Constant Rate Approximate Maximum Margin Algorithms

Petroula Tsampouka and John Shawe-Taylor

ECS, University of Southampton, UK

Abstract. We present a new class of Perceptron-like algorithms with margin in which the "effective" learning rate $\eta_{\rm eff}$, defined as the ratio of the learning rate to the length of the weight vector, remains constant. We prove that for $\eta_{\rm eff}$ sufficiently small the new algorithms converge in a finite number of steps and show that there exists a limit of the parameters involved in which convergence leads to classification with maximum margin. A soft margin extension for Perceptron-like large margin classifiers is also discussed.

1 Introduction

It is generally believed that the larger the margin of the solution hyperplane the greater is the generalisation ability of the learning machine [9,1]. The simplest online learning algorithm for binary linear classification, Rosenblatt's Perceptron [6], does not aim at any margin. The problem, instead, of finding the optimal margin hyperplane lies at the core of Support Vector Machines (SVMs) [9,1]. SVMs, however, require solving a quadratic programming problem which makes their efficient implementation difficult and often time consuming.

The difficulty in implementing SVMs has respurred a lot of interest in alternative large margin classifiers many of which are based on the Perceptron algorithm. The most well-known such variants are the standard Perceptron with margin [2,5] and the ALMA [4] algorithms. Here we address the maximum margin classification problem in the context of Perceptron-like algorithms which, however, differ from the above mentioned variants in that the ratio of the learning rate to the length of the weight vector remains constant. This new (class of) algorithm(s), called Constant Rate Approximate Maximum Margin Algorithm(s) (CRAMMA), emerges naturally if one attempts to classify Perceptron-like classifiers with margin in a few very broad categories according to the dependence on time of the misclassification condition or of the effect that an update has on the current weight vector. Under certain conditions CRAMMA converges in a finite number of steps to an approximation of the optimal solution which improves continually as its parameters follow a specific limiting process.

Maximal margin classifiers cannot be used directly in many real-world problems due to the inseparability of the data sets appearing in most applications. To cover the case of inseparable data we discuss a soft margin extension of the hard margin approach which is particularly suited for Perceptron-like large margin classifiers since it does not rely on convex optimisation theory.

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): ECML 2006, LNAI 4212, pp. 437–448, 2006.

A taxonomy of Perceptron-like large margin classifiers can be found in Sect. 2. CRAMMA is described in Sect. 3 together with an analysis regarding its convergence. Section 4 contains our discussion of the soft margin. Section 5 contains some experiments whereas Sect. 6 our conclusions.

2 Taxonomy of Perceptron-Like Large Margin Classifiers

In what follows we make the assumption that we are given a training set which, even if not initially linearly separable can, by an appropriate feature mapping into a space of a higher dimension [9,1], be classified into two categories by a linear classifier. This higher dimensional space in which the patterns are linearly separable will be the considered space. By adding one additional dimension and placing all patterns in the same position at a distance ρ in that dimension we construct an embedding of our data into the so-called augmented space [2]. The advantage of this embedding is that the linear hypothesis in the augmented space becomes homogeneous. Thus, only hyperplanes passing through the origin in the augmented space need to be considered even for tasks requiring bias. Throughout our discussion a reflection with respect to the origin in the augmented space of the negatively labelled patterns is assumed in order to allow for a uniform treatment of both categories of patterns. Also, we use the notation $R = \max\limits_k \|\boldsymbol{y}_k\|$, where $\boldsymbol{y}_k$ is the k^{th} augmented pattern. Obviously, $R \geq \rho$.

The relation characterising optimally correct classification of the training patterns $\boldsymbol{y}_k$ by a weight vector $\boldsymbol{u}$ of unit norm in the augmented space is

$$\boldsymbol{u} \cdot \boldsymbol{y}_k \geq \gamma_{\mathrm{d}} \equiv \max_{\boldsymbol{u}:\|\boldsymbol{u}\|=1} \min_i \{\boldsymbol{u} \cdot \boldsymbol{y}_i\} \qquad \forall k \ . \tag{1}$$

We call the quantity γ_{d} the maximum directional margin. It determines the maximum distance from the origin in the augmented space of the hyperplane normal to $\boldsymbol{u}$ placing all training patterns on the positive side and coincides with the maximum margin in the augmented space with respect to hyperplanes passing through the origin if no reflection is assumed. In the determination of this hyperplane only the direction of $\boldsymbol{u}$ is exploited with no reference to its projection onto the original space. Notice, however, that between γ_{d} and the maximum geometric margin γ in the original space the inequality

$$1 \leq \frac{\gamma}{\gamma_{\mathrm{d}}} \leq \frac{R}{\rho} \tag{2}$$

holds. In the limit $\rho \to \infty$, $R/\rho \to 1$ and from (2) $\gamma_{\mathrm{d}} \to \gamma$ [8]. Thus, with ρ increasing γ_{d} approaches γ.

We concentrate on algorithms that update the augmented weight vector $\boldsymbol{a}_t$ by adding a suitable positive amount in the direction of the misclassified (according to an appropriate condition) training pattern $\boldsymbol{y}_k$. The general form of such an update rule is

$$\boldsymbol{a}_{t+1} = (\boldsymbol{a}_t + \eta_t f_t \boldsymbol{y}_k)\, N_{t+1}^{-1} \ , \tag{3}$$

where η_t is the learning rate which could depend explicitly on the number t of updates that took place so far and f_t an implicit function of the current step (update) t, possibly involving the current weight vector $\boldsymbol{a}_t$ and/or the current misclassified pattern $\boldsymbol{y}_k$, which we require to be bounded by positive constants. We also allow for the possibility of normalising the newly produced weight vector $\boldsymbol{a}_{t+1}$ to a desirable length through a factor N_{t+1}. For the Perceptron $\eta_t = \eta$ is constant, $f_t = 1$ and $N_{t+1} = 1$. Each time the misclassification condition is satisfied by a training pattern the algorithm proceeds to the update of the weight vector. We adopt the convention of initialising t from 1.

A sufficiently general form of the misclassification condition is

$$\boldsymbol{u}_t \cdot \boldsymbol{y}_k \leq C(t) \ , \tag{4}$$

where $\boldsymbol{u}_t$ is the weight vector $\boldsymbol{a}_t$ normalised to unity and $C(t) > 0$ if we require that the algorithm achieves a positive margin. If $\boldsymbol{a}_1 = \boldsymbol{0}$ we treat the first pattern in the sequence as misclassified. We distinguish two cases depending on whether $C(t)$ is bounded from above by a strictly decreasing function of t which tends to zero or remains bounded from above and below by positive constants. In the first case the minimum directional margin required by such a condition becomes lower than any fixed value provided t is large enough. Algorithms with such a condition have the advantage of achieving some fraction of the unknown existing margin provided they converge. Examples of such algorithms are the well-known standard Perceptron algorithm with margin [2,5], in which the suppression of $C(t) = b/\|\boldsymbol{a}_t\|$ with t increasing is due to the growth of the length of the weight vector, and the ALMA$_2$ algorithm [4] with $C(t) = b/\|\boldsymbol{a}_t\| \sqrt{t}$. In the second case the condition amounts to requiring a directional margin, assumed to exist, which is not lowered arbitrarily with the number t of updates. In particular, if $C(t)$ is equal to a constant β [8] successful termination of the algorithm leads to a solution with margin larger than β. Obviously, convergence is not possible unless $\beta < \gamma_{\mathrm{d}}$. In this case an organised search through the range of possible β values is necessary.

An alternative classification of the algorithms with the perceptron-like update rule (3) is according to the dependence on t of the "effective" learning rate

$$\eta_{\mathrm{eff}\,t} \equiv \frac{\eta_t R}{\|\boldsymbol{a}_t\|} \tag{5}$$

which controls the impact that an update has on the current weight vector. More specifically, $\eta_{\mathrm{eff}\,t}$ determines the update of the direction $\boldsymbol{u}_t$

$$\boldsymbol{u}_{t+1} = \frac{\boldsymbol{u}_t + \eta_{\mathrm{eff}\,t} f_t \boldsymbol{y}_k / R}{\|\boldsymbol{u}_t + \eta_{\mathrm{eff}\,t} f_t \boldsymbol{y}_k / R\|} \ . \tag{6}$$

Again we distinguish two cases depending on whether $\eta_{\mathrm{eff}\,t}$ is bounded from above by a strictly decreasing function of t which tends to zero or remains bounded from above and below by positive constants. We do not consider the case that $\eta_{\mathrm{eff}\,t}$ increases indefinitely with t since we do not expect such algorithms to converge always in a finite number of steps. In the first category belong again the standard Perceptron algorithm, in which $\eta_t = \eta$ remains constant and $\|\boldsymbol{a}_t\|$ is bounded from below by a positive linear function of t, and ALMA$_2$ in which η_t decreases as $1/\sqrt{t}$. In the second category belong algorithms with the fixed directional margin condition $C(t) = \beta$, $\|\boldsymbol{a}_t\|$ normalised to a constant value and fixed learning rate [8].

In summary, the function $C(t)$ entering the misclassification condition and the effective learning rate ${\eta_{\text{eff}}}_t$ of a Perceptron-like algorithm could, roughly speaking, either be suppressed with time or remain practically constant. Thus, we are led to four broad categories of algorithms out of which the one with condition "relaxed" with time and a t-independent η_{eff} has not, to the best of our knowledge, been examined before. This is the subject of the present work.

3 The Constant Rate Approximate Maximum Margin Algorithm CRAMMA$^\epsilon$

We consider algorithms with constant effective learning rate ${\eta_{\text{eff}}}_t = \eta_{\text{eff}}$ in which the misclassification condition takes the form

$$\boldsymbol{u}_t \cdot \boldsymbol{y}_k \leq \frac{\beta}{t^\epsilon} \tag{7}$$

not involving $\|\boldsymbol{a}_t\|$. Here β and ϵ are positive constants. We assume that the initial value $\boldsymbol{u}_1$ of $\boldsymbol{u}_t$ is the unit vector in the direction of the first training pattern. Then,

$$\boldsymbol{u}_t \cdot \boldsymbol{u} > 0 \ . \tag{8}$$

This is true given that, on account of (6), $\boldsymbol{u}_t$ is a linear combination with positive coefficients of the training patterns $\boldsymbol{y}_k$ all of which have positive inner products with the optimal direction $\boldsymbol{u}$ because of (1). Additionally, we set $f_t = 1$. Since the misclassification condition (7) does not depend on $\|\boldsymbol{a}_t\|$ and given that the update (6) of $\boldsymbol{u}_t$ with $f_t = 1$ depends on $\|\boldsymbol{a}_t\|$ only through η_{eff} the algorithm does not depend separately on η_t and $\|\boldsymbol{a}_t\|$ but only on their ratio i.e. on η_{eff}.

```
Require: A linearly separable augmented training set with reflection
  assumed $S = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$
Define:
  For $k = 1, \ldots, m$
  $R = \max_k \|\boldsymbol{y}_k\|$ ,  $\bar{\boldsymbol{y}}_k = \boldsymbol{y}_k / R$
Fix: $\eta_{\text{eff}}$,  $\beta_1 \, (= \beta/R)$
Initialisation:
  $t = 1$,  $\boldsymbol{u}_1 = \bar{\boldsymbol{y}}_1 / \|\bar{\boldsymbol{y}}_1\|$
repeat until no update made within the for loop
  for $k = 1$ to $m$ do
    if $\boldsymbol{u}_t \cdot \bar{\boldsymbol{y}}_k \leq \beta_t$ then
      $\boldsymbol{u}_{t+1} = \frac{\boldsymbol{u}_t + \eta_{\text{eff}} \bar{\boldsymbol{y}}_k}{\|\boldsymbol{u}_t + \eta_{\text{eff}} \bar{\boldsymbol{y}}_k\|}$
      $t = t + 1$
      $\beta_t = \beta_1 / t^\epsilon$
```

Fig. 1. The Constant Rate Approximate Maximum Margin Algorithm CRAMMA$^\epsilon$

The above (family of) algorithm(s) parametrised in terms of the exponent ϵ and having a constant effective learning rate will be called the Constant Rate Approximate Maximum Margin Algorithm CRAMMA$^{\epsilon}$ and is summarised in Fig. 1. A justification of the qualification of the algorithm as an "Approximate Maximum Margin" one stems from the following theorem.

Theorem 1. *The* CRAMMA$^{\epsilon}$ *algorithm of Fig. 1 converges in a finite number of steps provided* $\eta_{\rm eff} < \frac{1}{2}\left(\sqrt{1+8\frac{\gamma_{\rm d}}{R}}-1\right)$. *Moreover, if* $\eta_{\rm eff}$ *is given a dependence on* β *through the relation* $\eta_{\rm eff} = \eta_0 \left(\frac{\beta}{R}\right)^{-\delta}$ *the directional margin* $\gamma'_{\rm d}$ *achieved by the algorithm tends in the limit* $\frac{\beta}{R} \to \infty$ *to the maximum one* $\gamma_{\rm d}$ *provided* $0 < \epsilon\delta < 1$.

Proof. Taking the inner product of (6) with the optimal direction $\boldsymbol{u}$ and expanding $\|\boldsymbol{u}_t + \eta_{\rm eff}\boldsymbol{y}_k/R\|^{-1}$ we have

$$\boldsymbol{u}_{t+1}\cdot\boldsymbol{u} = \left(\boldsymbol{u}_t\cdot\boldsymbol{u} + \eta_{\rm eff}\frac{\boldsymbol{y}_k\cdot\boldsymbol{u}}{R}\right)\left(1 + 2\eta_{\rm eff}\frac{\boldsymbol{y}_k\cdot\boldsymbol{u}_t}{R} + \eta_{\rm eff}^2\frac{\|\boldsymbol{y}_k\|^2}{R^2}\right)^{-\frac{1}{2}}$$

from where, by using the inequality $(1+x)^{-\frac{1}{2}} \geq 1 - \frac{x}{2}$, we get

$$\boldsymbol{u}_{t+1}\cdot\boldsymbol{u} \geq \left(\boldsymbol{u}_t\cdot\boldsymbol{u} + \eta_{\rm eff}\frac{\boldsymbol{y}_k\cdot\boldsymbol{u}}{R}\right)\left(1 - \eta_{\rm eff}\frac{\boldsymbol{y}_k\cdot\boldsymbol{u}_t}{R} - \eta_{\rm eff}^2\frac{\|\boldsymbol{y}_k\|^2}{2R^2}\right) .$$

Thus, we obtain for $\mathcal{D} \equiv \boldsymbol{u}_{t+1}\cdot\boldsymbol{u} - \boldsymbol{u}_t\cdot\boldsymbol{u}$

$$\frac{R}{\eta_{\rm eff}}\mathcal{D} \geq \boldsymbol{y}_k\cdot\boldsymbol{u} - (\boldsymbol{u}_t\cdot\boldsymbol{u})(\boldsymbol{y}_k\cdot\boldsymbol{u}_t) - \frac{\eta_{\rm eff}}{2R}\left(\|\boldsymbol{y}_k\|^2\,\boldsymbol{u}_t\cdot\boldsymbol{u} + 2(\boldsymbol{y}_k\cdot\boldsymbol{u})(\boldsymbol{y}_k\cdot\boldsymbol{u}_t)\right) - \frac{\eta_{\rm eff}^2}{2R^2}\|\boldsymbol{y}_k\|^2\,\boldsymbol{y}_k\cdot\boldsymbol{u} .$$

By employing (1), (7) and (8) we get a lower bound on $\mathcal{D}$

$$\frac{\mathcal{D}}{\eta_{\rm eff}} \geq \left(\frac{\gamma_{\rm d}}{R} - \frac{\eta_{\rm eff}}{2} - \frac{\eta_{\rm eff}^2}{2}\right) - (1+\eta_{\rm eff})\frac{\beta}{R}t^{-\epsilon} . \tag{9}$$

From the misclassification condition it is obvious that convergence of the algorithm is impossible unless $\beta/t^{\epsilon} < \gamma_{\rm d}$ i.e.

$$t > t_0 \equiv \left(\frac{\beta}{\gamma_{\rm d}}\right)^{\frac{1}{\epsilon}} . \tag{10}$$

A repeated application of (9) $(t-[t_0])$ times yields

$$\frac{\boldsymbol{u}_{t+1}\cdot\boldsymbol{u} - \boldsymbol{u}_{[t_0]+1}\cdot\boldsymbol{u}}{\eta_{\rm eff}} \geq \left(\frac{\gamma_{\rm d}}{R} - \frac{\eta_{\rm eff}}{2} - \frac{\eta_{\rm eff}^2}{2}\right)(t-[t_0]) - (1+\eta_{\rm eff})\frac{\beta}{R}\sum_{m=[t_0]+1}^{t} m^{-\epsilon}$$

with $[t_0]$ denoting the integer part of t_0. By employing the inequality

$$\sum_{m=[t_0]+1}^{t} m^{-\epsilon} \leq \int_{t_0}^{t} m^{-\epsilon} dm + t_0^{-\epsilon} = \frac{t^{1-\epsilon} - t_0^{1-\epsilon}}{1-\epsilon} + t_0^{-\epsilon}$$

and taking into account (8) we finally obtain

$$1 \geq \eta_{\text{eff}} \left(\frac{\gamma_{\text{d}}}{R}\right) \chi \left(t - t_0\right) - \eta_{\text{eff}} \left(1 + \eta_{\text{eff}}\right) \frac{\beta}{R} \frac{\left(t^{1-\epsilon} - t_0^{1-\epsilon}\right)}{1-\epsilon} - \omega \ . \qquad (11)$$

Here

$$\chi \equiv \left(1 - \frac{\eta_{\text{eff}}}{2} \left(1 + \eta_{\text{eff}}\right) \frac{R}{\gamma_{\text{d}}}\right) \quad \text{and} \quad \omega \equiv \eta_{\text{eff}} \left(1 + \eta_{\text{eff}}\right) \frac{\gamma_{\text{d}}}{R} \ .$$

Let us define the new variable $\tau \geq 0$ through the relation

$$t = t_0 \left(1 + \tau\right) = \left(\frac{\beta}{\gamma_{\text{d}}}\right)^{\frac{1}{\epsilon}} \left(1 + \tau\right) \ . \qquad (12)$$

In terms of τ (11) becomes

$$\frac{1}{\eta_{\text{eff}}} \left(\frac{\beta}{R}\right)^{-\frac{1}{\epsilon}} \left(\frac{\gamma_{\text{d}}}{R}\right)^{\left(\frac{1}{\epsilon}-1\right)} \left(1 + \omega\right) \geq \chi\tau - \left(1 + \eta_{\text{eff}}\right) \frac{\left(1+\tau\right)^{1-\epsilon} - 1}{1-\epsilon} \ . \qquad (13)$$

Let $g(\tau)$ be the r.h.s. of the above inequality. Since $\chi > 0$, given that $\eta_{\text{eff}} < \frac{1}{2}\left(\sqrt{1 + 8\frac{\gamma_{\text{d}}}{R}} - 1\right)$, it is not difficult to verify that $g(\tau)$ (with $\tau \geq 0$) is unbounded from above and has a single extremum, actually a minimum, at $\tau_{\text{min}} = \left(1 + \eta_{\text{eff}}\right)^{\frac{1}{\epsilon}} \chi^{-\frac{1}{\epsilon}} - 1 > 0$ with $g(\tau_{\text{min}}) < 0$. Moreover, the l.h.s of (13) is positive. Therefore, there is a single value τ_{b} of τ where (13) holds as an equality which provides an upper bound on τ

$$\tau \leq \tau_{\text{b}} \qquad (14)$$

satisfying $\tau_{\text{b}} > \tau_{\text{min}} > 0$. Combining (12) and (14) we obtain the bound on the number of updates

$$t \leq t_{\text{b}} \equiv \left(\frac{\beta}{\gamma_{\text{d}}}\right)^{\frac{1}{\epsilon}} \left(1 + \tau_{\text{b}}\right) \qquad (15)$$

proving that the algorithm converges in a finite number of steps. From (15) and taking into account the misclassification condition (7) we obtain a lower bound $\beta/t_{\text{b}}^{\epsilon}$ on the margin γ'_{d} achieved. Thus, the fraction f of γ_{d} that the algorithm achieves satisfies

$$f \equiv \frac{\gamma'_{\text{d}}}{\gamma_{\text{d}}} \geq f_{\text{b}} \equiv \frac{\beta/\gamma_{\text{d}}}{t_{\text{b}}^{\epsilon}} = \left(1 + \tau_{\text{b}}\right)^{-\epsilon} \ . \qquad (16)$$

Let us assume that $\frac{\beta}{R} \rightarrow \infty$ in which case from $\eta_{\text{eff}} = \eta_0 \left(\frac{\beta}{R}\right)^{-\delta}$ we have that $\eta_{\text{eff}} \rightarrow 0$. Consequently $\chi \rightarrow 1$, $\omega \rightarrow 0$ and (13) becomes

$$\frac{1}{\eta_0} \left(\frac{\beta}{R}\right)^{-\left(\frac{1}{\epsilon}-\delta\right)} \left(\frac{\gamma_{\text{d}}}{R}\right)^{\left(\frac{1}{\epsilon}-1\right)} \geq \tau - \frac{\left(1+\tau\right)^{1-\epsilon} - 1}{1-\epsilon} \ . \qquad (17)$$

Provided $\epsilon\delta < 1$ the l.h.s. of the above inequality vanishes in the limit $\frac{\beta}{R} \to \infty$. Then, since τ_{min} vanishes as well, the r.h.s. of the inequality becomes a strictly increasing function of τ and (17) obviously holds as an equality only for $\tau = 0$. Therefore,

$$\tau_{\mathrm{b}} \to \tau_{\mathrm{min}} \to 0 \quad \text{as} \quad \frac{\beta}{\mathrm{R}} \to \infty \ . \tag{18}$$

Combining (16) with (18) and taking into account that $f \leq 1$ by definition we conclude that

$$f \to 1 \quad \text{as} \quad \frac{\beta}{\mathrm{R}} \to \infty \ .$$

□

Remark 1. In the case $\epsilon = \frac{1}{2}$ by solving the quadratic equation derived from (13) we obtain explicitly an upper bound t_{b} on the number of updates and a lower bound f_{b} on the fraction f of the margin that the algorithm achieves. They are the ones of (15) and (16), respectively with

$$\tau_{\mathrm{b}} = \left\{ \frac{1+\eta_{\mathrm{eff}}}{\chi} + \sqrt{\left(\frac{1+\eta_{\mathrm{eff}}}{\chi} - 1\right)^2 + \eta_{\mathrm{eff}}^{-1} \left(\frac{\beta}{R}\right)^{-2} \frac{\gamma_{\mathrm{d}}\,(1+\omega)}{\chi R}} \right\}^2 - 1 \ . \tag{19}$$

As $\frac{\beta}{R} \to \infty$, $\eta_{\mathrm{eff}} = \eta_0 \left(\frac{\beta}{R}\right)^{-\delta} \to 0$, $\chi \to 1$ and $\omega \to 0$. Then, $\tau_{\mathrm{b}} \to 0$ given that $\eta_{\mathrm{eff}}^{-1} \left(\frac{\beta}{R}\right)^{-2} = \eta_0^{-1} \left(\frac{\beta}{R}\right)^{\delta-2} \to 0$ if $0 < \delta < 2$. This demonstrates explicitly the statement of Theorem 1. Explicit bounds t_{b} and f_{b} are also obtainable for $\epsilon = 2$.

4 Soft Margin Extension

Maximal margin classifiers, representing the hard margin approach, cannot be employed in many real-world problems since there is in general no linear separation in the feature space and the use of powerful kernels might lead to overfitting. The most widely accepted solution to this problem is the adoption of the so-called soft margin approach. In the SVM formulation [9,1] the soft margin approach is implemented through the introduction of "slack" variables in order to allow for violations of the margin condition by some training patterns.

Freund and Shapire [3] have shown how a function of the margin distribution different from the minimum margin one can be used to bound the number of mistakes of an online Perceptron algorithm. Their technique makes the data set linearly separable by extending the instance space by as many dimensions as the number of instances and placing each instance at a distance $|\varDelta|$ from the origin in the corresponding dimension. An interesting result in this connection is the observation that the hard margin optimisation task in the extended space is equivalent to the soft margin optimisation in the original instance space if the 2-norm of the slack variables is employed [7].

In the sequel, following the approach of [3], we show how one moves in the direction of minimising an objective function $\mathcal{J}$ involving the new margin distribution by making use of Perceptron-like algorithms which, however, are seeking a hard margin in the extended space. This may not be surprising in the light of the result just mentioned regarding the equivalence between the hard margin optimisation in the extended space and the soft margin one in the original space. Nevertheless, we hope that our analysis, which does not rely on convex optimisation theory, will contribute to a better understanding of what an algorithm running in the extended space actually achieves with respect to the original space. Although our instance space prior to its extension is the augmented one in the present section the instances $\boldsymbol{y}_k$ are explicitly accompanied by their labels l_k since we found convenient not to assume a reflection with respect to the origin.

Theorem 2. *Let $((\boldsymbol{y}_1, l_1), \ldots, (\boldsymbol{y}_m, l_m))$ be a sequence of m labelled instances, $\boldsymbol{u}$ a unit vector and $\gamma > 0$. Define $d_i = \max\{0, \gamma - l_i \boldsymbol{u} \cdot \boldsymbol{y}_i\}$ and set $D = \sqrt{\sum_i d_i^2}$. In addition define an extended instance space $\boldsymbol{y}_i^{\mathrm{ext}} = (\boldsymbol{y}_i, \Delta\delta_{1i}, \ldots, \Delta\delta_{mi})$ parametrised by Δ, where δ_{ij} is Kronecker's δ.*

1. *Let $\Gamma_{\Delta \mathrm{opt}}$ be the maximum margin in the extended space with respect to hyperplanes passing through the origin. Then, for any $\boldsymbol{u}$ and γ,*
$$\Gamma_{\Delta \mathrm{opt}}^{-2} \leq \mathcal{J}(\boldsymbol{u}, \gamma, \Delta) \equiv \frac{1}{\gamma^2} + \frac{1}{\Delta^2} \left(\frac{D}{\gamma}\right)^2 \quad . \tag{20}$$
2. *Assume that a zero-threshold algorithm converges in the extended space to a solution vector $\boldsymbol{a}^{\mathrm{ext}}$ which describes a hyperplane passing through the origin with margin Γ_Δ. Let $\boldsymbol{u} = \boldsymbol{a} / \|\boldsymbol{a}\|$ and $\gamma = \Gamma_\Delta \|\boldsymbol{a}^{\mathrm{ext}}\| / \|\boldsymbol{a}\|$, where $\boldsymbol{a}$ is the projection of $\boldsymbol{a}^{\mathrm{ext}}$ onto the original instance space. Then, employing such a $\boldsymbol{u}$ and γ provided by the algorithm, we have*
$$\Gamma_{\Delta \mathrm{opt}}^{-2} \leq \mathcal{J}(\boldsymbol{u}, \gamma, \Delta) \leq \Gamma_\Delta^{-2} \quad . \tag{21}$$

Proof. 1. Notice that $\mathcal{J}(\boldsymbol{u}, \gamma, \Delta) = Z^2/\gamma^2$ with $Z \equiv \sqrt{1 + D^2/\Delta^2}$. Then, (20) is equivalent to $\gamma / Z \leq \Gamma_{\Delta \mathrm{opt}}$ which is proved in [3].

2. Let us assume that a zero-threshold algorithm converges in the extended space to a weight vector $\boldsymbol{a}^{\mathrm{ext}}$ in the direction of the unit vector $\boldsymbol{u}^{\mathrm{ext}}$
$$\boldsymbol{u}^{\mathrm{ext}} = \frac{\boldsymbol{a}^{\mathrm{ext}}}{\|\boldsymbol{a}^{\mathrm{ext}}\|} = \frac{1}{Z'} \left(\boldsymbol{u}, l_1 \frac{d_1'}{\Delta}, \ldots, l_i \frac{d_i'}{\Delta}, \ldots, l_m \frac{d_m'}{\Delta}\right) \quad , \tag{22}$$
where $Z' = \|\boldsymbol{a}^{\mathrm{ext}}\| / \|\boldsymbol{a}\| = \sqrt{1 + D'^2/\Delta^2}$ with $D' = \sqrt{\sum_i d_i'^2}$. Here $\boldsymbol{a}$ is the projection of $\boldsymbol{a}^{\mathrm{ext}}$ onto the original instance space and $\boldsymbol{u}$ is the unit vector in the direction of $\boldsymbol{a}$. Let Γ_Δ be the margin achieved by $\boldsymbol{u}^{\mathrm{ext}}$ and $\gamma \equiv \Gamma_\Delta Z'$. We have
$$l_i \boldsymbol{u}^{\mathrm{ext}} \cdot \boldsymbol{y}_i^{\mathrm{ext}} = \frac{1}{Z'} \left(l_i \boldsymbol{u} \cdot \boldsymbol{y}_i + d_i'\right) \geq \Gamma_\Delta = \frac{\gamma}{Z'}$$
from where
$$d_i' \geq \gamma - l_i \boldsymbol{u} \cdot \boldsymbol{y}_i \quad . \tag{23}$$

The above inequality, taking into account the definition of d_i, leads to

$$|d'_i| \geq d_i \geq 0 \tag{24}$$

and consequently to $Z' \geq Z$. Therefore, taking into consideration the definition of γ, we obtain

$$\frac{\gamma}{Z} \geq \frac{\gamma}{Z'} = \mathit{\Gamma}_{\mathit{\Delta}} \tag{25}$$

which leads to

$$\mathcal{J}(\boldsymbol{u}, \gamma, \mathit{\Delta}) \leq \mathit{\Gamma}_{\mathit{\Delta}}^{-2} \tag{26}$$

given that $Z^2/\gamma^2 = \mathcal{J}(\boldsymbol{u}, \gamma, \mathit{\Delta})$. The proof is completed by combining (20) and (26). □

Remark 2. Let the zero-threshold algorithm be a Perceptron-like algorithm with initial weight vector $\boldsymbol{a}_1^{\text{ext}} = \sum_k \alpha_k l_k \boldsymbol{y}_k^{\text{ext}}$ and $\alpha_k \geq 0$. From the initialisation, the update rule (3) and the definition of the extended space follows that $d'_i \geq 0$.

Remark 3. If the algorithm converges to the maximal margin hyperplane passing through the origin in the extended space then $\mathit{\Gamma}_{\mathit{\Delta}} = \mathit{\Gamma}_{\mathit{\Delta}\text{opt}}$. Moreover, (20) is equivalent to $\gamma/Z \leq \mathit{\Gamma}_{\mathit{\Delta}\text{opt}}$ which combined with (25) and given that $\mathit{\Gamma}_{\mathit{\Delta}} = \mathit{\Gamma}_{\mathit{\Delta}\text{opt}}$ gives $Z' = Z$ or $D' = D$ from where $|d'_i| = d_i$ follows taking into account (24). In addition, $d'_i \geq 0$. Indeed, if $d'_i < 0$ then $d_i = 0$ because of (23) and the definition of d_i. But in this case $d'_i = d_i = 0$ contradicting our assumption that $d'_i < 0$. Thus, for the optimal extended space solution $d'_i = d_i$.

Remark 4. Setting $\boldsymbol{w} = \boldsymbol{u}/\gamma$, $\xi_i = |d'_i|/\gamma \geq d_i/\gamma = \max\{0, 1 - l_i \boldsymbol{w} \cdot \boldsymbol{y}_i\}$ and $C = \mathit{\Delta}^{-2}$ yields

$$\frac{1}{\gamma^2} + \frac{1}{\mathit{\Delta}^2}\left(\frac{D'}{\gamma}\right)^2 = \|\boldsymbol{w}\|^2 + C\sum_i \xi_i^2 \ .$$

We recognise the objective function of the primal form of the 2-norm soft margin optimisation problem in which the role of the constraints is played by (24) but the bias term is missing since it is, at least partially, incorporated in the augmented weight vector $\boldsymbol{w}$. If the optimal solution is found $d'_i = d_i$ and the "slack" variables ξ_i become $\xi_i = \max\{0, 1 - l_i \boldsymbol{w} \cdot \boldsymbol{y}_i\}$.

Theorem 2 shows that minimisation of the objective function $\mathcal{J}$ is equivalent to finding the maximum margin in the extended space. The $\boldsymbol{u}$ and γ for which the minimum $\mathcal{J}_{\min}$ is attained determine uniquely both the maximum margin $\mathit{\Gamma}_{\mathit{\Delta}\text{opt}} = \mathcal{J}_{\min}^{-\frac{1}{2}}$ and the direction $\boldsymbol{u}_{\text{opt}}^{\text{ext}}$ of the optimal weight vector in the extended space which is given by (22) with $d'_i = d_i$. Moreover, (21) provides an estimate of the deviation of the value of $\mathcal{J}$ achieved as a result of an incomplete optimisation from $\mathcal{J}_{\min}$ if we have an estimate of the difference between $\mathit{\Gamma}_{\mathit{\Delta}}$ and $\mathit{\Gamma}_{\mathit{\Delta}\text{opt}}$.

We conclude this section with a well-known lower bound on the margin $\mathit{\Gamma}_{\mathit{\Delta}\text{opt}}$. Let $\boldsymbol{u}^{\text{ext}} = \text{sgn}(\mathit{\Delta}) m^{-\frac{1}{2}} (\boldsymbol{0}, l_1, l_2, \ldots, l_m)$ be an extended unit vector with vanishing projection onto the original instance space. It is straightforward to see that $l_i \boldsymbol{u}^{\text{ext}} \cdot \boldsymbol{y}_i^{\text{ext}} = |\mathit{\Delta}|/\sqrt{m}$ meaning that $\boldsymbol{u}^{\text{ext}}$ achieves a margin of $|\mathit{\Delta}|/\sqrt{m}$. Thus,

$$\mathit{\Gamma}_{\mathit{\Delta}\text{opt}} \geq |\mathit{\Delta}|/\sqrt{m} \ . \tag{27}$$

Table 1. Results for the sonar data set. The directional margin γ'_{d} achieved and the number of updates (upds) are given for the Perceptron, ALMA$_2$ and CRAMMA$^{\frac{1}{2}}$. For CRAMMA$^{\frac{1}{2}}$ we choose $\eta_{\mathrm{eff}} = 0.001 \left(\frac{\beta}{R}\right)^{-1}$.

Perceptron			ALMA$_2$			CRAMMA$^{\frac{1}{2}}$		
$\frac{b}{\eta R^2}$	$10^3\gamma'_{\mathrm{d}}$	upds	α	$10^3\gamma'_{\mathrm{d}}$	upds	$\frac{\beta}{R}$	$10^3\gamma'_{\mathrm{d}}$	upds
1	5.78	247,140	0.75	5.65	415,119	0.73	5.73	248,267
3.1	7.14	660,698	0.55	7.08	1,584,785	1.52	7.12	669,170
5.4	7.45	1,117,124	0.45	7.45	3,133,968	2	7.46	1,047,757
20	7.80	3,977,612	0.35	7.76	6,657,109	3	7.82	2,143,989
90	7.91	17,647,271	0.3	7.91	10,170,590	3.45	7.92	2,762,005
200	7.92	39,131,402	0.2	8.11	28,339,340	5	8.11	5,531,113
500	7.93	97,717,549	0.1	8.27	137,693,242	10	8.28	21,220,354
1000	7.93	195,358,932	0.03	8.37	1,735,836,937	30.1	8.37	188,073,965

Table 2. Results for the sonar data set with CRAMMA^2 and $\eta_{\mathrm{eff}} = 0.4(\frac{\beta}{R})^{-0.3}$

$\frac{\beta}{R}$	10^6	10^7	10^8	10^9	10^{10}	10^{11}	10^{12}	10^{13}
$10^3\gamma'_{\mathrm{d}}$	1.03	3.66	5.52	6.69	7.37	7.80	8.10	8.27
upds	70,798	106,507	264,396	756,035	2,275,334	6,994,002	21,690,267	67,893,557

5 Experiments

In this section we present the results of experiments performed in order to verify our theoretical analysis and evaluate the performance of the CRAMMA$^{\epsilon}$ algorithm in comparison with the other two well-known similar in spirit algorithms, namely the Perceptron with margin and ALMA$_2$ [1].

First we analyse the training data set of the sonar classification problem as originally selected for the aspect-angle dependent experiment. It consists of 104 patterns each with 60 attributes obtainable from the UCI repository. Here the data are embedded in the augmented space at a distance $\rho = 1$ from the origin in the additional dimension leading to $R \simeq 3.8121$ and $\gamma_{\mathrm{d}} \simeq 0.00841$. The results of our comparative study of the Perceptron, ALMA$_2$ and CRAMMA$^{\frac{1}{2}}$ ($\epsilon = \frac{1}{2}$) algorithms are presented in Table 1. We observe that for values of the margin γ'_{d} near the maximum one CRAMMA$^{\frac{1}{2}}$ is certainly the fastest by far. Moreover, the data suggest that the Perceptron is not able to approach the maximum margin arbitrarily close. We also present in Table 2 results obtained by the CRAMMA^2 ($\epsilon = 2$) algorithm.

We additionally analyse a linearly separable data set, which we call WBC$_{-11}$, consisting of 672 patterns each with 9 attributes. It is constructed from the

[1] The parameters for ALMA$_2$ were chosen to correspond to the ones of the theorem in [4] if the data are normalised such that the longest pattern has unit length. The parameter $\alpha \in (0, 1]$ controls the accuracy to which the maximum margin is approximated.

Table 3. Results for the WBC$_{-11}$ data set for the algorithms Perceptron, ALMA$_2$ and CRAMMA$^{\frac{1}{2}}$. For CRAMMA$^{\frac{1}{2}}$ the choice $\eta_{\rm eff} = 0.0001 \left(\frac{\beta}{R}\right)^{-1}$ is made.

Perceptron			ALMA$_2$			CRAMMA$^{\frac{1}{2}}$		
$\frac{b}{\eta R^2}$	$10^2\gamma'_{\rm d}$	upds	α	$10^2\gamma'_{\rm d}$	upds	$\frac{\beta}{R}$	$10^2\gamma'_{\rm d}$	upds
0.52	1.784	1,718,705	0.8	1.783	2,704,553	0.22	1.794	259,036
0.9	2.008	2,720,447	0.7	2.008	6,254,523	0.32	2.019	431,543
1.4	2.141	3,976,477	0.6	2.141	13,320,425	0.42	2.143	660,486
2.1	2.228	5,734,457	0.5	2.228	27,666,246	0.49	2.238	824,120
4	2.317	10,508,566	0.35	2.315	88,363,792	0.8	2.318	2,044,555

Wisconsin Breast Cancer (WBC) data set obtainable from the UCI repository by first omitting the 16 patterns with missing features and subsequently removing from the data set containing the remaining 683 patterns the 11 patterns having the positions 2, 4, 191, 217, 227, 245, 252, 286, 307, 420 and 475. The value $\rho = 30$ is chosen for the parameter ρ of the augmented space leading to $R = \sqrt{1716}$ and $\gamma_{\rm d} \simeq 0.0243$. In Table 3 we present the results of a comparative study of the Perceptron, ALMA$_2$ and CRAMMA$^{\frac{1}{2}}$ algorithms. The superiority of the performance of the CRAMMA$^{\frac{1}{2}}$ on this data set is apparent.

Table 4. Results for the (extended) WBC data set (with $\varDelta = 1$). The relative deviations $\frac{\delta D}{D}$ and $\frac{\delta \varGamma}{\varGamma}$, the margin $\varGamma_\varDelta$ and the number of updates (upds) are given for the Perceptron, ALMA$_2$ and CRAMMA$^{\frac{1}{2}}$. For CRAMMA$^{\frac{1}{2}}$ we choose $\eta_{\rm eff} = \frac{1.7}{R\sqrt{683}} \left(\frac{\beta}{R}\right)^{-1}$.

Perceptron					ALMA$_2$					CRAMMA$^{\frac{1}{2}}$				
$\frac{b}{\eta R^2}$	$10\frac{\delta D}{D}$	$10\frac{\delta \varGamma}{\varGamma}$	$10\varGamma_\varDelta$	upds	α	$10\frac{\delta D}{D}$	$10\frac{\delta \varGamma}{\varGamma}$	$10\varGamma_\varDelta$	upds	$\frac{\beta}{R}$	$10\frac{\delta D}{D}$	$10\frac{\delta \varGamma}{\varGamma}$	$10\varGamma_\varDelta$	upds
1	2.22	2.14	1.0244	67,913	0.75	2.18	2.19	1.0185	79,061	0.95	2.36	2.22	1.0143	80,671
2.4	1.13	1.11	1.1585	144,938	0.6	1.13	1.16	1.1524	248,461	1.64	1.10	1.12	1.1568	185,687
10	0.39	0.38	1.2542	560,591	0.35	0.42	0.42	1.2481	1,625,682	3.1	0.36	0.37	1.2551	560,229
45	0.19	0.19	1.2789	2,474,607	0.2	0.19	0.19	1.2784	7,184,572	5	0.18	0.19	1.2791	1,401,588
700	0.15	0.15	1.2837	38,336,601	0.1	0.08	0.08	1.2933	35,542,412	11.5	0.08	0.08	1.2934	7,252,904

Finally, we test our algorithms on the extended instance space constructed from the linearly inseparable WBC data set comprising 683 patterns each with 9 attributes after ignoring the 16 patterns with missing attributes. We embed the data in the augmented space at a distance $\rho = 10$ from the origin in the additional dimension and we subsequently construct the extended instance space parametrised by $\varDelta = 1$. In order to determine the value of $\eta_{\rm eff}$ we take advantage of the lower bound (27) on the margin $\varGamma_{\varDelta{\rm opt}}$ of the extended space and set $\eta_{\rm eff} = 1.7|\varDelta|/R\sqrt{m}$ for $\beta = R$ which satisfies the constraint of Theorem 1. Here $m = 683$ and $R = \sqrt{917}$. We also take advantage of another property of the extended space in order to attempt an assessment of the relative deviation of the margin $\varGamma_\varDelta$ found from the (unknown) maximum $\varGamma_{\varDelta{\rm opt}}$: the quantities D and D' defined in Sect. 4 for which $D' \geq D$ holds become equal, according to Remark 3, if the optimal extended solution vector is found. Thus, we may take the relative deviation $\delta D/D \equiv (D' - D)/D$ as a measure of the departure from optimality. In Table 4 we give the results of our comparative study of the Perceptron, ALMA$_2$ and CRAMMA$^{\frac{1}{2}}$ algorithms. We observe that once again CRAMMA$^{\frac{1}{2}}$ is the fastest near the maximum margin $\mathit{\Gamma}_{\Delta\mathrm{opt}} \simeq 0.13033$ where the objective function $\mathcal{J}$ is minimised. Moreover, $\delta D/D$ proves a surprisingly accurate measure of the relative deviation $\delta\mathit{\Gamma}/\mathit{\Gamma} \equiv (\mathit{\Gamma}_{\Delta\mathrm{opt}} - \mathit{\Gamma}_{\Delta})/\mathit{\Gamma}_{\Delta\mathrm{opt}}$ of $\mathit{\Gamma}_{\Delta}$ from $\mathit{\Gamma}_{\Delta\mathrm{opt}}$.

6 Conclusions

We presented a new class of Perceptron-like large margin classifiers characterised by a constant effective learning rate. Our theoretical approach proved sufficiently powerful in establishing asymptotic convergence to the optimal hyperplane for a whole class of such algorithms in which the misclassification condition is relaxed with an arbitrary power of the number of updates. Thus, it becomes obvious that the ability to approach the maximum margin arbitrarily close is not a property of some very special algorithmic constructions but, instead, characterises larger families of algorithms under rather mild assumptions. We additionally discussed a soft margin extension for Perceptron-like large margin classifiers. Finally, we provided experimental evidence in support of our theoretical analysis.

References

1. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines (2000) Cambridge, UK: Cambridge University Press
2. Duda, R.O., Hart, P.E.: Pattern Classsification and Scene Analysis (1973) Wiley
3. Freund, Y., Shapire, R. E.: Large margin classification using the perceptron algorithm. Machine Learning **37**(3) (1999) 277–296
4. Gentile C.: A new approximate maximal margin classification algorithm. Journal of Machine Learning Research **2** (2001) 213–242
5. Krauth, W., Mézard, M.: Learning algorithms with optimal stability in neural networks. Journal of Physics **A 20** (1987) L745–L752
6. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review **65**(6) (1958) 386–408
7. Shawe-Taylor, J., Cristianini, N.: Further results on the margin distribution. In COLT'99 (1999) 278–285
8. Tsampouka, P., Shawe-Taylor, J.: Analysis of generic perceptron-like large margin classifiers. ECML 2005, LNAI **3720** (2005) 750–758, Springer-Verlag
9. Vapnik, V. N.: The Nature of Statistical Learning Theory (1995) Springer Verlag