The Complexity of Learning with Supportvector Machines - A Statistical Physics Study

Manfred Opper

Aston University, School of Engineering and Applied Science, Birmingham B4 7ET, United Kingdom

1 Introduction

Supportvector Machines (SVMs) have been introduced in recent years by V. Vapnik [1,2] and his collaborators as a novel approach in machine learning (for detailed reviews, see e.g. [3–5]. SVMs seem to have a variety of advantages over other learning machines which make them an important alternative to neural networks. E.g., the training of SVMs is performed by minimizing a cost function which has a single minimum only. Despite the mathematical simplicity and elegance of SVM training, SVMs are able to learn tasks of high complexity. In fact, unlike neural networks, which have a fixed number of adjustable parameters, the flexibility of an SVM is not fixed in advance but can grow with the data. This seems like a very pleasant feature because it allows SVMs to memorize arbitrary sets of examples perfectly. However, such an unbounded flexibility could present also a potential danger. For practical applications, when the number of training examples grows large, we would like the learning process to converge to a solution which makes the machine able to generalize well on data that it has not observed before An SVM on the other hand, observing an increasing number of examples might generate hypotheses about the learning task which become more and more complex by exploring more and more of its unbounded learning capabilities. Can we expect that such a process will actually converge to a satisfactory solution when the actual task to be learnt is of *finite* (but unknown) complexity?

To adress this question, we present results of an analysis which is based on a theoretical approach from Statistical Physics. As discussed in the contributions of Michael Biehl and Wolfgang Kinzel in this volume, the method has been already applied successfully to a variety of neural network learning problems (for a detailed review of methods and results, see e.g [6,7]). It provided exact results for the performance of such machines in controlled *analytical* experiments on statistical ensembles of *typical* learning tasks. The approach will allow us to study the generalization performance of SVMs in an important limit where the dimensionality of the space of input features grows large.

2 From Perceptrons to Supportvector Machines

The basic idea behind SVMs is best explained from the simplest neural network, the perceptron (see also Michael Biehl's contribution to this volume). Its output to an N dimensional vector $\mathbf{x} = (x_1, \ldots, x_N)$ of input features x_i is a binary label $y = \pm 1$ which is computed as

$$y = \mathrm{sign}\left[\sum_{i=1}^{N} w_i x_i\right] \quad . \tag{1}$$

The output y indicates to which of the two classes (-1 or $+1$) the input $\mathbf{x}$ is assigned by the perceptron. The numbers w_i give different weights to each input feature and are adjusted in such a way that the perceptron performs well on a set of input-output examples encountered during the network's training phase.

The perceptron is of limited capability as a classifier because it is able to learn only those datasets perfectly which are *linearly separable*. In such a case, *Rosenblatt's perceptron learning algorithm* (see also Wolfgang Kinzel's contribution to this volume) is guaranteed to find a vector $\mathbf{w} = (w_1, \ldots, w_N)$ of weights which specifies an $N-1$ dimensional hyperplane perpendicular to $\mathbf{w}$ which perfectly separates positive from negative labeled examples.

To overcome the limitations of perceptrons we might replace the simple linear input features x_i by more general features $\Phi_\mu(\mathbf{x})$ which are fixed *nonlinear* functions of the input vector $\mathbf{x}$. Such a *modified perceptron* would compute its output as

$$y = \mathrm{sign}\left[\sum_{i=1}^{M} w_\mu \Phi_\mu(\mathbf{x})\right] \quad . \tag{2}$$

If we take e.g. for the set of Φ_μ's linear and quadratic monomials in the x_i's, [1] the two classes of examples can now be separated by quadratic decision surfaces which are much more flexible than the linear hyperplanes provided by the simple perceptron. Nevertheless, the machine would be still trainable by a Rosenblatt type of algorithm. But since we don't know beforehand which of the $M \doteq N + N(N+1)/2$ linear and nonlinear features are actually needed for a given task, we would have to use all of them, leaving us with the problem of a vast increase in the number of unknown parameters which have to be adjusted to the data. One would not expect to get statistically reliable values for the huge number of weights w_μ, when the number of training data is limited.

A short glance at the mathematics of Rosenblatt's algorithm as applied to the modified perceptron (2) shows that the problem may be somewhat less severe than it seems. Each instance, when an input-ouput example $(\mathbf{x}, y)$ is presented to the perceptron, the change of each weight w_μ is proportional to $y\Phi_\mu(\mathbf{x})$, whenever the perceptron's response to the example was wrong. Hence, *after* successful training (starting from $\mathbf{w} = 0$) all weights w_μ will be a weighted sum

[1] ie. expressions of the type x_k and $x_k x_l$

of the m examples of the form

$$w_\mu = \sum_{k=1}^{m} \alpha_k y_k \Phi_\mu(\mathbf{x}^k) \ . \tag{3}$$

α_k is the number of times that the perceptron responded wrongly to example k. Obviously, the m numbers α_k have nonnegative values and some of them might be even zero. The inputs $\mathbf{x}^k$ with nonzero α_k are called the *supportvectors*. We may regard the α_k's as new, effective parameters of the learning machine rather than the huge number of weights w_μ. The reduction of the number of nonzero parameters to a number not exceeding the size of the set of training data (but typically growing with it) already shows the adaptive flexibility of the modified perceptron.

It turns out that there are simple and direct ways of dealing with the practical computations of the α_k's. Using (3), the sum (2) can always be expressed entirely in terms of the α_k's and a positive definite $m \times m$ matrix, the *kernel* matrix, which for any pairs of input vectors $\mathbf{x}$ and $\mathbf{x}'$ is defined as

$$K(\mathbf{x}, \mathbf{x}') = \sum_{\mu=1}^{M} \Phi_\mu(\mathbf{x}) \Phi_\mu(\mathbf{x}') \ . \tag{4}$$

This simplification holds for all operations that are required for learning and predictions with the "kernel perceptron". They can be formulated entirely in terms of the α_k's and the kernel matrix (4). More surprising, there is not even a need to specify the set of feature functions $\Phi_\mu(\mathbf{x})$ explicitely. Each function $K(\mathbf{x}, \mathbf{x}')$ of two variables which has the mathematical property of being *positive definite* can be expressed as a scalar product of the type (4) for some set of feature functions $\Phi_\mu(\mathbf{x})$. Since the scalar product (4) measures correlations between the features of two input vectors, kernels usually express some sort of similarity between two inputs $\mathbf{x}$ and $\mathbf{x}'$. E.g., in practice, often *radial basis function* (RBF) kernels which depend on the *distance* between two inputs and are of the form

$$K(\mathbf{x}, \mathbf{x}') = e^{-\lambda ||\mathbf{x} - \mathbf{x}'||^2} \tag{5}$$

are used. In the latter case, the number M of implicit features $\Phi_\mu(\mathbf{x})$ equals *infinity*! For more specialized learning problems, kernels have been invented which measure similarities of more complex input objects such as textstrings or strings of DNA.

While this preliminary construction already contains most of the basic ideas behind the SVM, one more important modification is needed. Although Rosenblatt's algorithm would provide us with a hyperplane in the space of features Φ_μ which separates the two classes of examples, there are (infinitely many) other hyperplanes (obtained e.g. by slightly tilting the original one) which also would do the same job. Among all the planes with this property, one defines the *maximal margin* hyperplane in such a way that the distance between the plane and the closest datapoint is maximized. Intuitively, one might expect that this is

beneficial for the machine's generalization ability, because the resulting decision surface will be quite *robust* against small changes (e.g. noise) in the dataset. The optimality criterion defining the maximal margin plane is mathematically expressed by a quadratic optimization problem for the parameters α_k which can be solved efficiently with a variety of algorithms.

3 Statistical Physics and the Complexity of Learning

The lerning approach provided by SVMs seems almost like a miracle. By choosing a proper kernel, say an RBF kernel (5), the set of implicit features $\Phi_\mu(\mathbf{x})$ contains functions of arbitrarily high complexity, which will allow the SVM to reproduce the desired classification on any set of training examples perfectly without an error. But it is not clear if this memorization will actually lead to a good understanding of the unknown *rule* that is behind the example data. Being exposed to an increasing amount of training data, the SVM might compute a decision surface of increasing complexity which becomes so wiggly that its predictions to novel inputs become equivalent to random guesses. How much do we have to pay for not knowing the complexity of the learning task but use a machine of unbounded flexibility?

We will investigate this question using an approach of Statistical Physics. Here, one does not specify a single learning problem but rather a *statistical ensemble* of such problems for which we can define our choices of "typicality" and complexity by the setting of a few control parameters. For specific types of data distributions we are able to compute a set of *order parameters*, which describe the global behaviour of SVMs trained on our ensemble of learning problems in the "thermodynamic limit" where the numbers of tunable parameters and examples are both large. This approach seems to resemble asymptotic methods known from mathematical statistics. However, while the latter methods are valid when the sizes of the training data sets are much larger than some intrinsic complexity of a learning machine, the *thermodynamic limit* of Statistical Physics allows us to simulate even some effects of small *relative* sample sizes.

We will begin by specifying an ensemble of learning tasks with an adjustable complexity. The learning task is modeled by using an ideal classifier (often called the *teacher*) which provides us with the correct outputs to a set of randomly generated input data. This teacher is defined by a generalized perceptron of the type (2) which computes its output as

$$y = \mathrm{sign}\left(\sum_j B_\mu \Phi_\mu(\mathbf{x})\right) \equiv \mathrm{sign}\left[\frac{1}{N}\sum_i B_i x_i + \frac{1}{N^2}\sum_{ij} B_{ij} x_i x_j\right] \quad . \qquad (6)$$

That is, the features Φ_μ consist of all N linear and all $N(N+1)/2$ quadratic monomials of the form $\frac{1}{N}x_k$ and $\frac{1}{N^2}x_k x_l$ that can be built from arbitrary components x_k and x_l of the input vector $\mathbf{x}$. We will define a "typical" task by chosing the B_i's and the B_{ij}'s as well as the components x_k^l of the training vectors $\mathbf{x}^l$ to be ± 1 independently at random. In this case, the scaling of the features with

N guarantees that the contribution of linear and quadratic features to the total output are typically of the same order of magnitude when N grows large. [2] This represents a model where the learning task contains a number of simpler elements (the N linear features) and a much larger number of finer, more complex details. Both parts influence the overall decision by roughly the same amount. The perfect knowledge of the linear part of the task would already allow us to predict the correct classification for inputs with a probability that is sufficiently larger than $\frac{1}{2}$, the value for random guessing. We can also easily model a task which contains only the simple elements by setting all $B_{ij} = 0$, corresponding to the nonlinear features.

An SVM "student" that is well matched to these types of teachers is constructed from the same set of features. As a corresponding kernel (4) we take

$$K(\mathbf{x}, \mathbf{x}') = \frac{1}{2}\frac{\mathbf{x}\cdot\mathbf{x}'}{N} + \frac{1}{2}\left(\frac{\mathbf{x}\cdot\mathbf{x}'}{N}\right)^2 . \tag{7}$$

The calculation of learning curves for SVMs within the Statistical Physics approach proceeds in a similar way as for other neural networks by defining a suitable statistical ensemble of kernel perceptrons which are all learning the same set of training data. The statistical weight of the members of the ensemble increases with their margin, and the variability of margins is controlled by a "temperature"like parameter. At the end of the calculation, the temperature is set to zero, such that all weight is concentrated on the SVM, ie. the kernel perceptron with maximal margin.

4 Results

The results of the calculations are expressed by a set of order parameters that are functions of the student's weights w_μ and the teacher's weights B_μ. The precise definition of these order parameters can be found in [8]. There is, however, a crucial difference. For the SVM, we can choose different ways to define a proper thermodynamic limit. Taking different limits allows us to zoom into different stages of the learning process.

Our experience with the Statistical Physics results for learning with neural networks suggests that we should supply the SVM with a number of examples that is proportional to the total number of features in the learning task (which is about N^2) and thus fix the ratio $\alpha^{(2)} \equiv \frac{m}{N^2}$, when we let N approach ∞. When the ratio $\alpha^{(2)}$ grows, the generalization error (which is the error on test data that were not in the set of training data) should decrease monotonically, and asymptotically approaching zero as $\alpha^{(2)}$ grows larger and larger. In fact, both our Statistical Physics theory and simulations [3] support these intuitive assumptions (see the upper curve on the right hand side of Figure 1.). A more interesting

[2] Since there are many more nonlinear features than linear ones, the contribution of the nonlinear ones are downweighted by the N dependent prefactor.

[3] Simulations were performed with $N = 201$ and averaged over 50 random samples.

problem is adressed when we keep the number of examples m much smaller, of the order $m = \alpha^{(1)} N$, ie. of the order of the the N simple linear features. Will the SVM at this scale be able to learn the simple part of the learning task, i.e. the linear features? Or will its ability to generalize be completely destroyed by the fact that most of its weights w_μ, those corresponding to the huge number of nonlinear features take random values? The left side of Figure 1. (upper curve)

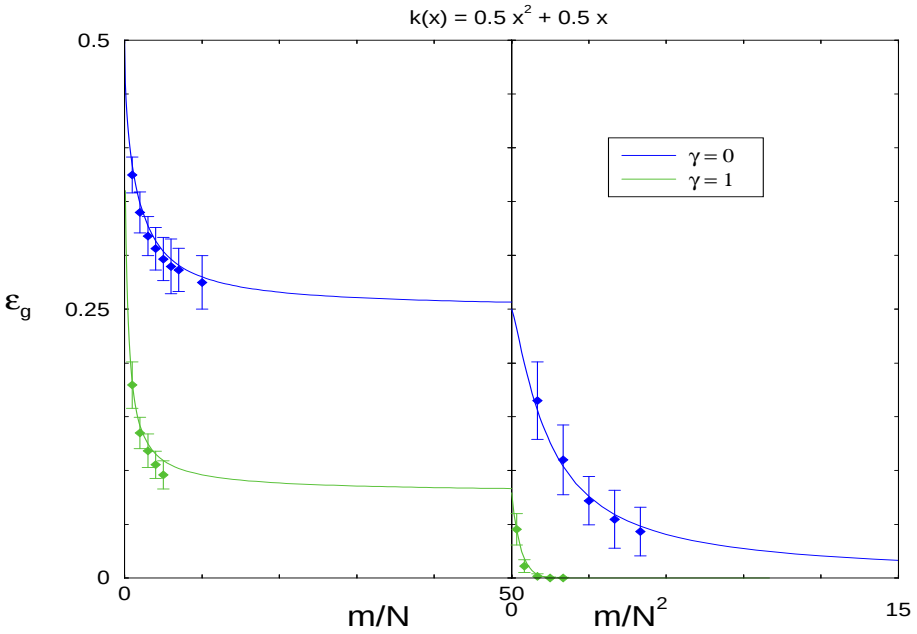

Fig. 1. Decrease of the generalization error on different scales of examples, for a random task ($\gamma = 0$) and a task which is correlated to the teacher ($\gamma = 1$). The definition of γ is given in section 5.

shows the results of our analysis for this type of thermodynamic limit scaling. The SVM actually begins to understand the linear part of the task already at this small scale of examples and is able to reduce its generalization error remarkably from the random guessing value $\varepsilon = \frac{1}{2}$ with increasing α. Since there is not enough information from the data to infer the remaining N^2 weights of the teacher's quadratic features, the generalization error of the SVM reaches a nonzero plateau as $\alpha^{(1)}$ grows large. A careful analysis of the order parameters shows that the weights w_μ corresponding to the quadratic features actually are random and do no show any resemblance to the corresponding teacher weights. Nevertheless, they are too small to destroy the alignment of the weights w_i

corresponding to the linear features of the "student" SVM with their teacher counterparts B_i.

In reality, the dimensionality N of the input vector is never infinite. A recent Statistical Physics study [9] has taken this problem into account. It provides results for learning curves at large but finite N which smoothly interpolate between the two different scaling regions.

The preceding results give us an idea about what happens if the learning task has a much lower complexity than the SVM. If we completely remove all nonlinear features from the teacher's task but still learn the examples with the much more complex SVM we are able to achieve an arbitrarily small generalization error on a scale $m = \alpha^{(1)}N$. Although the "nonlinear" weights of the SVM are found to be nonzero and random, they are too small to destroy the generalization ability. Their overall effect is a slowing down of the decay of the generalization error compared to a learning process which uses a simple linear SVM (a perceptron), ie. a machine whose flexibility is optimally matched to the teacher's task.

5 Towards more realistic Models

We hope that our simple model of learning explains at least qualitatively the robustness of SVMs in dealing with a variety of classification problems with an *a priori* unknown complexity. Focussing on a simple average case analysis, our model presents a complementary viewpont to that of other approaches in computational learning theory which are based on worst case scenarios [2,5]. E.g., our model and its solutions can be used to check the tightness of some of the rigorous bounds derived from the latter theories.

Nevertheles our approach can be criticized because our choice of a "typical" learning scenario is still far from being realistic. We will finish this contribution by mentioning two extensions of the basic model which aim at making the model more realistic.

First, we should not expect that the inputs in a classification problem fill a high dimensional space evenly. They might be rather concentrated in certain regions. Further, inputs and the "teacher" should not be chosen independently from each other. [4] If the distribution of inputs acts in a favourable way together with the teacher's learning task, learning is expected to progress faster. We have modeled such a situation by constructing an input distribution which is *correlated* with the teacher by being concentrated entirely inside a gap of size 2γ around the teacher's decision boundary, that is input vectors $\mathbf{x}^k$ are required to satisfy $\left|\sum_{\mu=1}^{M} B_\mu \Phi_\mu(\mathbf{x}^k)\right| < \gamma$.

For an SVM with kernel (7) learning from a quadratic teacher (6), we actually observe a faster decay of the generalization error than in the case of a uniform density. However, on the linear scale $m = \alpha^{(1)}N$ (left side of Figure 1,

[4] A realistic teacher should be able to observe some structure in the inputs which allow her/him to classify inputs well.

lower curve), the asymptotic decay towards the plateau is still relatively slow. A dramatic improvement occurs on the highest scale $m = \alpha^{(2)} N^2$, where the generalization error drops to zero faster than exponentially in $\alpha^{(2)}$ (right lower curve in Figure 1.).

Another possibility to make the model more realistic is to consider noise. This means that the classification task contains some degree of ambiguity or randomness such that it is impossible to predict with absolute certainty to which class an input must be assigned. E.g., even human experts are not able to classify any handwritten digit with a hundred percent success. We model such a problem by combining an "ideal", simple linear teacher (whose weights represent the best classifier that we can achieve) and a random process which changes the "ideal" outputs with some probability. In such a case, the SVM will need its full nonlinear complexity to fit the noisy training examples. In fact, a linear perceptron would not be able to learn a large number of noisy data. This might suggest that the decision surface between positive and negative examples created by the SVM could become more and more rugged which finally would make generalization impossible at all. Surprisingly, we find [10] that the teacher's weights can be perfectly learnt at the linear scale. This leads asymptotically to the smallest generalization error that is possible for the given noise.

This result is even more remarkable because other theoretical approaches in machine learning point out the importance to control the complexity of a learning machine in such a way that the difference between the error count on the training set and the generalization error is small. Such ideas are based on general bounds on the test error (see e.g. the article of John Shawe-Taylor and Nello Christianini in [4]). This has suggested to modify SVM training, forcing the SVM to commit errors on the training examples when there is noise in the data. In our noisy model, the training error equals zero and the generalization error is always positive. Neverheless, we achieve the best possible generalization ability asymptotically.

Acknowledgements: I would like to thank Rainer Dietrich, Dörthe Malzahn, Haim Sompolinsky and Robert Urbanczik for their pleasant collaboration on the Statistical Physics approach to Supportvector Machines. Their results and insights provided the basis for this contribution. Part of the work has been supported by EPSRC (grant no. GR/M81601) as well by DFG and the DAAD.

References

1. B. E. Boser, I. M. Guyon, and V. M. Vapnik: *A training algorithm for optimal margin classifiers* in: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144-152, Pittsburgh, PA, (ACM Press, 1992)
2. V. Vapnik *The Nature of Statistical Learning Theory* (Springer Verlag, 1995).
3. B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in Kernel Methods Support Vector Learning*: (The MIT Press, 1999)
4. A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmanns, (eds.): *Advances in Large Margin Classifiers* (The MIT Press, 2000)

5. N. Cristianini and J. Shawe-Taylor: *Support Vector Machines*, (Cambridge University Press, 2000).
6. A. Engel, and C. Van den Broeck, *Statistical Mechanics of Learning*, (Cambridge University Press, 2001).
7. H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing* (Oxford Science Publications, 2001)
8. R. Dietrich, M. Opper and H. Sompolinsky: *Statistical Mechanics of Support Vector Networks*, Phys. Rev. Lett. 82, 2975 (1999).
9. S. Risau-Gusman and M. Gordon. *Phys. Rev. E*, 62:7092–7099, 2000.
10. M. Opper and R. Urbanczik: *Universal learning curves of support vector machines*, Phys. Rev. Lett. 86, No 19, 4410-4413 (2001).