MinOver Revisited for Incremental Support-Vector-Classification

Thomas Martinetz

Institute for Neuro- and Bioinformatics
University of Lübeck
D-23538 Lübeck, Germany
martinetz@informatik.uni-luebeck.de
http://www.inb.uni-luebeck.de

Abstract. The well-known and very simple MinOver algorithm is reformulated for incremental support vector classification with and without kernels. A modified proof for its $\mathcal{O}(t^{-1/2})$ convergence is presented, with t as the number of training steps. Based on this modified proof it is shown that even a convergence of at least $\mathcal{O}(t^{-1})$ is given. This new convergence bound for MinOver is confirmed by computer experiments on artificial data sets. The computational effort per training step scales as $\mathcal{O}(N)$ with the number N of training patterns.

1 Introduction

The Support-Vector-Machine (SVM) [1], [12] is an extremely successful concept for pattern classification and regression and has found widespread applications (see, e.g. [6], [9], [11]). It became a standard tool like Neural Networks or classical approaches. A major drawback, particularly for industrial applications where easy and robust implementation is an issue, is its complicated training procedure. A large Quadratic-Programming problem has to be solved, which requires numerical optimization routines which many users do not want or cannot implement by themselves. They have to rely on existing software packages which are hardly comprehensive and, in many cases at least, error-free. This is in contrast to most Neural Network approaches where learning has to be simple and incremental almost by definition.

For this reason a number of different approaches to obtain more or less simple and incremental SVM training procedures have been introduced [2], [3], [10], [4], [7]. We will revisit the MinOver algorithm which was introduced by Krauth and Mézard [5] for spin-glass models of Neural Networks. As a slight modification of the perceptron algorithm, it is well-known that MinOver can be used for maximum margin classification. In spite of the fact that a training procedure can hardly be more simple, and in spite of the fact that advantageous learning behaviour has been reported [8], so far it has not become a standard training algorithm for maximum margin classification. To make MinOver more attractive we give a simplified formulation of this algorithm and show that, in contrast to the $\mathcal{O}(t^{-1/2})$ convergence bound given in [5], in fact one can expect a $\mathcal{O}(t^{-1})$ convergence, with t as the number of learning steps.

1.1 The Problem

Given a linearly separable set of patterns $\mathbf{x}_\nu \in \mathbb{R}^D$, $\nu = 1, \ldots, N$ with corresponding class labels $y_\nu \in \{-1, 1\}$. We want to find the hyperplane which separates the patterns of these two classes with maximum margin. The hyperplane for classification is determined by its normal vector $\mathbf{w} \in \mathbb{R}^D$ and its offset $b \in \mathbb{R}$. It achieves a separation of the two classes, if

$$y_\nu(\mathbf{w}^T\mathbf{x}_\nu - b) > 0 \qquad \text{for all} \qquad \nu = 1, \ldots, N$$

is valid. The margin Δ of this separation is given by

$$\Delta = \min_\nu [y_\nu(\mathbf{w}^T\mathbf{x}_\nu - b)/||\mathbf{w}||].$$

For convenience we introduce $\mathbf{z}_\nu = y_\nu(\mathbf{x}_\nu, -1) \in \mathbb{R}^{D+1}$ and $\mathbf{v} = (\mathbf{w}, b) \in \mathbb{R}^{D+1}$. We look for the $\mathbf{v}$ which maximizes $\Delta(\mathbf{v}) = \min_\nu[\mathbf{v}^T\mathbf{z}_\nu/||\mathbf{w}||]$. With

$$d(\mathbf{v}) = \min_\nu [\mathbf{v}^T\mathbf{z}_\nu/||\mathbf{v}||]$$

we introduce the margin of separation of the augmented patterns $(\mathbf{x}_\nu, -1)$ in the $(D+1)$-space. The $\mathbf{v}$ which provides the maximum margin d_* in the $(D+1)$-space also provides the maximum margin Δ^* in the D-dimensional subspace of the original patterns $\mathbf{x}_\nu \in \mathbb{R}^D$. This is the case since (i) the $\mathbf{v}_*$ which provides Δ^* also provides at least a local maximum of $d(\mathbf{v})$ and (ii) $d(\mathbf{v})$ and $\Delta(\mathbf{v})$ are convex and both have only one global maximum. Therefore,

$$\begin{aligned}\mathbf{v}_* = (\mathbf{w}_*, b_*) &= \arg \max_{||\mathbf{v}||=1} [\min_\nu (\mathbf{v}^T\mathbf{z}_\nu)/||\mathbf{w}||] \\ &= \arg \max_{||\mathbf{v}||=1} [\min_\nu (\mathbf{v}^T\mathbf{z}_\nu)]\end{aligned}$$

is valid. Instead of looking for the $\mathbf{v}_*$ which provides the maximum Δ, we look for the $\mathbf{v}_*$ which provides the maximum d. Both $\mathbf{v}_*$ are identical. Since $||\mathbf{v}_*||^2 = ||\mathbf{w}_*||^2 + b_*^2 = 1$, we obtain Δ^* from d_* and $\mathbf{v}_* = (\mathbf{w}_*, b_*)$ through

$$\Delta^* = \frac{d_*}{||\mathbf{w}_*||} = \frac{d_*}{\sqrt{1 - b_*^2}}.$$

2 The MinOver Algorithm Reformulated

The well-known MinOver algorithm is a simple and iterative procedure which provides the maximum margin classification in linearly separable classification problems. It was introduced in [5] for spin-glass models of Neural Networks. The MinOver algorithm yields a vector $\mathbf{v}_t$ the direction of which converges against $\mathbf{v}_*$ with increasing number of iterations t. This is valid as long as a full separation, i.e. a $\mathbf{v}_*$ with $\Delta^* > 0$ exists.

The MinOver algorithm works like the perceptron algorithm, with the slight modification that for training always the pattern $\mathbf{z}_\alpha(t)$ out of the training set $\mathcal{T} = \{\mathbf{z}_\nu | \nu =$

$1, \ldots, N\}$ with the worst, i.e. the minimum residual margin (overlap) $\mathbf{v}^T\mathbf{z}_\nu$ is chosen. Hence, the name MinOver.

Compared to [5] we present a simplified formulation of the MinOver algorithm, with the number of desired iterations t_{max} prespecified:

0. Set $t = 0$, choose a t_{max}, and set $\mathbf{v}_{t=0} = 0$.
1. Determine the $\mathbf{z}_\alpha(t)$ out of the training set $\mathcal{T}$ for which $\mathbf{v}_t^T\mathbf{z}$ is minimal.
2. Set $\mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{z}_\alpha(t)$.
3. Set $t = t + 1$ and go to 1.) if $t < t_{max}$.

2.1 MinOver in its Dual Formulation and with Kernels

The vector $\mathbf{v}_t$ which determines the dividing hyperplane is given by

$$\begin{aligned}\mathbf{v}_t &= \sum_{\tau=0}^{t-1} \mathbf{z}_\alpha(\tau) \\ &= \sum_{\mathbf{z}_\nu \in \mathcal{V}_t} n_\nu(t)\mathbf{z}_\nu\end{aligned}$$

with $\mathcal{V}_t \subseteq \mathcal{T}$ as the set of all patterns which have been used for training so far. The coefficient $n_\nu(t) \in \mathbb{N}$ denotes the number of times $\mathbf{z}_\nu \in \mathcal{V}_t$ has been used for training up to time step t. $\sum_{\mathcal{V}_t} n_\nu(t) = t$ is valid. With $V_t = |\mathcal{V}_t| \leq t$ we denote the number of training patterns which determine the normal vector $\mathbf{v}_t$.

In the dual representation the expression which decides the class assignment by being smaller or larger than zero can be written as

$$\mathbf{v}^T\mathbf{z} = \sum_{\mathbf{x}_\nu \in \mathcal{V}} n_\nu y_\nu (\mathbf{x}_\nu^T\mathbf{x}) - b \tag{1}$$

with

$$b = \sum_{y_\nu \in \mathcal{V}} n_\nu y_\nu. \tag{2}$$

In the dual formulation the training of the MinOver algorithm consists of either adding the training pattern $\mathbf{z}_\alpha$ to $\mathcal{V}$ as a further data point or, if $\mathbf{z}_\alpha$ is already element of $\mathcal{V}$, to increase the corresponding n_α by one.

If the input patterns $\mathbf{x} \in \mathbb{R}^D$ are transformed into another (usually higher dimensional) feature space $\mathbf{\Phi}(\mathbf{x}) \in \mathbb{R}^{D'}$ before classification, MinOver has to work with $\mathbf{z}_\nu = y_\nu(\mathbf{\Phi}(\mathbf{x}_\nu), -1)^T$. Due to Equation (1) it does not have to do it explicitly. With $K(\mathbf{x}_\nu, \mathbf{x}) = \mathbf{\Phi}^T(\mathbf{x}_\nu)\mathbf{\Phi}(\mathbf{x})$ as the kernel which corresponds to the transformation $\mathbf{\Phi}(\mathbf{x})$, we obtain

$$\mathbf{v}^T\mathbf{z} = y\left(\sum_{\mathbf{x}_\nu \in \mathcal{V}} n_\nu y_\nu K(\mathbf{x}_\nu, \mathbf{x}) - b\right), \tag{3}$$

with the b of Equation (2).

In its dual formulation the MinOver algorithm is simply an easy procedure of selecting data points out of the training set:

0. Set $t = 0$, choose a t_{max}, and set $\mathcal{V} = \emptyset$.
1. Determine the $\mathbf{z}_\alpha(t)$ out of the training set $\mathcal{T}$ for which $\mathbf{v}_t^T \mathbf{z}$ according to Equation (3) is minimal.
2. If $\mathbf{z}_\alpha(t) \notin \mathcal{V}$, add $\mathbf{z}_\alpha(t)$ to $\mathcal{V}$ and assign to it an $n_\alpha = 1$. If $\mathbf{z}_\alpha(t) \in \mathcal{V}$ already, increase its n_α by one.
3. Set $t = t + 1$ and go to 1.) if $t < t_{max}$.

3 Convergence Bounds for MinOver

Krauth and Mézard gave a convergence proof for MinOver [5]. Within the context of spin-glass Neural Networks they showed that the smallest margin $d_t = \mathbf{v}_t^T \mathbf{z}_\alpha(t)$ provided by $\mathbf{v}_t$ converges against the maximum margin d_* at least as $\mathcal{O}(t^{-1/2})$. We give a modified proof of this $\mathcal{O}(t^{-1/2})$ convergence. Based on this proof we show that the margin converges even at least as $\mathcal{O}(t^{-1})$ against the maximum margin.

3.1 $\mathcal{O}(t^{-1/2})$ Bound

We look at the convergence of the angle γ_t between $\mathbf{v}_t$ and $\mathbf{v}_*$. We decompose the learning vector $\mathbf{v}_t$ into

$$\mathbf{v}_t = \cos\gamma_t ||\mathbf{v}_t|| \mathbf{v}_* + \mathbf{u}_t \qquad \mathbf{u}_t \mathbf{v}_* = 0. \tag{4}$$

$||\mathbf{u}_t|| \leq R\sqrt{t}$ is valid, with R as the norm of the augmented pattern with maximum length, i.e., $R = \max_\nu ||\mathbf{z}_\nu||$. This can be seen from

$$\begin{aligned} \mathbf{u}_{t+1}^2 - \mathbf{u}_t^2 &= (\mathbf{u}_t + \mathbf{z}_\alpha(t) - [\mathbf{z}_\alpha(t)\mathbf{v}_*]\mathbf{v}_*)^2 - \mathbf{u}_t^2 \\ &= 2\mathbf{u}_t^T \mathbf{z}_\alpha(t) + \mathbf{z}_\alpha(t)^2 - [\mathbf{z}_\alpha(t)^T \mathbf{v}_*]^2 \\ &\leq R^2. \end{aligned} \tag{5}$$

We have used $\mathbf{u}_t^T \mathbf{z}_\alpha(t) \leq 0$. Otherwise the condition

$$\min_\nu \frac{(\lambda \mathbf{v}_* + \mathbf{u}_t)^T \mathbf{z}_\mu}{||\lambda \mathbf{v}_* + \mathbf{u}_t||} \leq \Delta^* \qquad \forall \lambda \in \mathbb{R}$$

would be violated. Since also

$$\mathbf{v}_*^T \mathbf{v}_t = \mathbf{v}_*^T \sum_{\tau=0}^{t-1} \mathbf{z}_\alpha(\tau) \geq d_* t$$

is valid, we obtain the bounds

$$\sin\gamma_t \leq \gamma_t \leq \tan\gamma_t = \frac{||\mathbf{u}_t||}{\mathbf{v}_*^T \mathbf{v}_t} \leq \frac{R\sqrt{t}}{d_* t} = \frac{R/d_*}{\sqrt{t}} \,. \tag{6}$$

The angle γ between the hyperplane provided by MinOver and the maximum margin hyperplane converges to zero at least as $\mathcal{O}(t^{-1/2})$.

After a finite number of training steps the $\mathbf{z}_\alpha(t)$ selected for learning will always be support vectors with $d_* = \mathbf{v}_*^T \mathbf{z}_\alpha(t)$. This can be seen from the following arguments: with Equation (4) we obtain

$$d_* \geq \frac{\mathbf{v}_t^T \mathbf{z}_\alpha(t)}{||\mathbf{v}_t||} = \mathbf{v}_*^T \mathbf{z}_\alpha(t) \cos\gamma_t + \frac{\mathbf{u}_t^T \mathbf{z}_\alpha(t)}{||\mathbf{u}_t||} \sin\gamma_t. \tag{7}$$

If $\mathbf{z}_\alpha(t)$ is not a support vector, $\mathbf{v}_*^T \mathbf{z}_\alpha(t) > d_*$ is valid. Since the prefactor of the sinus is bounded, after a finite number of training steps the right hand side would be larger than d_*. Hence, after a finite number of learning steps the $\mathbf{z}_\alpha(t)$ can only be support vectors.

Equation (7) yields the convergence of d_t. We obtain

$$d_* \geq d_t \geq d_* \cos\gamma_t - R \sin\gamma_t \geq d_*(1 - \gamma_t^2/2) - R\gamma_t \ .$$

With the term leading in γ_t and with our upper bound for γ_t, the convergence of the margin with increasing t is bounded by

$$0 \leq \frac{d_* - d_t}{d_*} \leq \frac{R}{d_*}\gamma_t \leq \frac{R^2/d_*^2}{\sqrt{t}} \ .$$

3.2 $\mathcal{O}(t^{-1})$ Bound

From Equation (6) we can discern that we obtain a $\mathcal{O}(t^{-1})$ bound for the angle γ_t and, hence, a $\mathcal{O}(t^{-1})$ convergence of the margin d_t to the maximum margin d_*, if $||\mathbf{u}_t||$ remains bounded. This is indeed the case:

We introduced $\mathbf{u}_t$ as the projection of $\mathbf{v}_t$ onto the maximum margin hyperplane given by the normal vector $\mathbf{v}_*$. In addition we introduce $\mathbf{s}_\nu = \mathbf{z}_\nu - (\mathbf{v}_*^T \mathbf{z}_\nu)\mathbf{v}_*$ as the projection of the training patterns $\mathbf{z}_\nu$ onto the maximum margin hyperplane given by $\mathbf{v}_*$. As we have seen above, after a finite $t = t_{start}$ each $\mathbf{s}_\alpha(t)$ corresponds to one of the N_S support vectors. Then looking for the $\mathbf{z}_\alpha(t)$ out of the training set $\mathcal{T}$ for which $\mathbf{v}_t^T \mathbf{z}$ is minimal becomes equivalent to looking for the $\mathbf{s}_\alpha(t)$ out of the set $\mathcal{S}'$ of projected support vectors for which $\mathbf{u}_t^T \mathbf{z} = \mathbf{u}_t^T \mathbf{s}$ is minimal.

We now go one step further and introduce $\mathbf{u'}_t$ as the projection of $\mathbf{u}_t$ onto the subspace spanned by the $\mathbf{s}_\nu \in \mathcal{S}'$. This subspace is at most N_S-dimensional. Since $\mathbf{u}_t^T \mathbf{s}_\nu = \mathbf{u'}_t^T \mathbf{s}_\nu$ for the $\mathbf{s}_\nu \in \mathcal{S}'$, we now look for the $\mathbf{s}_\alpha(t) \in \mathcal{S}'$ for which $\mathbf{u'}_t^T \mathbf{s}$ is minimal. Note that for $t \geq t_{start}$ always $\mathbf{u}_t^T \mathbf{z}_\alpha(t) = \mathbf{u}_t^T \mathbf{s}_\alpha(t) = \mathbf{u'}_t^T \mathbf{s}_\alpha(t) \leq 0$ is valid.

The following analysis of the development of $\mathbf{u}$ over time starts with $\mathbf{u}_{t_{start}}$. We have

$$\mathbf{u}_t = \mathbf{u}_{t_{start}} + \sum_{\tau=t_{start}}^{t-1} \mathbf{s}_\alpha(\tau).$$

$\mathbf{u}_t$ remains bounded, if $\mathbf{u'}_t$ remains bounded. We discriminate the following three cases:

i) $\max_{||\mathbf{u'}||=1} \min_{\mathbf{s}_\nu \in \mathcal{S}'} (\mathbf{u'}^T \mathbf{s}_\nu) < 0$
ii) $\max_{||\mathbf{u'}||=1} \min_{\mathbf{s}_\nu \in \mathcal{S}', ||\mathbf{s}_\nu||>0} (\mathbf{u'}^T \mathbf{s}_\nu) > 0$
iii) $\max_{||\mathbf{u'}||=1} \min_{\mathbf{s}_\nu \in \mathcal{S}', ||\mathbf{s}_\nu||>0} (\mathbf{u'}^T \mathbf{s}_\nu) = 0$

Note that the vector $\mathbf{u}'$ with $||\mathbf{u}'|| = 1$ varies only within the subspace spanned by the $\mathbf{s}_\nu \in \mathcal{S}'$. If this subspace is of dimension one, only i) or ii) can occur. For i) and ii) it can quickly be proven that $\mathbf{u}'_t$ remains bounded. Case iii) can be redirected to i) or ii), which is a little bit more tedious.

i) There is an $\epsilon > 0$ such that for each training step $\mathbf{u}'^T_t \mathbf{s}_\alpha(t) \leq -\epsilon ||\mathbf{u}'_t||$. Analog to Equation (5) we obtain

$$\begin{aligned} \Delta \mathbf{u}'^2_t &= 2\mathbf{u}'^T_t \mathbf{s}_\alpha(t) + \mathbf{s}_\alpha(t)^2 \\ &\leq -2\epsilon ||\mathbf{u}'_t|| + R^2. \end{aligned}$$

The negative contribution to the change of $||\mathbf{u}'_t||$ with each training step increases with $||\mathbf{u}'_t||$ and keeps it bounded.

ii) There is a $\mathbf{u}'$ such that $\mathbf{u}'^T \mathbf{s}_\nu > 0$ for each $||\mathbf{s}_\nu|| > 0$. In this case there is a $\mathbf{s}_\nu \in \mathcal{S}'$ with $||\mathbf{s}_\nu|| = 0$, since always $\mathbf{u}'^T \mathbf{s}_\alpha(t) \leq 0$ has to be valid. If $\mathbf{s}_\alpha(t) = 0$, the change of the vector $\mathbf{u}'_t$ terminates, since also $\mathbf{s}_\alpha(t+1)$ will be zero. Will $\mathbf{s}_\alpha(t)$ be zero after a finite number of training steps? It will since there is a $\mathbf{u}'$ which separates the $||\mathbf{s}_\nu|| > 0$ from $\mathbf{s}_\nu = 0$ with a positive margin. We know from perceptron learning that in this case also $\mathbf{u}'_t$ will separate these $||\mathbf{s}_\nu|| > 0$ after a finite number of learning steps. At the latest when this is the case $\mathbf{s}_\alpha(t)$ will be zero and $||\mathbf{u}'_t||$ will stay bounded.

iii) We will redirect this case to i) or ii). With $\mathbf{u}'_*$ we denote the $\mathbf{u}'$, $||\mathbf{u}'|| = 1$ which maximizes $\min_{\mathbf{s}_\nu \in \mathcal{S}', ||\mathbf{s}_\nu|| > 0} (\mathbf{u}'^T \mathbf{s}_\nu)$. The set of those $\mathbf{s}_\nu \in \mathcal{S}'$ with $\mathbf{u}'^T_* \mathbf{s}_\nu = 0$ we denote by $\mathcal{S}''$. The $\mathbf{s}_\nu \in \mathcal{S}'/\mathcal{S}''$ are separated from the origin by a positive margin. After a finite number of learning steps $\mathbf{s}_\alpha(t)$ will always be an element of $\mathcal{S}''$. Then looking for the $\mathbf{s}_\alpha(t)$ out of $\mathcal{S}'$ for which $\mathbf{u}'^T_t \mathbf{s}$ is minimal becomes equivalent to looking for the $\mathbf{s}_\alpha(t)$ out of the set $\mathcal{S}''$ for which $\mathbf{u}''^T_t \mathbf{s}$ is minimal, with $\mathbf{u}''_t$ as the projection of $\mathbf{u}'_t$ onto the subspace spanned by the $\mathbf{s}_\nu \in \mathcal{S}''$. Note that the dimension of this subspace is reduced by at least one compared to the subspace spanned by the $\mathbf{s}_\nu \in \mathcal{S}'$. For $\mathbf{s}_\nu \in \mathcal{S}''$ again $\mathbf{u}_t^T \mathbf{z}_\alpha(t) = \mathbf{u}_t^T \mathbf{s}_\alpha(t) = \mathbf{u}'^T_t \mathbf{s}_\alpha(t) = \mathbf{u}''^T_t \mathbf{s}_\alpha(t) \leq 0$ is valid. $\mathbf{u}'$ remains bounded, if $\mathbf{u}''$ remains bounded. We have the same problem as in the beginning, but within a reduced subspace. Either case i), ii), or iii) applies. If again case iii) applies, it will again lead to the same problem, but within a subspace reduced even further. After a finite number of these iterations the dimension of the respective subspace will be one. Then only case i) or ii) can apply and, hence, $||\mathbf{u}||$ will stay bounded.

It is possible to show that the $\mathcal{O}(t^{-1})$ convergence bound for $\tan \gamma_t$ is a tight bound. Due to the limited space we have to present the proof in a follow-up paper.

4 Computer Experiments

To illustrate these bounds with computer experiments, we measured the convergence of $\tan \gamma_t$ on two artificial data sets. Both data sets consisted of $N = 1000$ patterns $\mathbf{x}_\nu \in \mathbb{R}^D$, half of them belonging to class $+1$ and -1, respectively. The pattern space was two-dimensional ($D = 2$) for the first data set, and 100-dimensional ($D = 100$) for the second one.

Each data set was generated as follows: a random normal vector for the maximum margin hyperplane was chosen. On a hypersquare on this hyperplane with a sidelength of 2 the $N = 1000$ random input patterns were generated. Then half of them were shifted to one halfspace (class $+1$) by a random amount uniformly chosen from the interval $[0.1, 1]$, and the other half was shifted to the other halfspace (class -1) by a random amount uniformly chosen from $[-0.1, -1]$. To make sure that the chosen normal vector indeed defines the maximum margin hyperplane, for 30% of the patterns a margin of exactly 0.1 was chosen.

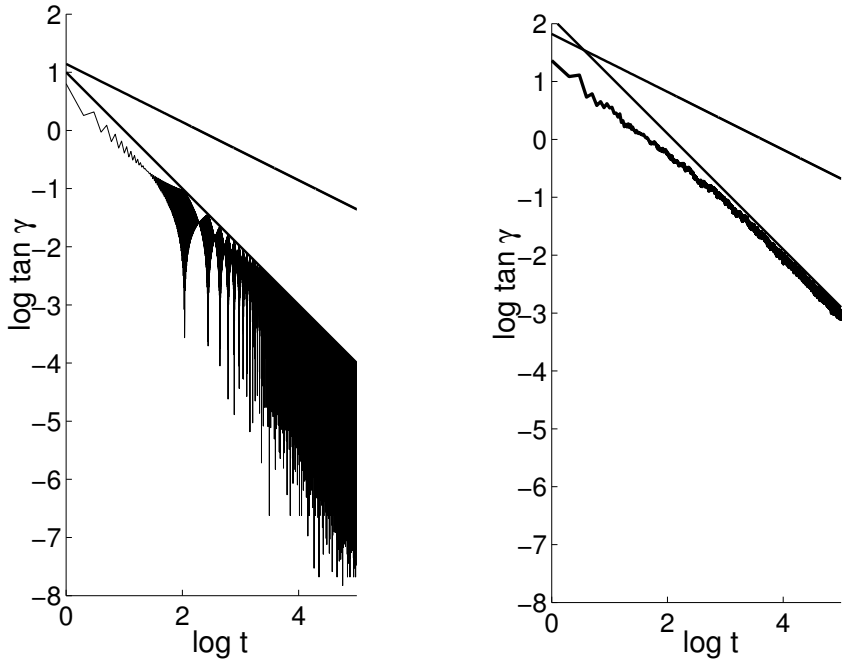

Fig. 1. Double-logarithmic plot of the angle γ_t between the maximum margin hyperplane and the hyperplane provided by MinOver against the number of learning steps t. After a finite number of learning steps the plot follows a line of slope -1, which demonstrates the $\mathcal{O}(t^{-1})$ convergence. For comparison the old $\mathcal{O}(t^{-1/2})$ convergence bound is shown. At the end of the learning procedure $\tan\gamma_t$ is about three orders of magnitude smaller than predicted by the old $\mathcal{O}(t^{-1/2})$-bound.

After each training step we calculated $\tan\gamma_t$ of the angle γ_t between the known maximum margin hyperplane and the hyperplane defined by $\mathbf{v}_t$. The result for both data sets is shown in Fig. 1. To visualize the convergence rate we chose a double logarithmic plot. As expected, in this double logarithmic plot convergence is bounded by a line with a slope of -1, which corresponds to the $\mathcal{O}(t^{-1})$ convergence we have proven. For comparison we also plotted the $\mathcal{O}(t^{-1/2})$-bound given by Equation (6), which cor-

responds to a line of slope $-1/2$. After 100.000 training steps $\tan\gamma_t$ is about three orders of magnitude smaller than predicted by the old $\mathcal{O}(t^{-1/2})$-bound.

5 Conclusions

The well-known MinOver algorithm as a simple and iterative procedure for obtaining maximum margin hyperplanes has been reformulated for the purpose of support vector classification with and without kernels. We have given an alternative proof for its well-known $\mathcal{O}(t^{-1/2})$ convergence. Based on this proof we have shown that the MinOver algorithm converges even at least as $\mathcal{O}(t^{-1})$ with increasing number of learning steps. We illustrated this result on two artificial data sets. With such a guarantee in convergence speed, with its simplicity, and with a computational effort which scales like $\mathcal{O}(N)$ with the number of training patterns the MinOver algorithm deserves a more widespread consideration in applications.

Acknowledgment
The author would like to thank Kai Labusch for his help preparing the manuscript.

References

1. C. Cortes and V. Vapnik. Support-vector-networks. *Machine Learning*, 20(3):273–297, 1995.
2. Y. Freund and R.E. Schapire. Large margin classification using the perceptron algorithm. In *Computational Learing Theory*, pages 209–217, 1998.
3. T.T. Friess, N. Cristianini, and C. Campbell. The kernel adatron algorithm: a fast and simple learning procedure for support vector machine. *Proc. 15th International Conference on Machine Learning*, 1998.
4. S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE-NN*, 11(1):124–136, January 2000.
5. W. Krauth and M. Mezard. Learning algorithms with optimal stability in neural networks. *J.Phys.A*, 20:745–752, 1987.
6. Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. *Int.Conf.on Artificial Neural Networks*, pages 53–60, 1995.
7. Y. Li and P.M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1-3):361–387, 2002.
8. H.D. Navone and T. Downs. Variations on a kernel-adatron theme. *VII Internacional Congress on Information Engineering, Buenos Aires*, 2001.
9. E. Osuna, R. Freund, and F. Girosi. Training support vector machines:an application to face detection. *CVPR'97*, pages 130–136, 1997.
10. J.C. Platt. *Advances in Kernel Methods - Support Vector Learning*, chapter Fast Training of Support Vector Machines using Sequential Minimal Optimization, pages 185–208. MIT Press, 1999.
11. B. Schölkopf. Support vector learning, 1997.
12. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.