Solving systems of linear equations via gradient systems with discontinuous righthand sides: application to LS-SVM

Leonardo V. Ferreira*, Eugenius Kaszkurewicz and Amit Bhaya

***Abstract*— A gradient system with discontinuous righthand side that solves an underdetermined system of linear equations in the L_1 norm is presented. An upper bound estimate for finite time convergence to a solution set of the system of linear equations is shown by means of the Persidskii form of the gradient system and the corresponding non-smooth diagonal type Lyapunov function. This class of systems can be interpreted as a recurrent neural network and an application devoted to solving least squares support vector machines (LS-SVM) is used as an example.**

I. Introduction

This paper proposes the use of a gradient dynamical system to solve underdetermined systems of linear equations in the L_1 norm. The system of linear equations is associated to an unconstrained convex optimization problem, which has the same solution set as the linear system. The unconstrained optimization problem, in turn is mapped into a gradient system with a discontinuous righthand side, which can be considered as a neural network with discontinuous activation functions [1]. The advantage of using this class of gradient systems is that convergence to the solution set of the system of linear equations occurs in finite time, and an upper bound for the latter is easily obtained. In addition, hardware implementation of this class of systems is simple.

Gradient systems were used to solve optimization problems for the first time in [2], where a method for solving linear programming problems on an analog computer is presented. Since then, this approach has been widely used [3]–[7].

Convergence analysis is performed by means of a Persidskii form of the gradient system in conjunction with a diagonal type Lyapunov function [8], [9]. The approach used in the present paper has been already used in [10], [11], where Persidskii systems, together with the associated diagonal type Lyapunov functions were used to derive convergence conditions of discontinuous gradient systems that solve linear programming problems and in [12], where a class of Persidskii systems with discontinuous righthand sides is analysed.

This research was partially financed by Project Nos 140811/2002-8, 551863/2002-1, 471262/03-0 of CNPq and also by the agencies CAPES and FAPERJ.

* Corresponding author. Tel.: +55-21-2562-8080; fax: +55-21-2562-8081.

E-mail addresses: lvalente@coep.ufrj.br (L. V. Ferreira); eugenius@nacad.ufrj.br (E. Kaszkurewicz); amit@nacad.ufrj.br (A. Bhaya).

Department of Electrical Engineering, NACAD–COPPE / Federal University of Rio de Janeiro. P.O Box 68504, 21945-970, Rio de Janeiro, RJ, BRAZIL

The proposed gradient dynamical system can be solved using standard ODE software and this could be an advantage, when the number of unknowns is large. Furthermore, implementations of these standard ODE methods on parallel computers could be used, in order to make it possible to deal with large datasets. In addition, gradient dynamical systems can be implemented as an analog circuit using only resistors, amplifiers and switches, which is appropriate for real time processing using VLSI technology [1].

The dynamical system proposed in the present paper is suitable for solving classification problems using least squares support vector machines (LS-SVMs). Support vector machines (SVMs) [13] are a powerful tool to approach several classes of problems, such as pattern classification. Modifications to the original approach have been proposed, such as the ν-SVM model [14] and the LS-SVM model [15]. The latter model is particularly appropriate for the technique proposed in the present paper, since it is modeled as a system of linear equations. For a detailed presentation of LS-SVM see [15].

II. Mathematical formulation of the problem

In the text that follows, matrices and vectors are denoted by boldface capital and lower case letters, such as $\mathbf{A}$ and $\mathbf{b}$, respectively. Scalars are represented by italic lower case letters, such as x, and sets are represented by capital Greek letters such as Δ. The functions $f : \mathbb{R}^n \to \mathbb{R}^n$ represent diagonal type functions, defined as $f(\mathbf{x}) := (f_1(x_1), \dots, f_n(x_n))^T$. In the present paper we are interested in solving systems of algebraic linear equations of the form:

$$\mathbf{Ax} = \mathbf{b}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ has full row rank, $m \le n$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$.

The assumption $\text{rank}(\mathbf{A}) = m$ ensures that the system (1) admits at least one solution. The least absolute deviation or L_1 approach to solving the system of linear equations (1) is to solve the following unconstrained optimization problem:

$$\text{Minimize } E(\mathbf{x}) = \|\mathbf{r}(\mathbf{x})\|_1,\ \mathbf{x} \in \mathbb{R}^n, \tag{2}$$

where $\mathbf{r}(\mathbf{x}) := \mathbf{Ax} - \mathbf{b}$, and $\|\cdot\|_1$ denotes the L_1 norm of the argument. In the context of neural networks, the objective function E is referred to as a *computational energy function*.

Let $r_i : \mathbb{R}^n \to \mathbb{R}$ be the components of vector $\mathbf{r}$. The set $\Delta := \{\mathbf{x} : \mathbf{r}(\mathbf{x}) = \mathbf{0}\}$ is defined as:

$$\Delta = \bigcap_{i=1}^{m} \Delta_i;\ \Delta_i := \{\mathbf{x} : r_i(\mathbf{x}) = 0\}, \tag{3}$$

The minimum of the energy function E in (2) is zero, consequently, a solution of problem (2) is a vector $\mathbf{x}^* \in \mathbb{R}^n$ such that $\mathbf{x}^* \in \Delta$. Notice that E is convex as a function of $\mathbf{r}$ and its unique minimizer is the zero vector $\mathbf{r}^* = \mathbf{0}$. The optimization problem (2) is solved in the present paper by steepest descent, which gives a gradient system of the form:

$$\dot{\mathbf{x}} = -\mathbf{M}\nabla E(\mathbf{x}), \tag{4}$$

where $\mathbf{M} = \text{diag}(\mu_1, \ldots, \mu_m)$, $\mu_i > 0$ for all i, is a positive diagonal matrix, and in the context of neural networks is referred to as a *learning matrix*, which is used to improve convergence speed [1]. For E as in (2), the gradient system (4) is:

$$\dot{\mathbf{x}} = -\mathbf{M}\mathbf{A}^T \text{sgn}(\mathbf{r}), \tag{5}$$

where $\text{sgn}(\mathbf{r})$, for a vector $\mathbf{r} = (r_i, \ldots, r_m)$, is defined as $(\text{sgn}(r_1), \ldots, \text{sgn}(r_m))^T$, such that for all i:

$$\text{sgn}(r_i) \begin{cases} = 1, & \text{if } r_i > 0 \\ \in [-1, 1], & \text{if } r_i = 0 \\ = -1, & \text{if } r_i < 0 \end{cases}$$

Notice that the function $E(\mathbf{x})$ in (2) is nondifferentiable at each Δ_i, leading to the discontinuous righthand side of (5). The solutions of (5) are considered in the sense of Filippov [16], and the sets Δ_i are referred to as *surfaces of discontinuity*. This class of Persidskii systems is analysed in [12] and [17], where the results of [12] are extended to a more general class of Persidskii systems with discontinuous righthand sides.

According to Filippov's theory, when the trajectories of (5) are not confined to any surface of discontinuity, the usual definition of solutions of differential equations holds. Otherwise, a solution of (5) is an absolutely continuous vector function $\mathbf{x}(t)$, defined in an interval $\mathcal{I}$, such that for almost all t in $\mathcal{I}$ the differential inclusion $\dot{\mathbf{x}} \in -\partial E(\mathbf{x})$ is satisfied. The set $\partial E(\mathbf{x})$ is the subdifferential of E at $\mathbf{x}$ and each element of this set is a subgradient of E at $\mathbf{x}$. If E is differentiable at $\mathbf{x}$, then $\partial E(\mathbf{x})$ has a single element, which is the gradient of E at $\mathbf{x}$. Further details and properties of subdifferentials and subgradients of convex functions can be found, for instance, in [18], [19].

If the trajectories of (5) are confined to some surface of discontinuity Δ_i, this motion is said to be a *sliding motion* or, equivalently, the system is said to be in *sliding mode*. This is equivalent to saying that the motion occurs in the hyperplane tangent to the surface of discontinuity. Further details about sliding modes can be found in [20], [21].

III. Convergence Analysis

Convergence analysis is performed using a Persidskii form of the gradient system (5) in conjunction with the corresponding candidate diagonal type Lyapunov function. The Persidskii form of (5) is obtained by premultiplying (5) by the matrix $\mathbf{A}$. Observe that since $\dot{\mathbf{r}} = \mathbf{A}\dot{\mathbf{x}}$, from (5) we get:

$$\dot{\mathbf{r}} = -\mathbf{A}\ \mathbf{M}\ \mathbf{A}^T \text{sgn}(\mathbf{r}). \tag{6}$$

Let $\sqrt{\mathbf{M}} = \text{diag}(\sqrt{\mu_1}, \ldots, \sqrt{\mu_m})$, notice that the righthand side of (6) can be written as $-\mathbf{A}\ \sqrt{\mathbf{M}}\sqrt{\mathbf{M}}\ \mathbf{A}^T \text{sgn}(\mathbf{r})$. Using this notation, we can prove the following proposition.

Proposition 1: The Persidskii system (6) is equivalent to the original gradient system (5), in the sense that $\dot{\mathbf{r}} \equiv \mathbf{0}$ iff $\dot{\mathbf{x}} \equiv \mathbf{0}$.

Proof: If $\dot{\mathbf{x}} \equiv \mathbf{0}$, it is immediate that $\dot{\mathbf{r}} \equiv \mathbf{0}$. On the other hand, if $\dot{\mathbf{r}} \equiv \mathbf{0}$ then the vector $\sqrt{\mathbf{M}}\nabla E = \sqrt{\mathbf{M}}\mathbf{A}^T \text{sgn}(\mathbf{r})$ belongs to the null space $\mathcal{N}(\mathbf{A}\sqrt{\mathbf{M}})$ of matrix $\mathbf{A}\sqrt{\mathbf{M}}$, however $\sqrt{\mathbf{M}}\nabla E$ is a vector in the row space $\mathcal{R}(\sqrt{\mathbf{M}}\mathbf{A}^T)$ of $\mathbf{A}\sqrt{\mathbf{M}}$. Since $\mathcal{N}(\mathbf{A}\sqrt{\mathbf{M}}) \perp \mathcal{R}(\sqrt{\mathbf{M}}\,\mathbf{A}^T)$, the only possible solution for $\dot{\mathbf{r}} \equiv \mathbf{0}$ is $\sqrt{\mathbf{M}}\,\nabla E \equiv \mathbf{0}$ and, consequently $\dot{\mathbf{x}} \equiv \mathbf{0}$. ∎

Proposition 1 is necessary since it ensures that the convergence results derived for the Persidskii system (6) also hold for the original gradient system (5). Since system (6) has a discontinuous righthand side, we choose the following nonsmooth candidate diagonal type Lyapunov function [12]:

$$V(\mathbf{r}) = \sum_{i=1}^{m} \int_0^{r_i} \text{sgn}(\tau)\, d\tau. \tag{7}$$

Observe that i) $V(\mathbf{r}) > 0$ for $\mathbf{r} \neq \mathbf{0}$; ii) $V(\mathbf{r}) = 0$ if and only if $\mathbf{r} = \mathbf{0}$. The time derivative of V along the trajectories of (6) is given by $\dot{V} = \nabla V^T \dot{\mathbf{r}}$, i.e.,

$$\dot{V}(\mathbf{r}) = -\text{sgn}^T(\mathbf{r})\, \mathbf{A}\ \mathbf{M}\ \mathbf{A}^T \text{sgn}(\mathbf{r}). \tag{8}$$

Notice that since $\mathbf{A}$ has full row rank and $\mathbf{M}$ is positive definite, then $\mathbf{A}\ \mathbf{M}\ \mathbf{A}^T$ is also positive definite. Consequently, $\dot{V} \equiv 0$ if and only if $\text{sgn}(\mathbf{r}) \equiv \mathbf{0}$ implying $\dot{\mathbf{r}} \equiv \mathbf{0}$ and, from Proposition 1, $\dot{\mathbf{x}} \equiv \mathbf{0}$.

Theorem 1: The trajectories of system (5) converge, from any initial condition, to the solution set of the system of linear equations (1) in finite time and remain in this set thereafter. Moreover, the convergence time t_f satisfies the bound $t_f \leq (V(\mathbf{r}_0)/\lambda_{min}(\mathbf{AMA}^T))$, where $\mathbf{r}_0 := \mathbf{r}(\mathbf{x}_0)$.

Proof: Consider system (6), the time derivative (8) of (7) and the partition of the set Δ, defined in (3). Considering the solutions of (6) in the sense of Filippov, two situations must be considered– first, when the trajectories have not reached any Δ_i and second, when the trajectories have already reached some set Δ_i. The aim is to show that in both situations there exist a scalar $\varepsilon > 0$, such that $\dot{V} \leq -\varepsilon$ and, finally to show that Δ is an invariant set.

i) $\mathbf{x}(t) \notin \Delta_i$, for every i. In this case the trajectories are not confined to any surface of discontinuity and the solutions of (6) are considered in the usual sense. Since $\mathbf{A}$ has full row rank and $\mathbf{M}$ is a positive diagonal matrix, then the matrix $\mathbf{AMA}^T$ is positive definite, and using the Rayleigh principle and the fact that $\|\text{sgn}(\mathbf{r})\|_2 = m^2 \geq 1$ for $r_i \neq 0$, we can write:

$$\dot{V}(\mathbf{r}) \leq -\lambda_{min}(\mathbf{AMA}^T)m^2 \leq -\lambda_{min}(\mathbf{AMA}^T), \tag{9}$$

where $\lambda_{min}(\mathbf{AA}^T) > 0$ is the smallest eigenvalue of $\mathbf{AA}^T$.

ii) $\mathbf{x}(t) \in \Delta_i$, for some i and almost all t in an interval $\mathcal{I}$. In this case the trajectories are confined to an intersection of k sets Δ_i, $k < m$, resulting in a sliding motion in this intersection. Thus, the vectors $\mathbf{e}$ that discribe this motion are subgradients of E at $\mathbf{x}$, i.e., $\dot{\mathbf{r}} = -\mathbf{AMe}$, $\mathbf{e} \in \partial E(\mathbf{x})$, where $\mathbf{e} = \mathbf{A}^T\mathbf{s}$ and $\mathbf{s} = (s_1, \ldots, s_m)^T$, with $s_i \in [-1, 1]$, for every i [18]. Since there exists at least one index i such that $\mathbf{x} \notin \Delta_i$, then $\|\mathbf{s}\|_2^2 \geq 1$, and using (8) and the Rayleigh principle, we obtain the inequality (9).

Therefore from items i) and ii), we conclude that the trajectories of (5) converge to the set Δ in finite time. It remains to show that the trajectories remain in Δ, i.e., that Δ is an invariant set. If $\mathbf{x}(t) \in \Delta$, then $V(t) = 0$ and $\dot{V}(t) = 0$. If for some $t = T$ the state vector $\mathbf{x}(T)$ leaves some Δ_i, then $\dot{V}(T) < 0$ and $V(T) > 0$, which is a contradiction, since V is nonincreasing along the trajectories of (6). Thus the trajectories of (6) reach Δ and remain in this set. From Proposition 1, this result also holds for the original gradient system (5).

We need to obtain the bound for convergence time. From (9) we can write $V(t) \leq V(t_0) - \lambda_{min}(\mathbf{AMA}^T)\, t$, thus, the time t_f for $\mathbf{r}$ to reach zero does not exceed $V_0/\lambda_{min}(\mathbf{AMA}^T)$, concluding the proof. ■

IV. Support vector machines (SVMs)

Given two classes A and B, the problem of finding the best surface that separates the elements of the given classes can be solved by means of support vector machines. This is known as the *training phase* of the SVM. One of the main features of SVMs is that training is performed by solving a quadratic optimization problem with linear constraints, which ensures the existence of a unique separating surface.

The training of the SVM is performed by means of *training pairs*, each pair consisting of one element of one of the classes and a label, indicating which class the element belongs to. This type of training is referred to as *supervised training*. Consider the following training pairs:

$$(y_1, \mathbf{z}_1), \ldots, (y_N, \mathbf{z}_N), \quad y_i \in \{-1, +1\}, \tag{10}$$

where the vectors $\mathbf{z}_i$ belong to the input space and the scalars y_i define the position of the vectors $\mathbf{z}_i$ in relation to the surface that separates the classes, i.e., if $y_i = +1$ the vector $\mathbf{z}_i$ is located above the separating surface and if $y_i = -1$, this vector is located below the separating surface. If given a set of pairs as in (10), a single hyperplane can be chosen such that for all i, $y_i = \pm 1$, then the set of points $\{\mathbf{z}_i\}_{i=1}^N$ is said to be *linearly separable*. This is known as a binary classification problem [22].

Let classes A and B, not necessarily linearly separable, be labeled as $y_i = +1$ is $\mathbf{z}_i \in A$ and $y_i = -1$ if $\mathbf{z}_i \in B$. The problem of finding the best hyperplane $\Pi := \{\mathbf{u} : \mathbf{u}^T\mathbf{z} + c = 0\}$ that separates the elements of classes A and B is modeled by the following quadratic optimization problem [13]:

$$\begin{aligned} &\text{minimize}_{\mathbf{u},\mathbf{e},c} \quad \left(\frac{1}{2}\mathbf{u}^T\mathbf{u} + b\sum_{i=1}^N e_i^p\right) \\ &\text{subject to} \quad y_i(\mathbf{u}^T\mathbf{z}_i + c) \geq 1 - e_i, \\ &\qquad\qquad\quad e_i \geq 0, \quad i = 1, \ldots, N, \end{aligned} \tag{11}$$

where $b > 0$ is a parameter, p is a positive integer, $\mathbf{u}$, $\mathbf{z}_i \in \mathbb{R}^n$ and $e_i, \in \mathbb{R}$. The quantity $y_i(\mathbf{u}^T\mathbf{z}_i + c)$ is defined as the margin of the input $\mathbf{z}_i$ with respect to the hyperplane Π, and the slack variables e_i are introduced in order to provide tolerance to misclassifications. They are necessary because, whenever the classes are not linearly separable, the optimization problem (11) without the slack variables would be infeasible. In the case of linearly separable classes, the slack variables vanish. The hyperplane Π that solves problem (11) is the so called soft margin hyperplane, which, roughly speaking, maximizes the margin [22], [23]. For nonlinear classification, a feature function ϕ, that maps the input space into a higher dimensional space is introduced. In this case, the constraints of problem (11) become $y_i(\mathbf{u}^T\phi(\mathbf{z}_i) + c) \geq 1 - e_i$, $i = 1, \ldots, N$.

The traditional approach is to solve the dual of (11), since in this case, instead of the function ϕ, another class of functions, known as kernel functions and defined as $K(\mathbf{z}, \mathbf{z}_i) = \phi^T(\mathbf{z})\phi(\mathbf{z}_i)$ is used, with the advantage that it is not necessary to know the feature function ϕ. The feature function ϕ is defined implicitly by the kernel which is assumed to satisfy the Mercer conditions [13], [22], [23].

Least Squares Support Vector Machines (LS-SVM)

The LS-SVM model is a modification of the original SVM model (11), in which the inequality constraints are replaced by equality constraints. The LS-SVM is modeled by the following constrained optimization problem [15]:

$$\begin{aligned} &\text{minimize}_{\mathbf{u},\mathbf{e},c} \quad \left(\frac{1}{2}\mathbf{u}^T\mathbf{u} + \frac{b}{2}\sum_{i=1}^N e_i^2\right) \\ &\text{subject to} \qquad y_i(\mathbf{u}^T\phi(\mathbf{z}_i) + c) = 1 - e_i,\ i = 1, \ldots, N. \end{aligned} \tag{12}$$

The dual problem of (12) is given by the following system of linear equations, also known as a KKT linear system [15]:

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \mathbf{Q} + b^{-1}\mathbf{I} \end{array}\right] \left[\begin{array}{c} c \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{1} \end{array}\right], \tag{13}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N$ is the vector of dual variables, $\mathbf{y}$ is the column vector composed of the labels of the classes, $\mathbf{1}$ is a N-dimensional column vector of ones, $\mathbf{Q}$ is a symmetric matrix given by $q_{ij} = y_i y_j K(\mathbf{z}_i, \mathbf{z}_j)$ and K is defined by the kernel $K(\mathbf{z}, \mathbf{z}_j) = \phi^T(\mathbf{z})\,\phi(\mathbf{z}_j)$, that must satisfy the Mercer conditions, meaning that the kernel K is positive-definite and so is matrix $\mathbf{Q}$ [15], [22].

Notice that since $b > 0$, $\mathbf{y} \neq \mathbf{0}$ and $\mathbf{Q}$ is symmetric positive-definite, it follows that the coefficient matrix of the KKT linear system (13) is nonsingular and, consequently, the solution of (13) is unique.

In the LS-SVM model, the problem of determining the best separating surface for classes A and B is reduced to solving the system of linear equations (13), and since it is in the form of equation (1), it can be solved using the gradient system (5). Theorem 1 ensures that the trajectories of the gradient system (5) converge in finite time to the solution of (13).

Application example: The example is the Iris plants database, taken from the UCI Repository of Machine Learning [24], that consists of three classes with 50 elements each. The first class is linearly separable from the other two, but the latter are not linearly separable. In the training phase we used 80 elements from the first and second classes, and the remaining ones were used for validation by means of the following discrimination function [15]:

$$y(\mathbf{z}) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{K}(\mathbf{z}, \mathbf{z}_i) + c\right). \qquad (14)$$

In this example the linear system consists of 81 variables and 81 equations, which implies that the gradient system has 81 differential equations. We used the Gaussian kernel, given by $K(\mathbf{z}, \mathbf{z}_i) = \exp(-\|\mathbf{z} - \mathbf{z}_i\|^2/\sigma^2)$, where σ is a scalar. This kernel is positive definite, and so is the matrix $\mathbf{Q}$. This kernel is widely used in the SVM literature.

The initial conditions were chosen arbitrarily at the origin. We used the Gaussian kernel with $\sigma = 1$, the LS-SVM parameter was $b = 10$ and the learning matrix of (5) was $\mathbf{M} = 10\,\mathbf{I}$. Some trajectories are shown in figure 1 and the use of the discrimination function (14) with the data points not used in the training showed that classification was done without errors, illustrating the good generalization capacity of the LS-SVM and the efficiency of the gradient system (5) for this application.

Speed of convergence to the solution of the linear system is proportional to the gain $\mu = 10$ ($\mathbf{M} = 10\mathbf{I}$): the larger the μ, the quicker the convergence. This is shown in Theorem 1, and the finite time estimate, which is an upper bound estimate for the time taken by $\mathbf{r}$ to become zero is given by $t_f \leq N/10\,\lambda_{min}(\mathbf{A}\mathbf{A}^T)$, where $\mathbf{A}$ denotes the coefficient matrix of system (13).

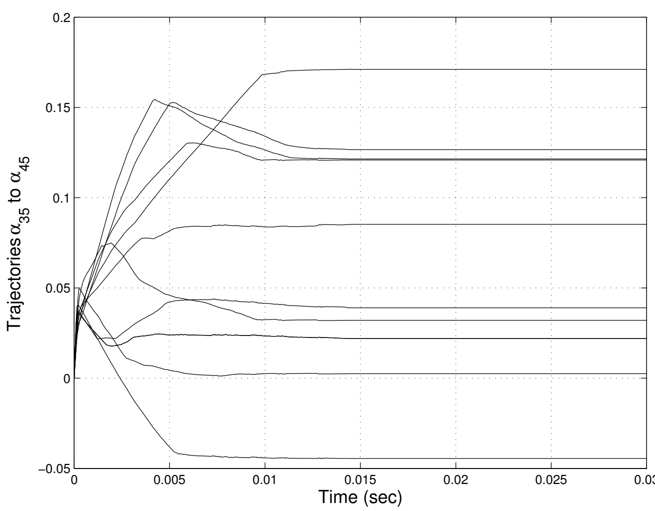

Fig. 1. Some trajectories, arbitrarily chosen, of the gradient system (5) for the Iris example, showing finite time convergence to the unique solution of the system of linear equations (13).

V. Concluding remarks

In the present paper we proposed a dynamical gradient system to find a solution set of systems of linear equations in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$. We proved finite time convergence to the solution set by means of a diagonal type Lyapunov function for the dynamical system, represented in a Persidskii form.

The proposed gradient system was applied to the training phase of the LS-SVM model, which amounts to solving a system of linear equations with unique solution. The LS-SVM model has, in comparison with other SVM models, the advantage that the dual of the optimization problem that models the LS-SVM is a system of linear equations in the form (1). This makes the LS-SVM particularly appropriate for the approach proposed in the present paper.

The number of parameters in the system of ODEs is small, consisting only of the learning matrix $\mathbf{M}$, that controls the transient time, making the implementation of the gradient system (5) by means of analog integrated circuits simple and suitable for real time processing. Moreover, the proposed gradient system can be solved using standard ODE software and this could be an advantage with respect to other methods (e.g. for LS-SVM classifiers), when the number of unknowns is large. In addition, finite time convergence to the solution of the system linear equations is ensured.

Acknowledgement

The authors would like to thank Prof. J. A. K. Suykens for conversations during IJCNN'03 and '04 that encouraged the authors to carry out the research reported in this letter.

References

[1] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*. John Wiley and Sons, New York, NY, 1993.
[2] I. B. Pyne, "Linear programming on an electronic analogue computer," *Trans. AIEE*, vol. 75, pp. 139–143, May 1956.
[3] M. V. Rybashov, "Gradient methods of solving linear and quadratic programming problems on electronic analog computers," *Automation and Remote Control*, vol. 26, no. 12, pp. 2151–2162, December 1965.
[4] N. N. Karpinskaya, "Method of "penalty" functions and the foundations of Pyne's method," *Automation and Remote Control*, vol. 28, no. 1, pp. 124–129, January 1967.
[5] A. Rodríguez-Vázquez, A. R. Rafael Domínguez-Castro, J. Huertas, and E. Sanchez-Sinencio, "Nonlinear switched-capacitor neural networks for optimization problems," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 384–398, 1990.
[6] M. P. Glazos, S. Hui, and S. H. Żak, "Sliding modes in solving convex programming problems," *SIAM Journal of Control and Optimization*, vol. 36, no. 2, pp. 680–697, March 1998.
[7] E. K. P. Chong, S. Hui, and S. H. Żak, "An analysis of a class of neural networks for solving linear programming problems," *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 1995–2006, 1999.
[8] S. K. Persidskii, "Problem of absolute stability," *Automation and Remote Control*, vol. 12, pp. 1889–1895, 1969.
[9] E. Kaszkurewicz and A. Bhaya, *Matrix Diagonal Stability in Systems and Computation*. Birkhäuser, Boston, MA, 2000.
[10] L. V. Ferreira, E. Kaszkurewicz, and A. Bhaya, "Convergence analysis of neural networks that solve linear programming problems," in *Procedings of the International Joint Conference on Neural Networks 2002, Honolulu, Hawaii, USA*, vol. 3, May 12-17 2002, pp. 2476–2481.
[11] ——, "Synthesis of a k-winners-take-all neural network using linear programming with bounded variables," in *Procedings of the International Joint Conference on Neural Networks 2003, Portland, OR, USA*, vol. 3, July 20-24 2003, pp. 2360–2365.
[12] L. Hsu, E. Kaszkurewicz, and A. Bhaya, "Matrix-theoretic conditions for the realizability of sliding manifolds," *Systems & Control Letters*, vol. 40, pp. 145–152, 2000.
[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[14] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, pp. 1245–2000, 2000.
[15] J. A. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.
[16] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academic Publishers, Dordrecht, 1988.

[17] L. V. Ferreira, E. Kaszkurewicz, and A. Bhaya, “Support vector classifiers via gradient systems with discontinuous righthand sides,” in *Procedings of the International Joint Conference on Neural Networks 2004, Budapest, Hungary*, vol. 4, July 25-29 2004, pp. 2997–3002.
[18] F. H. Clarke, *Optimization and Nonsmooth Analysis*. John Wiley and Sons, New York, NY, 1983.
[19] V. F. Dem’yanov and L. V. Vasil’ev, *Nondifferentiable Optimization*. Optimization Software, Inc., New York, NY, 1985.
[20] V. Utkin, *Sliding Modes in Control and Optimization*. Springer-Verlag, Berlin, 1992.
[21] C. Edwards and S. K. Spurgeon, *Sliding mode control: Theory and Applications*. Taylor & Francis, London, U.K., 1998.
[22] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, U.K., 2000.
[23] B. Schölkopf and A. Smola, *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.
[24] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases,” `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998, University of California, Irvine, Dept. of Information and Computer Sciences.