# On the Equality of Kernel AdaTron and Sequential Minimal Optimization in Classification and Regression Tasks and Alike Algorithms for Kernel Machines

Vojislav Kecman[1], Michael Vogt[2], Te Ming Huang[1]

[1] School of Engineering, The University of Auckland, Auckland, New Zealand
[2] Institute of Automatic Control, TU Darmstadt, Darmstadt,, Germany
*e-mail: v.kecman@auckland.ac.nz, mvogt@iat.tu-darmstadt.de*

**Abstract:** The paper presents the equality of a kernel AdaTron (KA) method (originating from a gradient ascent learning approach) and sequential minimal optimization (SMO) learning algorithm (based on an analytic quadratic programming step) in designing the support vector machines (SVMs) having positive definite kernels. The conditions of the equality of two methods are established. The equality is valid for both the nonlinear classification and the nonlinear regression tasks, and it sheds a new light to these seemingly different learning approaches. The paper also introduces other learning techniques related to the two mentioned approaches, such as the nonnegative conjugate gradient, classic Gauss-Seidel (GS) coordinate ascent procedure and its derivative known as the successive over-relaxation (SOR) algorithm as a viable and usually faster training algorithms for performing nonlinear classification and regression tasks. The convergence theorem for these related iterative algorithms is proven.

## 1. Introduction

One of the mainstream research fields in learning from empirical data by support vector machines, and solving both the classification and the regression problems, is an implementation of the incremental learning schemes when the training data set is huge. Among several candidates that avoid the use of standard quadratic programming (QP) solvers, the two learning approaches which have recently got the attention are the KA (Anlauf, Biehl, 1989; Frieß, Cristianini, Campbell, 1998; Veropoulos, 2001) and the SMO (Platt, 1998, 1999; Vogt, 2002). Due to its analytical foundation the SMO approach is particularly popular and at the moment the widest used, analyzed and still heavily developing algorithm. At the same time, the KA although providing similar results in solving classification problems (in terms of both the accuracy and the training computation time required) did not attract that many devotees. There are two basic reasons for that. First, until recently (Veropoulos, 2001), the KA seemed to be restricted to the classification problems only and second, it 'lacked' the fleur of the strong theory (despite its beautiful 'simplicity' and strong convergence proofs). The KA is based on a gradient ascent technique and this fact might have also distracted some researchers being aware of problems with gradient ascent approaches faced with possibly ill-conditioned kernel matrix. Here we show when and why the recently developed algorithms for SMO using positive definite kernels or models without a bias term, (Vogt, 2002), and the KA for both classification (Friess, Cristianini, Campbell, 1998) and regression (Veropoulos, 2001) are identical. Both the KA and the SMO algorithm attempt to solve the following QP problem in the case of *classification* (Vapnik, 1995; Cherkassky and Mullier, 1998; Cristianini and Shawe-Taylor, 2000; Kecman, 2001; Schölkopf and Smola, 2002) - *maximize* the dual Lagrangian

$$L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)\,, \qquad (1)$$

$$\text{subject to } \alpha_i \geq 0, \qquad i = 1, \ldots, l \qquad \text{and} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0\,. \qquad (2)$$

where $l$ is the number of training data pairs, $\alpha_i$ are the dual Lagrange variables, $y_i$ are the class labels (±1), and the $K(\mathbf{x}_i, \mathbf{x}_j)$ are the kernel function values. Because of noise or generic class' features, there will be an overlapping of training data points. Nothing, but constraints, in solving (1) changes and they are

$$0 \leq \alpha_i \leq C, \qquad i = 1, ..., l \qquad \text{and} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0\,, \qquad (3)$$

where $0 < C < \infty$, is a penalty parameter trading off the size of a margin with a number of misclassifications.

In the case of the *nonlinear regression* the learning problem is the maximization of a dual Lagrangian below

$$L_d(\alpha,\alpha^*) = -\varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)y_i - \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\mathbf{x}_i, \mathbf{x}_j)\,, (4)$$

$$\text{s.t.} \qquad \sum\nolimits_{i=1}^{l} \alpha_i^* = \sum\nolimits_{i=1}^{l} \alpha_i\,, \qquad (4a)$$

$$0 \leq \alpha_i^* \leq C, \qquad 0 \leq \alpha_i \leq C, \qquad i = 1, ..., l. \qquad (4b)$$

where $\varepsilon$ is a prescribed size of the insensitivity zone, and $\alpha_i$ and $\alpha_i^*$ ($i$ = 1, ..., $l$) are Lagrange multipliers for the points above and below the regression function respectively. Learning results in $l$ Lagrange multiplier *pairs* ($\alpha$, $\alpha^*$). Because no training data can be on both sides of the tube, either $\alpha_i$ or $\alpha_i^*$ will be nonzero, i.e., $\alpha_i\alpha_i^* = 0$.

## 2. The KA and SMO learning algorithms without-bias-term

It is known that *positive definite kernels* (such as the most popular and the most widely used RBF Gaussian kernels as well as the complete polynomial ones) do not require bias term (Evgeniou, Pontil, Poggio, 2000). Below, the KA and the SMO algorithms will be presented for such a fixed (i.e., no-) bias design problem and compared for the classification and regression cases. The equality of two learning schemes and resulting models will be established. Originally, in (Platt, 1998, 1999), the SMO *classification* algorithm was developed for solving the problem (1) including the constraints related to the bias $b$. In these early publications the case when bias $b$ is fixed variable was also mentioned but the detailed analysis of a fixed bias update was not accomplished.

## 2.1 Incremental Learning in Classification

### a) Kernel AdaTron in classification

The classic AdaTron algorithm as given in (Anlauf and Biehl, 1989) is developed for linear classifier. The KA is a variant of the classic AdaTron algorithm in the feature space of SVMs (Frieß et al., 1998). The KA algorithm solves the maximization of the dual Lagrangian (1) by implementing the gradient ascent algorithm. The update $\Delta\alpha_i$ of the dual variables $\alpha_i$ is given as

$$\Delta\alpha_i = \eta \frac{\partial L_d}{\partial \alpha_i} = \eta \left(1 - y_i \sum\nolimits_{j=1}^{l} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)\right) = \eta\left(1 - y_i f_i\right), \tag{5a}$$

where $f_i$ is the value of the decision function $f$ at the point $\mathbf{x}_i$, i.e., $f_i = \sum\nolimits_{j=1}^{l} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and $y_i$ denotes the value of the desired target (or the class' label) which is either +1 or -1. The update of the dual variables $\alpha_i$ is given as

$$\alpha_i \leftarrow \min(\max(0, \alpha_i + \Delta\alpha_i), C) \qquad (i = 1, ..., l) \tag{5b}$$

In other words, the dual variables $\alpha_i$ are clipped to zero if $(\alpha_i + \Delta\alpha_i) < 0$ . In the case of the soft nonlinear classifier ($C < \infty$) $\alpha_i$ are clipped between zero and $C$, ($0 \le \alpha_i \le C$). The algorithm converges from any initial setting for the Lagrange multipliers $\alpha_i$.

### b) SMO without-bias-term in classification

Recently (Vogt, 2002) derived the update rule for multipliers $\alpha_i$ that includes a detailed analysis of the Karush-Kuhn-Tucker (KKT) conditions for checking the optimality of the solution. (As referred above, a fixed bias update was mentioned in Platt's papers). The following update rule for $\alpha_i$ for a no-bias SMO algorithm was proposed

$$\Delta\alpha_i = -\frac{y_i E_i}{K(\mathbf{x}_i, \mathbf{x}_i)} = -\frac{y_i f_i - 1}{K(\mathbf{x}_i, \mathbf{x}_i)} = \frac{1 - y_i f_i}{K(\mathbf{x}_i, \mathbf{x}_i)}, \tag{6}$$

where $E_i = f_i - y_i$ denotes the difference between the value of the decision function $f$ at the point $\mathbf{x}_i$ and the desired target (label) $y_i$. Note the equality of (5a) and (6) when the learning rate in (5a) is chosen to be $\eta_i = 1/K(\mathbf{x}_i, \mathbf{x}_i)$ . The important part of the SMO algorithm is to check the KKT conditions with precision $\tau$ (e.g., $\tau = 10^{-3}$) in each step. An update is performed only if

$$\begin{aligned} &\alpha_i < C \ \wedge \ y_i E_i < -\tau, \ \text{or} \\ &\alpha_i > 0 \ \wedge \ y_i E_i > \tau \end{aligned} \tag{6a}$$

After an update, the same clipping operation as in (5b) is performed

$$\alpha_i \leftarrow \min(\max(0, \alpha_i + \Delta\alpha_i), C) \qquad (i = 1, ..., l) \tag{6b}$$

It is the nonlinear clipping operation in (5b) and in (6b) that strictly equals the KA and the SMO without-bias-term algorithm in solving nonlinear classification problems. This fact sheds new light on both algorithms. This equality is not that obvious in the case of a 'classic' SMO algorithm with bias term due to the heuristics involved in the selection of active points which should ensure the largest increase of the dual Lagrangian $L_d$ during the iterative optimization steps.

### 2.2 Incremental Learning in Regression

Similarly to the case of classification, there is a strict equality between the KA and the SMO algorithm when positive definite kernels are used for nonlinear regression.

**a) Kernel AdaTron in regression**

The first extension of the Kernel AdaTron algorithm for regression is presented in (Veropoulos, 2001) as the following gradient ascent update rules for $\alpha_i$ and $\alpha_i^*$

$$\Delta\alpha_i = \eta_i \frac{\partial L_d}{\partial \alpha_i} = \eta_i \left( y_i - \varepsilon - \sum\nolimits_{j=1}^{l} (\alpha_j - \alpha_j^*) K(\mathbf{x}_j, \mathbf{x}_i) \right) = \eta_i \left( y_i - \varepsilon - f_i \right) = -\eta_i \left( E_i + \varepsilon \right), \tag{7a}$$

$$\Delta\alpha_i^* = \eta_i \frac{\partial L_d}{\partial \alpha_i^*} = \eta_i \left( -y_i - \varepsilon + \sum\nolimits_{j=1}^{l} (\alpha_j - \alpha_j^*) K(\mathbf{x}_j, \mathbf{x}_i) \right) = \eta_i \left( -y_i - \varepsilon + f_i \right) = \eta_i \left( E_i - \varepsilon \right), \tag{7b}$$

where $y_i$ is the measured value for the input $\mathbf{x}_i$, $\varepsilon$ is the prescribed insensitivity zone, and $E_i = f_i - y_i$ stands for the difference between the regression function $f$ at the point $\mathbf{x}_i$ and the desired target value $y_i$ at this point. The calculation of the gradient above does not take into account the geometric reality that no training data can be on both sides of the tube. In other words, it does not use the fact that either $\alpha_i$ or $\alpha_i^*$ or both will be nonzero. i.e., that $\alpha_i\alpha_i^* = 0$ must be fulfilled in each iteration step. Below we derive the gradients of the dual Lagrangian $L_d$ accounting for geometry. This new formulation of the KA algorithm strictly equals the SMO method and it is given as

$$\begin{aligned} \frac{\partial L_d}{\partial \alpha_i} &= -K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i - \sum\nolimits_{j=1, j\neq i}^{l} (\alpha_j - \alpha_j^*) K(\mathbf{x}_j, \mathbf{x}_i) + y_i - \varepsilon + K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^* - K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^* \\ &= -K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^* - (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}_i) - \sum\nolimits_{j=1, j\neq i}^{l} (\alpha_j - \alpha_j^*) K(\mathbf{x}_j, \mathbf{x}_i) + y_i - \varepsilon \\ &= -K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^* + y_i - \varepsilon - f_i = -\left( K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^* + E_i + \varepsilon \right) \end{aligned} \tag{8a}$$

For the $\alpha_i^*$ multipliers, the value of the gradient is

$$\frac{\partial L_d}{\partial \alpha_i^*} = -K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i + E_i - \varepsilon \cdot \tag{8b}$$

The update value for $\alpha_i$ is now

$$\Delta\alpha_i = \eta_i \frac{\partial L_d}{\partial \alpha_i} = -\eta_i \left( K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^* + E_i + \varepsilon \right), \tag{9a}$$

$$\alpha_i \leftarrow \alpha_i + \Delta\alpha_i = \alpha_i + \eta_i \frac{\partial L_d}{\partial \alpha_i} = \alpha_i - \eta_i \left( K(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^* + E_i + \varepsilon \right) \tag{9b}$$

For the learning rate $\eta_i = 1/K(\mathbf{x}_i, \mathbf{x}_i)$ the gradient ascent learning KA is defined as,

$$\alpha_i \leftarrow \alpha_i - \alpha_i^* - \frac{E_i + \varepsilon}{K(\mathbf{x}_i, \mathbf{x}_i)} \tag{10a}$$

Similarly, the update rule for $\alpha_i^*$ is

$$\alpha_i^* \leftarrow \alpha_i^* - \alpha_i + \frac{E_i - \varepsilon}{K(\mathbf{x}_i, \mathbf{x}_i)} \tag{10b}$$

Same as in the classification, $\alpha_i$ and $\alpha_i^*$ are clipped between zero and $C$,

$$\alpha_i \leftarrow \min(\max(0, \alpha_i), C) \qquad (i = 1, ..., l), \tag{11a}$$

$$\alpha_i^* \leftarrow \min(\max(0, \alpha_i^*), C) \qquad (i = 1, ..., l). \tag{11b}$$

**b) SMO without-bias-term in regression**

The first algorithm for the SMO without-bias-term in regression (together with a detailed analysis of the KKT conditions for checking the optimality of the solution) is derived in (Vogt, 2002). The following learning rules for the Lagrange multipliers $\alpha_i$ and $\alpha_i^*$ updates were proposed

$$\alpha_i \leftarrow \alpha_i - \alpha_i^* - \frac{E_i + \varepsilon}{K(\mathbf{x}_i, \mathbf{x}_i)}, \tag{12a}$$

$$\alpha_i^* \leftarrow \alpha_i^* - \alpha_i + \frac{E_i - \varepsilon}{K(\mathbf{x}_i, \mathbf{x}_i)} \cdot \tag{12b}$$

The equality of equations (10a, b) and (12a, b) is obvious when the learning rate, as presented above in (10a, b), is chosen to be $\eta_i = 1/K(\mathbf{x}_i, \mathbf{x}_i)$. Thus, in both the classification and the regression, the optimal learning rate is not necessarily equal for all training data pairs. For a Gaussian kernel, $\eta = 1$ is same for all data points, and for a complete $n^{\text{th}}$ order polynomial each data point has different learning rate $\eta_i = 1/(\mathbf{x}_i^T\mathbf{x}_i + 1)^n$. Similar to classification, a joint update of $\alpha_i$ and $\alpha_i^*$ is performed only if the KKT conditions are violated by at least $\tau$, i.e. if

$$\begin{aligned} &\alpha_i < C \;\wedge\; \varepsilon + E_i < -\tau, \text{ or} \\ &\alpha_i > 0 \;\wedge\; \varepsilon + E_i > \tau, \text{ or} \\ &\alpha_i^* < C \;\wedge\; \varepsilon - E_i < -\tau, \text{ or} \\ &\alpha_i^* > 0 \;\wedge\; \varepsilon - E_i > \tau \end{aligned} \tag{13}$$

After the changes, the same clipping operations as defined in (11) are performed

$$\alpha_i \leftarrow \min(\max(0, \alpha_i), C) \qquad (i = 1, ..., l), \tag{14a}$$

$$\alpha_i^* \leftarrow \min(\max(0, \alpha_i^*), C) \qquad (i = 1, ..., l). \tag{14b}$$

The KA learning as formulated in this paper and the SMO algorithm without-bias-term for solving regression tasks are strictly equal in terms of both the number of iterations required and the final values of the Lagrange multipliers. The equality is strict despite the fact that the implementation is slightly different. In every iteration step, namely, the KA algorithm updates both weights $\alpha_i$ and $\alpha_i^*$ without any checking whether the KKT conditions are fulfilled or not, while the SMO performs an update according to equations (13).

## 3. The Coordinate Ascent Based Learning for Nonlinear Classification and Regression Tasks

When positive definite kernels are used, the learning problem for both tasks is same. In a vector-matrix notation, in a dual space, the learning is represented as:

*maximize* $$L_d(\alpha) = -0.5\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} + \mathbf{f}^T\boldsymbol{\alpha} \tag{15}$$

*s.t.* $$0 <= \alpha_i <= C, \qquad (i = 1, ..., n), \tag{16}$$

where, in the classification $n = l$ and the matrix $\mathbf{K}$ is an $(l, l)$ symmetric positive definite matrix, while in regression $n = 2l$ and $\mathbf{K}$ is a $(2l, 2l)$ symmetric semipositive definite one. Note that the constraints (16) define a convex subspace over which the convex dual Lagrangian should be maximized. It is very well known that the vector $\boldsymbol{\alpha}$ may be looked at as the solution of a system of linear equations

$$\mathbf{K}\boldsymbol{\alpha} = \mathbf{f} \tag{17}$$

subject to the same constraints as given by (16).

Thus, it may seem natural to solve (17), subject to (16), by applying some of the well known and established techniques for solving a general linear system of equations. The size of training data set and the constraints (16) eliminate direct techniques. Hence, one has to resort to the *iterative approaches* in solving the problems above. There are three possible iterative avenues that can be followed. They are; the use of the Non-Negative Least Squares (NNLS) technique (Lawson and Hanson, 1974), application of the Non-Negative Conjugate Gradient (NNCG) method (Hestenes, 1980) and the implementation of Gauss-Seidel (GS) i.e., the related Successive Over-Relaxation technique (SOR). The first two methods solve for the non-negative constraints only. Thus, they are not suitable in solving 'soft' tasks, when penalty parameter $C < \infty$ is used, i.e., when there is an upper bound on maximal value of $\alpha_i$. Nevertheless, in the case of nonlinear regression, one can apply NNLS and NNCG by taking $C = \infty$ and compensating (i.e. smoothing or 'softening' the solution) by increasing the sensitivity zone $\varepsilon$. However, the two methods (namely NNLS and NNCG) are not suitable for solving soft margin ($C < \infty$ ) classification problems in their present form, because there is no other parameter that can be used in 'softening' the margin.

Here we show how to extend the application of GS and SOR to both the nonlinear classification and to the nonlinear regression tasks. The Gauss-Seidel method solves (17) by using the $i^{\text{th}}$ equation to update the $i^{\text{th}}$ unknown doing it iteratively, i.e., starting in the $k^{\text{th}}$ step with the first equation to compute the $\alpha_1^{k+1}$, then the second equation is used to calculate the $\alpha_2^{k+1}$ by using new $\alpha_1^{k+1}$ and $\alpha_i^k$ $(i > 2)$ and so on. The iterative learning takes the following form,

$$\alpha_i^{k+1} = \left( f_i - \sum_{j=1}^{i-1} K_{ij}\alpha_j^{k+1} - \sum_{j=i+1}^{n} K_{ij}\alpha_j^{k} \right) / K_{ii} = \alpha_i^k - \frac{1}{K_{ii}}\left( \sum_{j=1}^{i-1} K_{ij}\alpha_j^{k+1} + \sum_{j=i}^{n} K_{ij}\alpha_j^{k} - f_i \right) = \alpha_i^k + \frac{1}{K_{ii}} \frac{\partial L_d}{\partial \alpha_i}\bigg|_{k+1} \tag{18}$$

where we use the fact that the term within a second bracket (called the residual $r_i$ in mathematics' references) is the $i^{\text{th}}$ element of the gradient of a dual Lagrangian $L_d$ given in (15) at the $k+1^{\text{th}}$ iteration step. The equation (18) above shows that GS method is a *coordinate* gradient ascent procedure as well as the KA and the SMO are. *The KA and SMO for positive definite kernels equal the GS!* Note that the optimal learning rate used in both the KA algorithm and in the SMO without-bias-term approach is exactly equal to the coefficient $1/K_{ii}$ in a GS method. Based on this equality, the convergence theorem for the KA, SMO and GS (i.e., SOR) in solving (15) subject to constraints (16) can be stated and proved as follows:

**Theorem:** For SVMs with positive definite kernels, the iterative learning algorithms KA i.e., SMO i.e., GS i.e., SOR, in solving nonlinear classification and regression tasks (15) subject to constraints (16), converge starting from any initial choice of $\boldsymbol{\alpha}_0$.

***Proof:*** The proof is based on the very well known theorem of convergence of the GS method for symmetric positive definite matrices in solving (17) without constraints (Ostrowski, 1966). First note that for positive definite kernels, the matrix **K** created by terms $y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ in the second sum in (1), and involved in solving classification problem, is also positive definite. In regression tasks **K** is a symmetric positive semidefinite (meaning still convex) matrix, which after a mild regularization given as ($\mathbf{K} \leftarrow \mathbf{K} + \lambda\mathbf{I}$, $\lambda$ ~ 1e-12) becomes positive definite one. (Note that the proof in the case of regression does not need regularization at all, but there is no space here to go into these details). Hence, the learning without constraints (16) converges, starting from any initial point $\boldsymbol{\alpha}_0$, and each point in an $n$-dimensional search space for multipliers $\alpha_i$ is a viable starting point ensuring a convergence of the algorithm to the maximum of a dual Lagrangian $L_d$. This, naturally, includes all the (starting) points within, or on a boundary of, any convex subspace of a search space ensuring the convergence of the algorithm to the maximum of a dual Lagrangian $L_d$ over the given subspace. The constraints imposed by (16) preventing variables $\alpha_i$ to be negative or bigger than $C$, and implemented by the clipping operators above, define such a convex subspace. Thus, each 'clipped' multiplier value $\alpha_i$ defines a new starting point of the algorithm guaranteeing the convergence to the maximum of $L_d$ over the subspace defined by (16). For a convex constraining subspace such a constrained maximum is unique. *Q.E.D.*

Due to the lack of the space we do not go into the discussion on the convergence rate here and we leave it to some other occasion. It should be only mentioned that both KA and SMO (i.e. GS and SOR) for positive definite kernels have been successfully applied for many problems (see references given here, as well as many other, benchmarking the mentioned methods on various data sets). Finally, let us just mention that the standard extension of the GS method is the method of successive over-relaxation that can reduce the number of iterations required by proper choice of relaxation parameter ω significantly. The SOR method uses the following updating rule

$$\alpha_i^{k+1} = \alpha_i^k - \omega \frac{1}{K_{ii}} \left( \sum_{j=1}^{i-1} K_{ij} \alpha_j^{k+1} + \sum_{j=i}^{n} K_{ij} \alpha_j^k - f_i \right) = \alpha_i^k + \omega \frac{1}{K_{ii}} \frac{\partial L_d}{\partial \alpha_i} \bigg|_{k+1} \tag{19}$$

and similarly to the KA, SMO, and GS its convergence is guaranteed.

## 4. Conclusions

Both the KA and the SMO algorithms were recently developed and introduced as alternatives to solving quadratic programming problem while training support vector machines on huge data sets. It was shown that when using positive definite kernels the two algorithms are identical in their analytic form and numerical implementation. In addition, for positive definite kernels both algorithms are strictly identical with a classic iterative GS (optimal coordinate ascent) learning and its extension SOR. Till now, these facts were blurred mainly due to different pace in posing the learning problems and due to the 'heavy' heuristics involved in an SMO implementation that shadowed an insight into the possible identity of the methods. It is shown that in the so-called no-bias SVMs, both the KA and the SMO procedure are the coordinate ascent based methods. Finally, due to the many ways how all the three algorithms (KA, SMO and GS i.e., SOR) can be implemented there may be some differences in their overall behaviour. The introduction of the relaxation parameter $0 < \omega < 2$ will speed up the algorithm. The exact optimal value $\omega_{opt}$ is problem dependent.

**Acknowledgment:** The results presented are initiated during the stay of the first author at the Prof. Rolf Isermann's Institute and sponsored by the Deutsche Forschungsgemeinschaft (DFG). He is thankful to both Prof. Rolf Isermann and DFG for all the support during this stay.

## 5. References

1. Anlauf, J. K., Biehl, M., The AdaTron - an adaptive perceptron algorithm. *Europhysics Letters*, 10(7), pp. 687–692, 1989
2. Cherkassky, V., Mulier, F., *Learning From Data: Concepts, Theory and Methods*, John Wiley & Sons, New York, NY, 1998
3. Cristianini, N., Shawe-Taylor, J., *An introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, Cambridge, UK, 2000
4. Evgeniou, T., Pontil, M., Poggio, T., Regularization networks and support vector machines, Advances in Computational Mathematics, 13, pp.1-50, 2000.
5. Frieß, T.-T., Cristianini, N., Campbell, I. C. G., The Kernel-Adatron: a Fast and Simple Learning Procedure for Support Vector Machines. In Shavlik, J., editor, *Proceedings of the 15th International Conference on Machine Learning*, Morgan Kaufmann, pp. 188–196, San Francisco, CA, 1998
6. Kecman V., *Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, The MIT Press, Cambridge, MA, (http://www.support-vector.ws), 2001
7. Lawson, C. I., Hanson, R. J., *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, N.J., 1974
8. Ostrowski, A.M., *Solutions of Equations and Systems of Equations*, 2nd ed., Academic Press, New York, 1966
9. Platt, J. C., Sequential minimal optimization: A fast algorithm for training support vector machines. TR MSR-TR-98-14, Microsoft Research, 1998
10. Platt, J.C., Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Ch. 12 in Advances in Kernel Methods – Support Vector Learning*, edited by B. Schölkopf, C. Burges, A. Smola, The MIT Press, Cambridge, MA, 1999
11. Schölkopf B., Smola, A., *Learning with Kernels – Support Vector Machines, Optimization, and Beyond*, The MIT Press, Cambridge, MA, 2002
12. Veropoulos, K., *Machine Learning Approaches to Medical Decision Making*, PhD Thesis, The University of Bristol, Bristol, UK, 2001
13. Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer Verlag Inc, New York, NY, 1995
14. Vogt, M., SMO Algorithms for Support Vector Machines without Bias, Institute Report, Institute of Automatic Control, TU Darmstadt, Darmstadt, Germany, (http://w3.rt.e-technik.tu-darmstadt.de/~vogt/), 2002