The Kernel-Adatron Algorithm: a Fast and Simple Learning Procedure for Support Vector Machines

Thilo-Thomas Frieß
Dept. of Automatic Control
and Systems Engineering
University of Sheffield, UK
friess@acse.shef.ac.uk

Nello Cristianini
Dept. of Engineering Mathematics
University of Bristol, UK
nello.cristianini@bristol.ac.uk

Colin Campbell
Dept. of Engineering Mathematics
University of Bristol, UK
c.campbell@bristol.ac.uk

Abstract

Support Vector Machines work by mapping training data for classification tasks into a high dimensional feature space. In the feature space they then find a maximal margin hyperplane which separates the data. This hyperplane is usually found using a quadratic programming routine which is computationally intensive, and is non trivial to implement. In this paper we propose an adaptation of the Adatron algorithm for classification with kernels in high dimensional spaces. The algorithm is simple and can find a solution very rapidly with an exponentially fast rate of convergence (in the number of iterations) towards the optimal solution. Experimental results with real and artificial datasets are provided.

Keywords: Support Vector Machine, Large Margin Classifier, Adatron, Statistical Mechanics

1 INTRODUCTION

Support Vector (SV) machines are an algorithm introduced by Vapnik and co-workers [5, 4] theoretically motivated by VC theory. They are based on the following idea: input points are mapped to a high dimensional feature space, where a separating hyperplane can be found. The algorithm is chosen in such a way to maximize the distance from the closest patterns, a quantity which is called the margin.

This is achieved by reducing the problem to a quadratic programming problem, which is then usually solved with optimization routines from numerical libraries. This step is computational intensive, can be subject to stability problems and it is non trivial to implement.

SV machines have a proven impressive performance on a number of real world problems such as optical character recognition and face detection [5, 6, 19, 17]. However, their uptake has been limited in practice because of the mentioned problems with the current training algorithms.

An analogous problem has been studied in the Statistical Mechanics literature, which has produced a number of perceptron learning procedures aimed at finding maximal margin hyperplanes in the input space [11, 15, 13]. For some of them also theoretical guarantees are provided, as in the case of Adatron [2], where not only the convergence toward the optimal solution has been proved, but also an exponential rate of convergence in the number of iterations.

We propose a "hybrid" algorithm, the Kernel-Adatron (KA), which combines the implementational simplicity of Adatron with the capability of working in nonlinear feature spaces as SV machines do. By introducing Kernels into the algorithm it is possible to maximize the margin in the feature space, which is equivalent to nonlinear decision boundaries in the input space. The algorithm comes with all the theoretical guarantees given by VC theory for large margin classifiers, as well as the convergence properties studied in the Statistical Mechanics literature.

The result is a fast, robust and extremely simple procedure which implements the same ideas and principles as SV machines at much smaller cost. Experimental results are provided which show that indeed the predictive power of our algorithm is equivalent to that of a SV machine. Furthermore, we show that the running time can be orders of magnitude faster.

2 SUPPORT VECTOR MACHINES

Support Vector (SV) machines implement complex (nonlinear) decision rules in terms of hyperplanes in high dimensional spaces and were originally introduced by Vapnik and co-workers [24, 5, 10].

The decision function realized by SV machines can conceptually be described in two steps: first the training points are mapped by a nonlinear function ϕ to a high-dimensional space where they are linearly separable. Then a separating hyperplane is found which maximizes its distance from the training set, called the margin.

Theoretical results exist from VC theory [24, 21], which guarantee that such solution has high predictive power, in the sense that it minimizes an upper bound on the test error (a complete survey covering the generalization power of SV machines can be found in [3]).

Let $S = \{(x_1, y_1), (x_2, y_2), ..., (x_p, y_p)\}$ be a sample of points $x_i \in \mathcal{X}$ labelled by $y_i \in \{-1, +1\}$.

Consider a hyperplane defined by (w, θ), where w is a weight vector and θ a threshold value. Let $S = (X, Y)$ a labeled sample of inputs from $\mathcal{X}$ that has empty intersection with the hyperplane, so that

$$\gamma = \min_{x \in X} |\langle x, w \rangle + \theta| > 0$$

We call this distance the *margin* of the hyperplane w with respect to the sample S.

We also say that the hyperplane is in canonical form with respect to the sample if

$$\min_{x \in X} |\langle x, w \rangle + \theta| = 1$$

It is possible to prove that for canonical hyperplanes

$$\gamma = 1/||w||_2$$

The following theorem holds:

Theorem: [**21**] Suppose inputs are drawn independently according to a distribution whose support is contained in a ball in $\Re^n$ centered at the origin, of radius R. If we succeed in correctly classifying m such inputs by a canonical hyperplane with $||w|| = 1/\gamma$ and $|\theta| \leq R$, then with confidence $1 - \delta$ the generalization error will be bounded from above by

$$\epsilon(m, \gamma) = \frac{1}{m} \left(k \log \left(\frac{8em}{k} \right) \log(32m) + \log \left(\frac{8m}{\delta} \right) \right)$$

where $k = \lfloor 577R^2/\gamma^2 \rfloor$.

The quantity which upper bounds the generalization error does not depend on the dimension of the input space, and this is the theoretical reason why SV machines can use high dimensional spaces without overfitting.

Two main ideas (data-dependent representation and kernels) make it possible to efficiently deal with very high dimensional feature spaces.

The first is based on the identity:

$$\sum_{i=1}^{N} w_i \phi_i(x) + \theta = \sum_{k=1}^{p} \alpha_k \phi(x_k) \phi(x) + \theta$$

which provides an alternative, data-dependent, representation of the hypothesis itself, and the other is the use of kernels:

$$K(x', x) = \sum_i \phi_i(x') \phi_i(x)$$

which are equivalent to computing the dot product of the images of two vectors in the feature space [1], provided some (nontrivial) conditions are satisfied.

A common choice are Radial Basis Functions (RBF) such as gaussians,

$$K(x, x') = e^{-||x - x'||^2/2\sigma^2}$$

or polynomial kernels

$$K(x, x') = (\langle x, x' \rangle + 1)^d$$

which satisfy such conditions.

The use of the kernels instead of the dot product, in the data-dependent representation of the decision function, automatically provides a way to represent hyperplanes in a feature space rather than in the input space, as first described in [1].

The second conceptual step, aimed at finding the large margin hyperplane, is performed in SV machines by trasforming the problem into a Quadratic Programming one, subject to linear constraints.

Kuhn-Tucker theory [24] provides the framework under which the problem can be solved and gives the properties of the solution.

In the data-dependent representation, the lagrangian

$$L = \sum_{i=1}^{p} \alpha_i - 1/2 \sum_{i,j=1}^{p} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

has to be maximized with respect to the α_i, subject to the constraints

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{p} \alpha_i y_i = 0$$

This formulation has a number of interesting properties, characterizing the behaviour of the optimal hyperplane.

There is a Lagrange multiplier α_i for each training point. Only the points which lie closest to the hyperplane, (on parallel hyperplanes at distance γ from the optimal one) have $\alpha_i > 0$ and are called *support vectors*. All the others have $\alpha_i = 0$.

This means that in the representation of the solution, only the points which are closest to the hyperplane contribute: in fact they represent the hypothesis itself, (and their number can also be used to give an independent bound on its reliability [3]).

The resulting decision function can be written as:

$$f(x) = \text{sign}\left(\sum_{i \in \mathtt{SV}} y_i \alpha_i^o K(x, x_i) - \theta\right)$$

where α_i^o is the solution of the constrained maximization problem and `SV` represents the (indexes of) support vectors.

Such a scheme has proved to be very resistent to overfitting in many classification problems [19, 6, 24]. However this scheme is non trivial to implement, and computationally expensive. Furthermore, in some conditions, it can suffer from numerical conditioning problems.

It is interesting to note that other algorithms which were developed with different motivations have been shown to use a similar technique, equivalent to mapping points to a high dimensional feature space and separating them with a large margin hyperplane. This is the case for Adaboost [18], and for Bayesian Classifiers [7] where the margin distribution over all the training set is used as an estimator, rather than the margin.

This is justified by a theorem from Schapire *et al.* [18] proving that the fraction of training points which are classified with large margin controls the predictive power, and that valid generalization can be guaranteed even when few points lie near the boundary and hence the margin of the sample is small.

3 THE KERNEL-ADATRON (KA) ALGORITHM

In the Statistical Mechanics approach to learning [25], a very similar problem has been studied, with different motivations. The "perceptron with optimal stability" has been the object of extensive theoretical and experimental work, [15, 11, 2], and a number of simple iterative procedures have been proposed, aimed at finding hyperplanes which have "optimal stability" or - in our terms - maximal margin.

One of them, the Adatron, comes with theoretical guarantees of convergence to the optimal solution, and of a rate of convergence exponentially fast in the number of iterations [2, 15], provided that a solution exists.

We demonstrate that such models can be adapted, with the introduction of kernels, to operate in a high-dimensional feature space, and hence to learning non-linear decision boundaries. This provides a procedure which emulates SV machines but doesn't need to use the quadratic programming toolboxes.

In this section we will briefly sketch the Adatron algorithm, and we will list the theoretical results which can be proved for it (in the Statistical Mechanics framework), pointing to the relevant papers for the proofs of the theorems. Finally we will show how it is possible to introduce the kernels. The next section will be devoted to experimental comparisons between KA and SV machine, and to benchmarking.

The Adatron is a an on-line algorihm for learning perceptrons which has an attractive fixed point corresponding to the maximal-margin consistent hyperplane, when this exists.

By writing the Adatron in the data-dependent representation, and by substituting the dot products with kernels, we obtain the following algorithm:

The Kernel-Adatron Algorithm.

1. Initialise $\alpha_i = 1$.

2. Starting from pattern $i = 1$, for labeled points (x_i, y_i) calculate $z_i = y_i \sum_{j=1}^{p} \alpha_j y_j K(x_i, x_j)$.

3. For all patterns i calculate $\gamma_i = y_i z_i$ and execute steps **4** to **5** below.

4. Let $\delta\alpha^i = \eta(1 - \gamma^i)$ be the proposed change to the multipliers α^i.

5.1. If $(\alpha^i + \delta\alpha^i) \leq 0$ then the proposed change to the multipliers would result in a negative α^i. Consequently to avoid this problem we set $\alpha^i = 0$.

5.2 If $(\alpha^i + \delta\alpha^i) > 0$ then the multipliers are updated through the addition of the $\delta\alpha^i$ i.e. $\alpha^i \leftarrow \alpha^i + \delta\alpha^i$.

6. Calculate the bias b from

$$b = \frac{1}{2}\left(\min\left(z_i^+\right) + \max\left(z_i^-\right)\right)$$

where z_i^+ are those patterns i with class label $+1$ and z_i^- are those with class label -1.

7. If a maximum number of presentations of the pattern set has been exceeded then **stop**, otherwise return to step **2.**

The kernel $K(x, x')$ can be any function satisfying Mercer's condition, in particular it is possible to use RBF or polynomial kernels given in section 2.

Important Remarks

Using results reported in the Statistical Mechanics literature, the following important properties of the Adatron can be derived:

1. (Anlhauf and Biehl [2]) *Every stable point for the Adatron algorithm is a maximal margin point and vice versa.*

Proof Sketch

By inserting the Kuhn-Tucker conditions for the maximal margin ($\alpha_i > 0 \Leftrightarrow \gamma_i = 1$, $\alpha_i = 0 \Leftrightarrow \gamma_i > 1$) in the Adatron updating rule it follows that the optimal margin is a fixed point. Vice versa by imposing $\delta\alpha_i = 0\ \forall i$ the Kuhn-Tucker conditions are obtained.

2. (Anlhauf and Biehl [2]) *The algorithm converges in a finite number of steps to a stable point if a solution exists.*

Proof Sketch The functional

$$L = \sum_{i=1}^{p} \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

can be shown to be upper bounded, and to increase monotonically at each updating step of the Adatron. So it has to find a fixed point in a finite number of steps.

3. (Opper [16], [15]) *The rate of convergence to the optimal solution follows an exponential law in the number of iterations.*

The proof makes use of replica calculations from Statistical Mechanics (and the standard assumptions of that model [25]).

Note: The convergence proof relies on an adequate choice of η, which also controls the speed of the convergence itself. The issues regarding the choice of η cannot be discussed here for lack of space, but we observe that the theory provides an interval within which a valid η can be chosen. Results will be presented elsewhere.

4 EXPERIMENTAL RESULTS

We have evaluated the performance of the KA algorithm with gaussian kernels on a number of standard classification datasets, both artificial and real. The artificial datasets include the two-spirals problem [8], n-parity [14], mirror symmetry [14]. The real world data include the sonar classification problem [9], the Wisconsin breast cancer dataset [23] and a database of handwritten digits collected by the US Postal Service [12].

4.1 THE TWO SPIRALS PROBLEM AND n-PARITY

For the two spirals problem the task is to discriminate between two sets of points which lie in two spirals in a plane.

The solution found by the KA algorithm is illustrated in Figure 2 and compared with the solution provided by a kernel-perceptron, i.e. a generic hyperplane in the feature space (Fig. 1).

The diagrams present different decision functions; in the kernel-perceptron's case the small margin yields a highly non smooth boundary while for the KA algorithm a smooth and centered solution has been found.

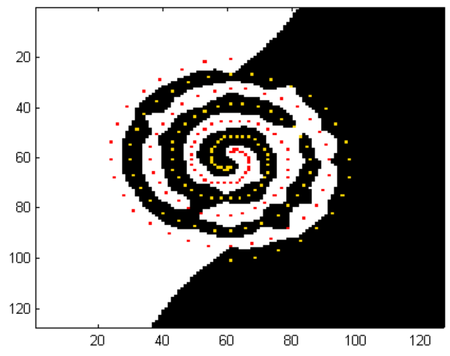

Figure 1: Kernel-Perceptron (small margin) clearly overfits

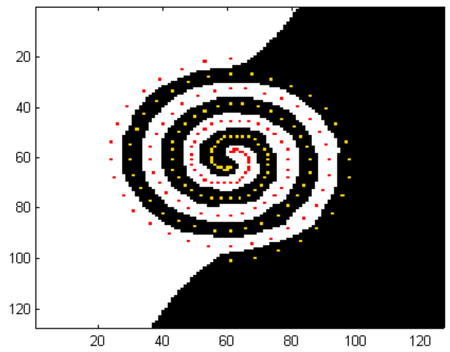

Figure 2: Kernel-Adatron learns a much smoother boundary

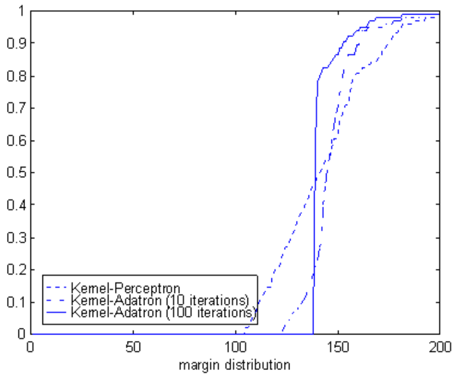

Figure 3: Cumulative margin distribution for kernel-perceptron and for KA. Note the scaling of the margins: we denote with 100 the null margin (points on the boundary).

Useful insight about the differences between these two learning machines can be obtained by observing the margin distribution graphs in Fig. 3, which present the cumulative distribution of the margins of all individual training points, i.e. the fraction of patterns (vertical axis) which have a margin larger than a given value θ (horizontal axis). It is interesting to note that the effect on the margin distribution of the training in KA is similar to the one in Adaboost, discussed in [18].

The solution for the n-parity problem [14], which is hard to separate for neural networks, was found in 1 epoch for n = 3 and n = 6, while it took respectively 3 and 5 epochs to maximise the margin.

4.2 MIRROR SYMMETRY

In the mirror symmetry problem [14] the output y is a 1 if the input pattern x (with components from {-1, +1}) is exactly symmetrical about its centre, otherwise the output is a -1. For randomly constructed input strings the output would be a -1 with a high probability. Consequently the labels ± 1 are selected with a 50% probability and the first half of the input string is randomly constructed from components in {-1, +1} (both selected with a 50% probability) and the second half of the string is symmetrical or random depending on the target value given. Generalisation was evaluated using a test set drawn from the same distribution (eliminating any instances for which the input string is identical to a member of the training set).

In Figure 4 we plot the generalisation error on the test set (100,000 examples including repetitions) versus σ for the KA algorithm trained to 200 epochs with η = 1.0. 200 training examples were used with input strings consisting of 30 components. The generalisation error passes through a mimimum between $\sigma = 4 - 5$ with a maximum generalisation of 95.1%. To compare with other algorithms in a machine independent way we have implemented all algorithms in MATLAB (using its optimization toolbox) and estimated the individual speeds using FLOPS (Table 1). We see that the KA algorithm is substantially faster that Support Vector machines while also having a comparable generalisation performance to the latter (TR is the number training errors, TS the number of test errors on a set of 100 patterns). It also performs much better than k-nearest neighbour (kNN) on the test set.

Alg.	EP.	σ	TR.	TS.	FLOPS
kNN(k=1)	-	-	0	25	0
kNN(k=7)	-	-	0	22	0
SVM	-	3.5	0	3	0.173×10^9
SVM	-	4.2	0	2	0.318×10^9
SVM	-	5.0	0	5	0.694×10^9
KA	10	3.5	0	4	1210000
KA	10	4.2	0	5	1210000
KA	10	5.0	0	6	1210000
KA	100	3.5	0	3	12100000
KA	100	4.2	0	3	12100000
KA	100	5.0	0	4	12100000
KA	250	3.5	0	3	30250000
KA	250	4.2	0	4	30250000
KA	250	5.0	0	5	30250000

Table 1: comparison for mirror symmetry

4.3 SONAR CLASSIFICATION

The sonar classification problem of Gorman and Sejnowski [9] consists of 208 instances formed by 60 analogue inputs, representing returns from a roughly cylindrical rock or a metal cylinder, equally divided into training and test sets. For the aspect-angle dependent dataset [9] they trained a standard back-propagation neural network with 60 inputs and 2 output nodes. Experiments were performed with up to 24 hidden nodes and each neural network was trained with 300 epochs through the training set. Their results are reproduced in Table 2.

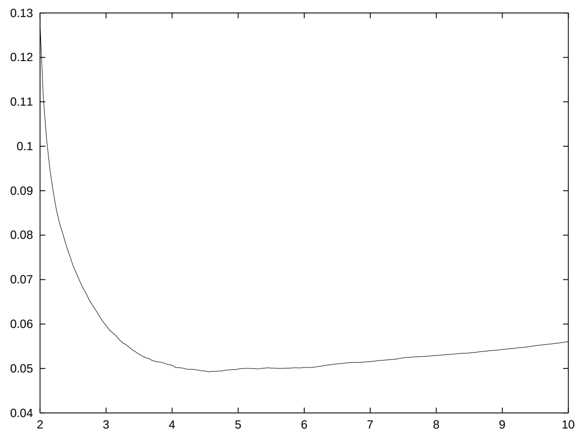

Figure 4: Generalisation error (vertical axis) vs. σ (horizontal axis): mirror symmetry problem

# hidden	0	2	3	6	12	24
% gen.	73.1	85.7	87.6	89.3	90.4	89.2

Table 2: Gorman and Sejnowski results for sonar

For the KA algorithm we plot σ against generalisation error in Figure 5 and the best generalisation performance is 95.2% by comparison. The KA algorithm is also very fast. Figure 6 illustrates the approach of the margin towards 1 (for $\sigma = 1.0$ and $\eta = 1.0$). The training error fell to 0 in the second epoch (it was 0.077 at the end of the first epoch). We also show the generalisation error versus number of epochs (Figure 7). As for mirror symmetry we give a comparison with Support Vector Machines in Table 3.

Alg.	EP.	σ	TR.	TS.	FLOPS
kNN(k=1)	-	-	0	10	0
kNN(k=3)	-	-	0	19	0
SVM	-	0.57	0	8	3.476×10^9
SVM	-	0.71	0	7	6.750×10^9
SVM	-	0.85	0	7	8.878×10^9
KA	10	0.57	0	6	329680
KA	10	0.71	0	9	329680
KA	10	0.85	0	8	329680
KA	100	0.57	0	6	3296800
KA	100	0.71	0	6	3296800
KA	100	0.85	0	7	3296800
KA	250	0.57	0	6	8242000
KA	250	0.71	0	6	8242000
KA	250	0.85	0	7	8242000

Table 3: comparison for sonar classification

4.4 WISCONSIN BREAST CANCER DATASET

The Wisconsin breast cancer dataset contains 699 patterns with 10 attributes for a binary classification task (the tumour is malignant or benign).

This dataset has been extensively studied by other authors. CART gives a generalisation of 94.2%, an RBF neural network gave 95.9%, a linear discriminant method gave 96.0% and a multi-layered neural network (trained via Back-Propagation) 96.6% (all the results have been obtained using 10-fold cross-validation [23]). Our optimal test performance was of 99.48%, which is superior to the previous reported results. However we regard this result as simply indicating that we are comparable with other approaches, as this difference can also be due to other factors and requires further investigation. Among them are differences in the handling of instances with missing values (16 in the database), in the preprocessing (we have removed the first column of the database reporting the patient's code number, like some other authors) and in the choice of σ. We note that the test error is insensitive to the choice of σ in a broad interval, as can be seen in Fig. 11. In this diagram we give a plot of generalisation error versus σ for 10-fold cross validation on the 699 instances (50 iterations were used and $\eta = 1.0$).

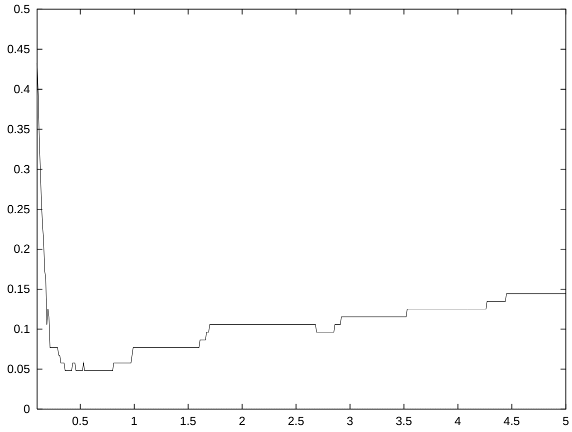

Figure 5: Generalisation error of KA (vertical axis) vs. σ (horizontal axis) for the sonar classification problem.

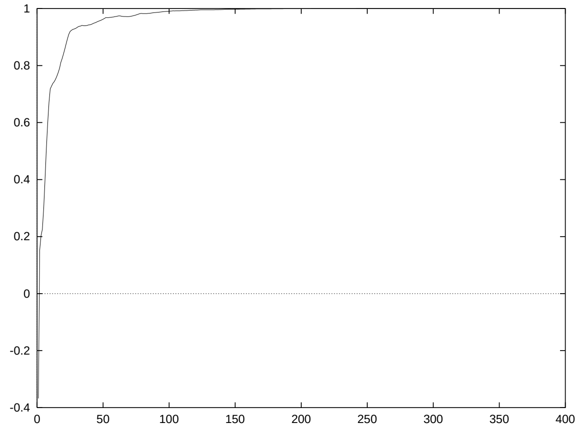

Figure 6: Margin (vertical axis) vs. number of epochs (horizontal axis) for sonar classification ($\sigma = 1.0$, $\eta = 1.0$).

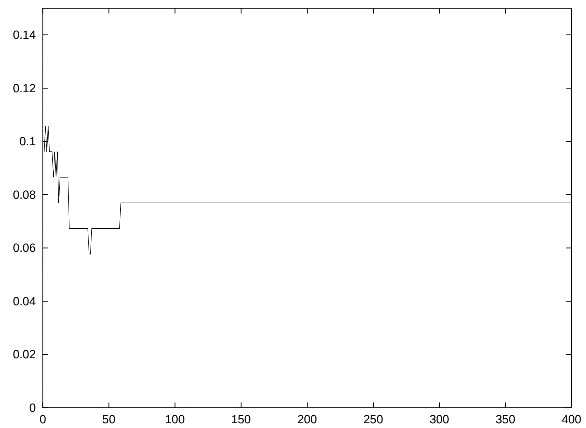

Figure 7: Generalization error (vertical axis) vs. number of epochs (horizontal axis) for sonar classification ($\sigma = 1.0$, $\eta = 1.0$).

Furthermore, for a particular split of the database with 550 training examples and 149 test examples, $\sigma = 3.2$ and $\eta = 1.0$, we give plots of the generalisation error (Fig. 8), margin (Fig. 9) (all versus number of epochs and the final spectrum of α values (Figure 10).

To compare the computational cost of KA with other classifiers we have used a matlab implementation of them, and run it on a reduced subset of the database (199 training and 168 testing points) using the FLOPS as an indication of the algorithmic complexity. The results are reported in Table 4 and indicate that KA can achieve about the same generalization performance of SV machines at a cost which is orders of magnitude smaller.

Alg.	EP.	σ	TR.	TS.	FLOPS
kNN(k=1)	-	-	0	13	0
kNN(k=3)	-	-	0	9	0
SVM	-	0.28	0	11	2.4541×10^9
SVM	-	0.35	0	10	2.7763×10^9
SVM	-	0.42	0	11	2.8043×10^9
KA	10	0.28	0	8	1197980
KA	10	0.35	0	9	1197980
KA	10	0.42	0	11	1197980
KA	100	0.28	0	10	11979800
KA	100	0.35	0	9	11979800
KA	100	0.42	0	9	11979800
KA	250	0.28	0	10	29949500
KA	250	0.35	0	9	29949500
KA	250	0.42	0	10	29949500

Table 4: comparison for cancer classification (a subset has been used for this comparison

4.5 US POSTCODE DATABASE

The benchmarking of classification algorithms of the class of SV machines has traditionally been performed using the database of handwritten digits from US Postal Codes [12, 20].

This dataset consists of a training set of 7,291 examples and a test set of 2,007. Each digit is given by a 16×16 vector with components which lie in the range -1 to 1. In this experiment we have performed two-class classification i.e. separating a particular digit from the others. To find suitable values for σ the training set was split into a smaller training set of 6,000 examples and a validation set of 1,291. The best value of σ was found by evaluating performance on the validation set across the range $(1, 10)$. The full training set of 7,291 was then used with the selected value of σ to train the system to classify each digit.

The results are shown in Table 5 where the last column shows the best value of σ found from the validation

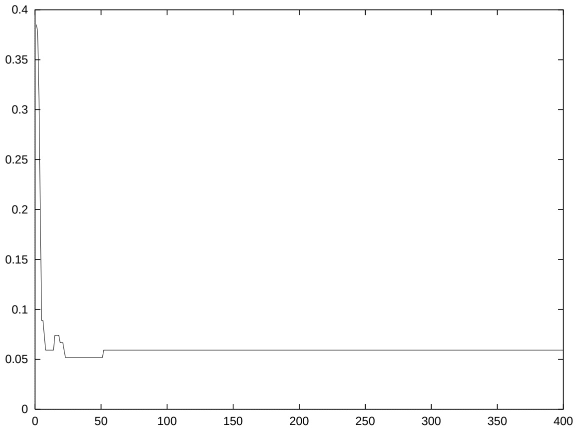

Figure 8: Generalization error (vertical axis) vs. number of epochs (horizontal axis) for cancer classification

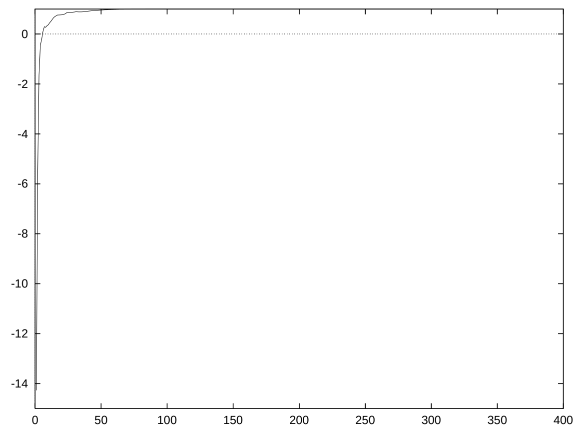

Figure 9: Margin (vertical axis) vs. number of epochs (horizontal axis) for cancer classification

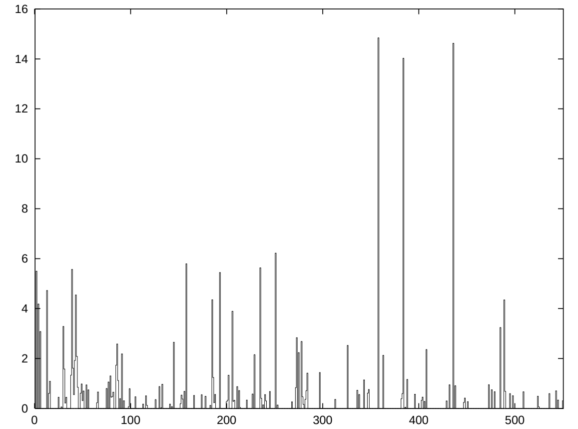

Figure 10: Spectrum of α values for the 550 patterns found by the KA algorithm (cancer classification experiment

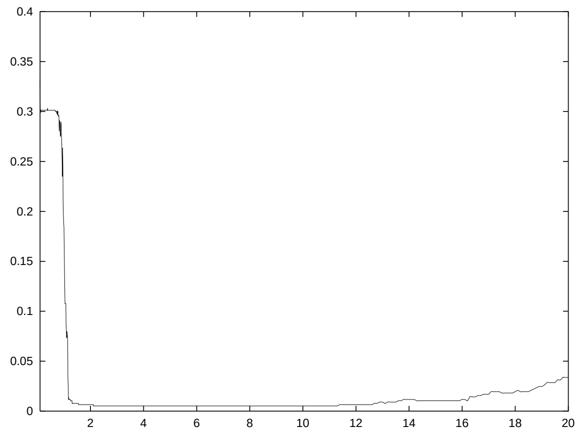

Figure 11: Generalisation error (vertical axis) vs. σ (horizontal axis) for cancer classification ($\eta = 1.0$)

study. The other columns show the number of errors on the test set of 2,007 examples for the KA algorithm and 3 comparative algorithms as reported by [20]. The latter three algorithms are an RBF neural network, a Support Vector Machine (SVM) and a hybrid model in which the support vectors found by the SVM are used as the centers of receptive fields in an RBF network [20].

Digit	RBF	SVM	Hybrid	KA	σ
0	20	16	9	13	1.8
1	16	8	12	10	1.6
2	43	25	27	21	2.4
3	38	19	24	24	2.0
4	46	29	32	26	4.0
5	31	23	24	19	1.8
6	15	14	19	15	2.4
7	18	12	16	11	2.8
8	37	25	26	26	3.2
9	26	16	16	14	1.6

Table 5: comparative performance on the USPS database (number of errors in a 2007 points test set)

The performance of KA is comparable with the other algorithms.

5 CONCLUSIONS AND FUTURE WORK

We have presented an algorithm which finds maximal margin hyperplanes in a high dimensional feature space, emulating Vapnik's Support Vector machines.

Experiments performed on artificial and real data show that the generalization performance of this algorithm is comparable with that of SV machines, while the computational cost of finding the hypothesis is significantly smaller. Also, the introduction of kernels into the Adatron provides a very simple, compact and robust algorithm.

Further work is now needed to introduce the capability of tolerating training errors, so that the machine can deal with outliers and noisy datasets, following the soft-margin approch used in SV machines.

Acknowledgements This work was partially funded by EPSRC. Nello wishes to thank John Shawe-Taylor and Jason Weston for many useful discussions. Thilo would like to thank Rob Harrison and Klaus-Robert Müller, and Univ. of Sheffield/AC&SE for support.

References

[1] Aizerman, M., Braverman, E., and Rozonoer, L. (1964). Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning, *Automations and Remote Control*, **25**:821-837.

[2] Anlauf, J.K., and Biehl, M. (1989). *Europhysics Letters* **10**:687

[3] Bartlett P., Shawe-Taylor J., (1998). Generalization Performance of Support Vector Machines and Other Pattern Classifiers. 'Advances in Kernel Methods - Support Vector Learning', Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola (eds.), MIT Press, Cambridge, USA.

[4] Boser, B., Guyon, I., Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Fifth Annual Workshop on Computational Learning Theory.* ACM Press.

[5] Cortes, C., and Vapnik, V. (1995). Support Vector networks, *Machine Learning* 20:273-297.

[6] Cortes, C. (1995). *Prediction of Generalization Ability in Learning Machines.* PhD Thesis, Department of Computer Science, University of Rochester.

[7] Cristianini, N., Shawe-Taylor, J., Sykacek, P., (1998). Bayesian Classifiers are Large Margin Hyperplanes in a Hilbert Space, in Shavlik, J., ed., *Machine Learning: Proceedings of the Fifteenth International Conference*, Morgan Kaufmann Publishers, San Francisco, CA.

[8] FrießT-T., Harrison R.F., (1998), Pattern Classification using Support Vector Machines, Eng. and Int. Sys. (EIS98 Conf. Proc.)

[9] Gorman R.P. & Sejnowski, T.J. (1988) *Neural Networks* **1**:75-89.

[10] Guyon, I., Matic, N., & Vapnik, V. (1996). Discovering Informative Patterns and Data Cleaning, *Advances in Knowledge Discovery and Data Mining* ed by U.M.Fayyad, G. Piatelsky-Shapiro, P. Smyth and R. Uthurusamy AAAI Press/ MIT Press.

[11] Kinzel, W.,(1990) Statistical Mechanics of the Perceptron with Maximal Stability. *Lecture Notes in Physics* (Springer-Verlag) **368**:175-188.

[12] LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Muller, U. A., Sackinger, E., Simard, P. and Vapnik, V., (1995). Comparison of learning algorithms for handwritten digit recognition, *International Conference on Artificial Neural Networks*, Fogelman, F. and Gallinari, P. (Ed.), pp. 53-60.

[13] Krauth W. and Mezard. M. (1987) *J.Phys.* **A20**:L745

[14] Minsky M.L. & Papert, S.A. (1969) *Perceptrons*, MIT Press: Cambridge.

[15] Opper, M. (1988). Learning Times of Neural Networks: Exact Solution for a Perceptron Algorithm. *Physical Review* **A38**:3824

[16] Opper, M. (1989). Learning in Neural Networks: Solvable Dynamics. *Europhysics Letters*, **8**:389

[17] Osuna E., Freund R., Girosi F., (1997) "Training Support Vector Machines: An Application to Face Detection", Proc. Computer Vision and Pattern Recognition '97, 130-136

[18] Schapire. R., Freund, Y., Bartlett, P., & Sun Lee, W. (1997). Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods, *Proceedings of International Conference on Machine Learning.*

[19] Schoelkopf, B., (1997). *Support Vector Learning.* PhD Thesis. R. Oldenbourg Verlag, Munich.

[20] Schoelkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., and Vapnik, V., Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers, M.I.T. Preprint (A.I. Laboratory), A.I. Memo No. 1599.

[21] Shawe-Taylor, J., Bartlett, P., Williamson, R. & Anthony, M. (1996). Structural Risk Minimization over Data-Dependent Hierarchies NeuroCOLT Technical Report NC-TR-96-053 (`ftp://ftp.dcs.rhbnc.ac.uk /pub/neurocolt/tech_reports`).

[22] `(Sonar dataset)` `http://www.boltz.cs.cmu.edu/benchmarks/sonar.html`

[23] Ster, B., & Dobnikar, A. (1996) Neural networks in medical diagnosis: comparison with other methods. In A. Bulsari et al. (ed.) *Proceedings of the International Conference EANN'96*, p. 427-430.

[24] Vapnik, V. (1995) *The Nature of Statistical Learning Theory*, Springer Verlag.

[25] Watkin, T., Ran, A. & Biehl, M. (1993). The Statistical Mechanics of Learning a Rule, *Rev. Mod. Phys.* 65(2).