrix} \\ &+\begin{bmatrix}-\mathbf{I}_p & \mathbf{H}_p^{-1} \\ -\mathbf{H}_p & \mathbf{I}_p\end{bmatrix}\begin{bmatrix}\mathbf{H}_p{}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_p\end{bmatrix} \\ =& -2\begin{bmatrix}\mathbf{H}_p{}^{-1} & \mathbf{I}_p \\ \mathbf{I}_p & \mathbf{H}_p\end{bmatrix}\end{aligned} \tag{26}$$

The matrix $\mathbf{B}$ in (26) is negative semi-definite if the sub-matrix $\mathbf{H}_p$ is positive semi-definite or the kernel function selected is positive semi-definite. According to Theorem 3.1(i) the sub-system defined by the differential equation (20) has a stable equilibrium point at the origin. If the kernel function is positive definite then the system has an asymptotically stable equilibrium point. Otherwise, the system is unstable. Furthermore, at the equilibrium point the energy of the system is zero which corresponds to zero change in the objective function as expected at convergence. It is not too difficult to extend this to the entire problem by viewing it as a larger system composed of smaller sub-systems. The problem is solved when this system reaches equilibrium which is equivalent to all possible sub-systems achieving equilibrium. Now since we have shown that an arbitrary sub-system is stable depending on the choice of kernel function, and if all sub-systems achieve equilibrium then by logical argument the entire system will be stable. In optimization sense, the main problem converges when solved by this form of decomposition.

IV. Discussion

We have presented a convergence result for the decomposition method using stability analysis of dynamical systems. The analysis has been generalized in the sense that it can be equivalently applied to other Support Vector Machine models e.g. regression which are also constrained optimization problems. It is interesting to note that our previous notion of potential overstep has an analogous energy interpretation as shown by (22) which acts in the opposite direction of energy loss. In optimization, we are generally interested in increasing the rate of convergence which translates to increasing the output of energy. This further reinforces our believe that taking into account the constraints when solving a sub-problem is a viable method for increasing the speed of the algorithm. In particular, we may want to select a sequence of sub-problems that minimize potential energy to maximize the energy loss.

In this work, the main assumption is that the duration T of the iteration is relatively large which in our opinion is valid since small T is of no interest to us when trying to increase the rate of convergence. However, it would be interesting to investigate sub-systems that are near unstable due to the choice of working set that causes **B** to be nearly positive definite or indefinite. The assumption made that all possible sub-systems are stable thus applies only to well behaved problems. Furthermore, the systems model we derived is coupled and the effects of coupling on the rate of convergence is also another question of interest. In short, the dynamics of a sub-system affects the states of all the other sub-systems and is worth studying further in order to strengthen the convergence result.

V. Conclusion

In this work, we modelled the decomposition method using a Newtonian update as a second order dynamic system. The vector differential equations are written in terms of the gradients of the sub-problem and the potential of overstep. We then showed that the change in objective function has a direct analogy with the energy of the dynamic system. The system was then shown to be asymptotically stable through the use of Lyapunov's direct method provided all possible sub-systems were asymptotically stable. Stability of the system is then taken to mean convergence of the optimization method.

References

[1] V. N. Vapnik, *The nature of statistical learning theory*, 2nd ed., ser. Statistics for engineering and information science. New York: Springer, 2000.

[2] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130–136.

[3] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods-Support Vector Learning*, B. Schlkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge MIT Press, 1998, pp. 185–208.

[4] T. Joachims, "Making large scale support vector machine learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Schlkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge , MIT Press, 1998, pp. 169–184.

[5] C.-W. Hsu and C.-J. Lin, "A simple decomposition method for support vector machines." *Machine Learning*, vol. 46, pp. 291–314. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/bsvm/

[6] C.-C. Chang, C.-W. Hsu, and C.-J. Lin, "The analysis of decomposition methods for support vector machines," *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 1003–1008, 2000.

[7] C.-J. Lin, "On the convergence of the decomposition method for support vector machines," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1288–1298, 2001.

[8] ——, "Linear convergence for a decomposition method for support vector machines," November 2001.

[9] P. Laskov, "Feasible direction decomposition algorithms for training support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 315–349, 2002. [Online]. Available: citeseer.ist.psu.edu/laskov01feasible.html

[10] S. S. Keerthi and E. G. Gilbert, "Convergence of a generalized SMO algorithm for SVM classifier design," *Machine Learning*, vol. 46, no. 1-3, pp. 351–360, 2002. [Online]. Available: citeseer.nj.nec.com/keerthi00convergence.html

[11] A. Shilton, M. Palaniswami, D. Ralph, and A. C. Tsoi, "Incremental training of support vector machines," *to appear in IEEE Transactions on Neural Networks*, 2005.

[12] D. Lai, M. Palaniswami, and N. Mani, "Fast linear stationary methods for automatically biased support vector machines," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, 2003, pp. 2060–2065.

[13] D. Lai, N. Mani, and M. Palaniswami, "Effect of constraints on sub-problem selection for solving support vector machines using space decomposition," in *The 6th International Conference on Optimization: Techniques and Applications (ICOTA6) accepted*, Ballarat,Australia, 2004.

[14] J. K. Hale and H. Koak, *Dynamics and bifurcations*, ser. Texts in applied mathematics ; 3. New York: Springer-Verlag, 1991.

[15] L. Perko, *Differential equations and dynamical systems*, ser. Texts in applied mathematics ; 7. New York: Springer-Verlag, 1991.

[16] E. Mosca, *Optimal, predictive, and adaptive control*. Englewood Cliffs, N.J.: Prentice Hall, 1995.

Author/s:
Lai, Daniel; SHILTON, ALISTAIR; Mani, N.; PALANISWAMI, MARIMUTHU

Title:
Stability Analysis of the Decomposition Method for solving Support Vector Machines

Date:
2005

Citation:
Lai, Daniel and Shilton, Alistair and Mani, N. and Palaniswami, Marimuthu (2005) Stability Analysis of the Decomposition Method for solving Support Vector Machines, in Proceedings, Seconnd International Conference on Intelligent Sensing and Information Processing, ICISIP05, Bangalore, India.

Publication Status:
Published

Persistent Link:
http://hdl.handle.net/11343/34080

File Description:
Stability Analysis of the Decomposition Method for solving Support Vector Machines