2	2	2	0	0
2	2		0	0
2	2	1	1	0
2/3		1	1	0
3	3	1	1	1

Fig. 5. Kohonen map where $0 =$ positive P wave, $1 =$ negative, $2 =$ bi-phase and $3 =$ double bump.

The proposed asymmetric basis function network was used as the feature extractor in a problem of P wave classification with a Kohonen network [16]. Defining the feature vector as

$$v = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ |\beta_2 - \beta_1| \\ \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{9}$$

and for a two-dimensional map of 5×5, the Kohonen network successfully clustered the four basic types of P waves: $0 =$ positive, $1 =$ negative, $2 =$ bi-phase and $3 =$ double bump signals (Fig. 5).

V. Conclusion

A simple neural network with only two asymmetric basis functions was shown to be an economical feature extractor for ECG P waves. The radial basis function network is known to be an universal approximator for continuous functions, but it may, eventually, require a large and variable number of basis functions in order to achieve a prescribed approximation error, when the P wave is M shaped or is strongly asymmetric. For this type of signals, the feedforward neural nets also require a large and variable number of hidden neurons, so that in the context of P wave classification, the proposed net with only two fixed number of asymmetric basis function seems to be a convenient feature extractor.

The proposed neural network can also deal with slow variations in the morphology of the signal with time when the training mechanism is kept active by the use of a small learning rate.

Acknowledgment

The authors are indebted to the anonymous reviewers for many valuable suggestions and to Dante Pazzanese Institute of Cardiology for some of the ECG data. The authors also gratefully acknowledge Dr. M. Zarrop of the Control Systems Centra, UMIST, U.K., for his contributions on the use of asymmetric basis functions.

References

[1] S. M. Bozic, *Digital and Kalman Filtering*. London, U.K.: Edward Arnold, 1979.

[2] E. Braunwald, *Heart Disease—A Textbook of Cardiovascular Medicine*. Philadelphia, PA: W. B. Saunders, 1996.

[3] G. M. Friesen, T. C. Jannet, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. Troy Nagle, "A comparison of the noise sensitivity of nine QRS detection algorithms," *IEEE Trans. Biomed. Eng.*, vol. 17, pp. 85–98, Jan. 1990.

[4] A. B. Geva, "Feature extraction and state identification in biomedical signals using hierarchical fuzzy clustering," *Med. Biol. Engr. Comput.*, vol. 36, no. 5, pp. 608–614, Sept. 1998.

[5] F. Gritzali, G. Frangakis, and G. Papakonstantinou, "Detection of the P and T Waves in an ECG," *Comput. Biomed. Res.*, vol. 22, no. 1, pp. 83–91, Feb. 1989.

[6] N. Maglaveras, T. Stamkopoulos, K. Diamantaras, C. Pappas, and M. Strintzis, "ECG pattern recognition and classification using nonlinear transformations and neural networks: A Review," *Int. J. Med. Inform.*, vol. 52, no. 1–3, pp. 191–208, Oct.–Dec. 1998.

[7] "MIT-BIH Database," Beth Israel Hospital, Biomed. Eng., Division KB-26,, Boston, MA.

[8] C. L. Nascimento Jr., "Artificial Neural Networks in Control and Optimization," Ph.D. dissertation, Control Systems Centre, Univ. Manchester Inst. Sci. Technol. (UMIST), U.K., 1994.

[9] L. Nordgren, "A new method to detect P waves in ECG signals," *Acta Soc. Med. Ups.*, vol. 74, no. 5–6, pp. 264–268, 1969.

[10] M. Okada, "A digital filter for the QRS complex detection," *IEEE Trans. Biomed. Engr.*, vol. BME-26, no. 12, pp. 700–703, Dec. 1979.

[11] T. Pike and R. A. Mustard, "Automated recognition of corrupted arterial waveforms using neural network techniques," *Comput. Biol. Med.*, vol. 22, no. 3, pp. 173–179, 1992.

[12] E. Soria-Olivas, M. Martinez-Sober, J. Calpe-maravilla, J. F. Guerrero-Martinez, J. Chorro-Gasco, and J. Espi-Lopez, "Application of adaptive signal processing for determining the limits of P and T waves in an ECG," *IEEE Trans. Biomed. Eng.*, vol. BME 45, pp. 1077–1080, Aug. 1998.

[13] T. Stamkopoulos, K. Diamantaras, N. Maglaveras, and M. Strintzis, "ECG analysis using nonlinear PCA neural networks for ischemia detection," *IEEE Trans. Signal Processing*, vol. 46, no. 11, pp. 3058–3067, Nov. 1998.

[14] G. Vijaya, V. Kumar, and H. K. Verma, "ANN-based QRS-complex analysis of ECG," *J. Med. Eng. Technol.*, vol. 22, no. 4, pp. 160–167, July 1998.

[15] Z. Yang, L. Li, and J. Ling, "Approach to recognition of ECG P waves based on approximating functions," *J. Biomed. Eng.*, vol. 15, no. 2, pp. 120–122, Jun. 1998.

[16] J. M. Zurada, *Introduction to Artificial Neural Systems*. Boston, MA: PWS, 1992.

The Generalization Error of the Symmetric and Scaled Support Vector Machines

Jianfeng Feng and Peter Williams

***Abstract*—It is generally believed that the support vector machine (SVM) optimizes the generalization error and outperforms other learning machines. We show analytically, by concrete examples in the one dimensional case, that the SVM does improve the mean and standard deviation of the generalization error by a constant factor, compared to the worst learning machine. Our approach is in terms of extreme value theory and both the mean and variance of the generalization error are calculated exactly for all cases considered. We propose a new version of the SVM (scaled SVM) which can further reduce the mean of the generalization error of the SVM.**

***Index Terms*—Generalization error, scaled SVM, SVM (SVM).**

Manuscript received November 20, 2000; revised March 29, 2001.

The authors are with the School of Cognitive and Computing Sciences, University of Sussex, Falmer, Brighton BN1 9QH, U.K. (e-mail: jianfeng@cogs.susx.ac.uk).

Publisher Item Identifier S 1045-9227(01)05527-8.

I. Introduction

Multilayer perceptrons, radial basis function (RBF) networks and support vector machines (SVMs) are three approaches widely used in pattern recognition. Compared to multilayer perceptrons and radial-basis function networks, the SVM optimizes its margin of separation and ensures the uniqueness of the final result. It seems that SVMs have become established as a powerful technique for solving a variety of classification, regression and density estimation tasks [1]. In practical applications, it is also recently reported that the SVM outperforms conventional learning algorithms [2].

How much does the SVM improve a machine's generalization capability? A number of authors have carried out a detailed analysis on the performance of the SVM [1], [3], [4]. Nevertheless, the exact behavior of the SVM on generalization remains elusive; all results obtained to date are upper bounds of the mean of the generalization error (see next section for a definition).

Here we propose a novel approach in terms of extreme value theory [5]–[7] to calculate the generalization error of an SVM exactly. Although we confine ourselves to the case of one dimension, the conclusions obtained are illuminating. First the mean and variance (or distribution) of the generalization error are calculated exactly. In the literature only upper bounds of the mean are estimated. Second, our approach enables us to go a step further in comparing different learning algorithms. We assert that the SVM improves both the mean and the variance of the generalization error by a constant factor. Third, we propose a new version of the SVM, called the *scaled* SVM, which can further reduce the mean of the generalization error. The basic idea of the scaled SVM is to employ not only the support vectors but also the means of the classes. The potential advantages are clear. The essence of the SVM is to rely only on the set of samples which take extreme values, the so-called support vectors. From the statistics of extreme values, we know that the disadvantage of such an approach is that the information contained in most samples (not extreme values) is lost, so that such an approach is bound to be less efficient than one that takes into account the lost information.

II. Models

The models we are going to consider are the SVM and the worst learning machine. For the former, the basic assumption is that learning depends only on the support vectors, in accordance with maximization of the separation margin. For the latter, we only assume that, after learning, the machine is able to recognize the learned samples correctly. Unlike most approaches in the literature, where learning machines with high dimensional inputs are considered, here we consider only the one-dimensional (1-D) case for the following reasons. First, in the 1-D case, we can easily carry out a rigorous calculation of the mean and variance of the generalization error. Second, we can fully understand why and how the SVM outperforms the worst learning machine and gain insights into how to further improve the generalization capability of a learning machine.

Let us first introduce the model used here. Suppose that the correct separating function, or target hyperplane, is sign(ξ) as shown in Fig. 1. Suppose that we observe t positive examples $x(1), \ldots, x(t) > 0$ and t negative examples $y(1), \ldots, y(t) < 0$ and let us write

$$x(tt) = \min\{x(i),\ i = 1, \ldots, t\}$$

for the minimum of the positive examples and

$$y(tt) = \max\{y(i),\ i = 1, \ldots, t\}$$

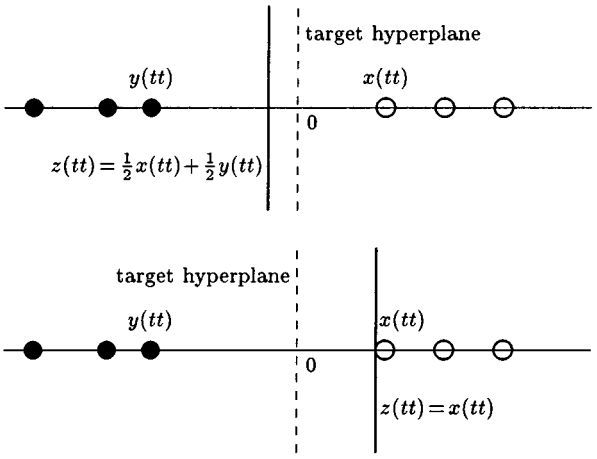

Fig. 1. Schematic representation of the SVM (above) and the worst learning machine (below). The task is to separate the disks (filled) from the circles (hollow). The true separation is assumed to be given by the dashed vertical line. After learning t examples, the separating hyperplane for the SVM is at $z(tt) = (1/2)x(tt) + (1/2)y(tt)$ (above) while the separating hyperplane for the worst learning machine is at $z(tt) = x(tt)$ (below). The error region, in either case, is the region between the dashed line and the solid line.

for the maximum of the negative examples. This case is separable so that the SVM will use the threshold

$$z(tt) = \tfrac{1}{2}\, x(tt) + \tfrac{1}{2}\, y(tt) \tag{1}$$

for classifying future cases, as shown in the upper part of Fig. 1. A newly observed ξ will be said to belong to the x or y populations depending on whether $\xi > z(tt)$ or $\xi < z(tt)$. Note that $z(tt)$ is a random variable.

Any threshold value $z(tt)$ lying between $y(tt)$ and $x(tt)$ will correctly classify the data. We call the extreme case

$$z(tt) = x(tt) \tag{2}$$

the *worst learning machine*, where we have chosen the upper endpoint for definiteness, as shown in the lower part of Fig. 1.

A. Generalization Error

We define the *generalization error* to be the probability of misclassification. Suppose that ξ is observed. An error occurs if either

1) $z(tt)$ and ξ are both positive, but ξ is less than $z(tt)$, or
2) $z(tt)$ and ξ are both negative, but ξ is greater than $z(tt)$.

The generalization error $\epsilon(t)$ is therefore a random variable

$$\epsilon(t) = P(0 < \xi < z(tt))I_{\{z(tt)>0\}} + P(z(tt) < \xi < 0) \cdot I_{\{z(tt)<0\}} \tag{3}$$

where I_A is the indicator function of the event A, in other words I_A has the value 1 if A occurs and 0 otherwise. The meaning of (3) is that if $z(tt) > 0$, the value of $\epsilon(t)$ is the probability that ξ lies between 0 and $z(tt)$; alternatively, if $z(tt) < 0$, the value of $\epsilon(t)$ is the probability that ξ lies between $z(tt)$ and 0. In brief, the generalization error $\epsilon(t)$ is the probability of error on the next case, after observing the data $x(i)$ and $y(i)$ for $i = 1, \ldots, t$, and assuming that the next case is classified using the threshold $z(tt)$.

The generalization error $\epsilon(t)$ is a random variable depending on ξ and $z(tt)$, where $z(tt)$ itself depends on the values of the $x(i)$ and $y(i)$. In the literature the *expectation* of $\epsilon(t)$ is called the generalization

error. Here we are able to calculate not only the mean of $\epsilon(t)$, but also its variance etc., so we prefer to call the random variable $\epsilon(t)$ itself the generalization error.

III. Symmetric Cases

Assume first that the $x(i)$ and $-y(i)$ are independently identically distributed (i.i.d.). This means that the distributions of $x(i)$ and $y(i)$ are antisymmetric about the origin. We assume that ξ has probability 1/2 of having the same distribution as an $x(i)$ and probability 1/2 of having the same distribution as a $y(i)$. In other words ξ has an equal chance of belonging to the two classes.

If the threshold $z(tt)$ is defined by the SVM formula given by (1), then both $z(tt)$ and ξ have symmetric distributions about the origin. The two probabilities in the generalization error (3) are therefore the same, so that $\epsilon(t)$ can be written as

$$\epsilon(t) = \tfrac{1}{2}\, P(|\xi| < |z(tt)|). \tag{4}$$

The probability of error must be less than one-half since there can be no error if $z(tt)$ is negative and ξ has a positive distribution, or if $z(tt)$ is positive and ξ has a negative distribution. We now establish results for the distribution of $\epsilon(t)$ in various cases.

A. Uniform Distribution

Suppose that each $x(i) \sim U(0,\, 1)$ is uniformly distributed on $[0,\, 1]$ for $i = 1,\, \ldots,\, t$ and, similarly, that each $y(i) \sim U(-1,\, 0)$ is uniformly distributed on $[-1,\, 0]$. Then, by assumption, ξ is $U(-1,\, 1)$ so that, in view of (4) we have

$$\epsilon(t) = \tfrac{1}{2}\, |z(tt)|. \tag{5}$$

To calculate the mean and variance of $\epsilon(t)$ we first introduce a lemma.[1]

Lemma 1: Suppose that $x(i) \sim U(0,\, 1)$ are identically and independently distributed for $i = 1,\, \ldots,\, t$. When $t \to \infty$ we have

$$P\left(x(tt) \geq \frac{x}{t}\right) = \exp(-x) \qquad (x > 0). \tag{6}$$

In other words, the distribution density of $x(tt)$ is $t\exp(-tx)$.

Proof: From example 1.7.9 in [5] we know that $P(\eta(tt) \leq 1 - x/t) = \exp(-x)$ for $\eta(tt)$ representing the largest maximum of $x(i)$. Then equation (6) is a simple consequence of the symmetry between 1 and 0 of the uniform distribution. ∎

Lemma 1 tells us the asymptotic distribution of $x(tt)$ when t is sufficiently large. In fact this extreme value distribution is applicable to a wide class of initial distributions [5] so that our approximation is less restrictive than might appear. For a given random sequence $x(i)$ we could calculate its exact distribution rather than its asymptotic distribution, which would provide further information about the behavior in small samples. Nonetheless, from here on in this section we shall assume that both $x(i)$ and $y(i)$ are uniformly distributed, which implies that the distribution of ξ is also (piecewise) uniform, and we shall use the approximation given by Lemma 1 for the distributions of the extremes.

According to (5) the generalization error is $|z(tt)|/2$. Now

$$|z(tt)| = z(tt) I_{\{z(tt)>0\}} - z(tt) I_{\{z(tt)<0\}}$$

so that, by symmetry, and using $\langle\cdot\rangle$ to denote the expected value, we have

$$\begin{aligned}\langle |z(tt)| \rangle &= 2\left\langle z(tt) I_{\{z(tt)>0\}} \right\rangle \\ &= \left\langle x(tt) I_{\{x(tt)+y(tt)>0\}} \right\rangle \\ &\quad + \left\langle y(tt) I_{\{x(tt)+y(tt)>0\}} \right\rangle. \end{aligned} \tag{7}$$

We can evaluate the two terms in (7) using Lemma 1 as follows. The first term is

$$\begin{aligned}&\left\langle x(tt) I_{\{x(tt)+y(tt)>0\}} \right\rangle \\ &\quad = \left\langle x(tt) \int_{-x(tt)}^{0} t\exp(ty)\, dy \right\rangle \\ &\quad = \langle x(tt)(1 - \exp(-tx(tt))) \rangle \\ &\quad = \int_0^\infty x(1 - \exp(-tx))t\exp(-tx))\, dx \\ &\quad = \frac{1}{t} - \frac{1}{4t} = \frac{3}{4t}. \end{aligned}$$

The second term is

$$\begin{aligned}&\left\langle y(tt) I_{\{x(tt)+y(tt)>0\}} \right\rangle \\ &\quad = \left\langle y(tt) \int_{-y(tt)}^{\infty} t\exp(-tx)\, dx \right\rangle \\ &\quad = \langle y(tt)\exp(ty(tt)) \rangle \\ &\quad = \int_{-\infty}^{0} y\exp(ty)t\exp(ty)\, dy \\ &\quad = -\frac{1}{4t}. \end{aligned}$$

Hence $\langle |z(tt)| \rangle = 1/2t$. Since $\epsilon(t) = |z(tt)|/2$, we have the following theorem.

Theorem 1: The mean of the generalization error of the SVM is given by

$$\langle \epsilon(t) \rangle = \frac{1}{4t}.$$

Although the proof of Theorem 1 is straightforward, it is very interesting to see the implications of its conclusion. In the literature, different upper bounds for the mean of the generalization error of the SVM have been found (see, for example, [3]). However, it seems that the result of Theorem 1 is the first derivation of the exact value of the mean.

It is generally believed that the generalization error of the SVM is improved relative to other conventional learning rules. By how much is it improved? We answer in the following theorem.

Theorem 2: For the worst learning machine, the mean of the generalization error is given by

$$\langle \epsilon(t) \rangle = \frac{1}{2t}.$$

Proof: For the worst learning machine we have $z(tt) = x(tt)$ so that the second term in (3) vanishes. In that case

$$\epsilon(t) = P(0 < \xi < x(tt)) = x(tt)/2$$

assuming ξ is $U(-1,\, 1)$. The result then follows from Lemma 1. Note that the same result would follow if we chose $z(tt) = y(tt)$ as the worst learning machine, or if we chose either $x(tt)$ or $y(tt)$ at random. ∎

It is well known in the literature that the mean of the generalization error of a learning machine decays at a rate of $O(1/t)$ independently of the distribution of input samples. The mean of the generalization error of both the SVM and the worst learning machine is of order $1/t$ as we should expect. The illuminating fact here is that the SVM improves the mean of the generalization error by a factor of two compared with the worst learning machine. We should emphasize that the conclusion in Theorem 2 is independent of distributions, i.e., the generalization error of the worst learning machine is universally $1/2t$ (see Lemma 3 in [7] for a proof). Nevertheless, for the SVM, the conclusion in Theorem

[1] In the following, we use the convention that all terms of order $O(\exp(-t))$ in an equality are omitted.

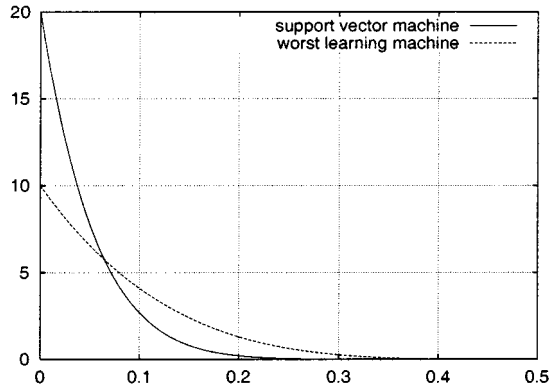

Fig. 2. Figure showing the exact distribution of the generalization error $\epsilon(t)$ for $t = 5$. The distribution is concentrated on the interval $0 \leq x \leq 0.5$. The distribution for the SVM has density $4t(1-2x)^{2t-1}$ assuming exponential distributions for the underlying variables. The distribution for the worst learning machine has density $2t(1-2x)^{t-1}$.

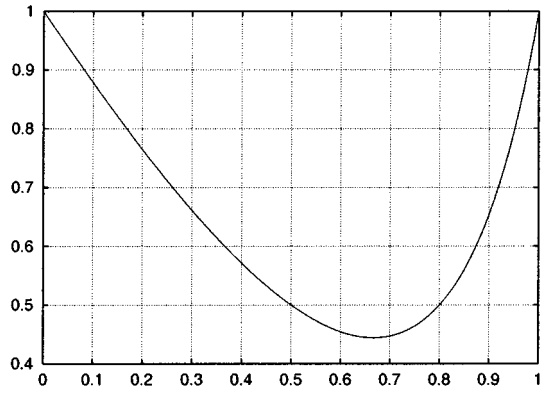

Fig. 3. Graph of $2t\langle\epsilon_\lambda(t)\rangle$ as a function of λ in the case $\alpha = 0.2$ and $\beta = 0.8$.

1 is obtained using the assumption of a uniform distribution of input samples. For any given distribution, we could calculate its mean generalization error as developed in Theorem 1. For example, the exact mean generalization error for the symmetric uniform case is in fact $t/2(2t+1)(t+1) \approx 1/4t$ and for the exponential distribution mentioned below it is $1/2(2t+1) \approx 1/4t$. The key and most challenging question is whether this conclusion is universal, i.e., independent of input distribution, or not. A detailed analysis is outside the scope of the present letter and we will report it in [8].

The generalization error of the SVM is often expressed in terms of the separation margin. We can do so here as well. Denote the separation margin by $d = x(tt) - y(tt)$.

Theorem 4: The mean of the generalization error of the SVM is

$$\langle\epsilon(t)\rangle = \frac{\langle d\rangle}{8}.$$

Proof: Since $\langle x(tt)\rangle = \langle -y(tt)\rangle = 1/t$, the conclusion follows from Theorem 1. ■

B. Variance

So far we have shown how the SVM improves performance in terms of the mean of the generalization error. How does the *variance* of the generalization error of the SVM compare with conventional learning rules? We have the following results.

Theorem 4: For the worst learning machine the variance of the generalization error $\epsilon(t)$ is

$$\text{var}(\epsilon(t)) = \frac{1}{4t^2}.$$

For the SVM the variance is

$$\text{var}(\epsilon(t)) = \frac{1}{16t^2}.$$

Proof: From the proof of Theorem 2, we know that $\epsilon(t) = x(tt)/2$ for the worst learning machine, so that the result follows from Lemma 1. For the SVM we have $\epsilon(t) = |x(tt) + y(tt)|/4$. Now

$$\begin{aligned}
&\langle |x(tt) + y(tt)|^2 \rangle \\
&\quad = \langle x(tt)^2 \rangle + 2\langle x(tt)\rangle\langle y(tt)\rangle + \langle y(tt)^2\rangle \\
&\quad = \text{var}(x(tt)) + \text{var}(y(tt)) \\
&\qquad + (\langle x(tt)\rangle + \langle y(tt)\rangle)^2 \\
&\quad = \text{var}(x(tt)) + \text{var}(y(tt)) = 2/t^2
\end{aligned}$$

where we have used the symmetry of the distributions of $x(tt)$ and $-y(tt)$ to set $\langle x(tt)\rangle + \langle y(tt)\rangle = 0$, and Lemma 1 to obtain the variances. Hence $\langle\epsilon(t)^2\rangle = 1/8t^2$. Since the variance is $\langle\epsilon(t)^2\rangle - \langle\epsilon(t)\rangle^2$ and $\langle\epsilon(t)\rangle = 1/4t$, the result follows. ■

Theorem 4 implies that the SVM also improves the standard deviation of the generalization error by a factor of two compared with the worst leaning machine. It seems that results on $\text{var}(\epsilon(t))$ have not previously been reported in the literature. We could go further and calculate the exact distribution of the generalization error. For example, Fig. 2 illustrates the distribution of the generalization error for the SVM and the worst learning machine. For the SVM, suppose that the $x(i)$ and $y(i)$ are positively and negatively exponentially distributed, so that ξ has a Laplace distribution. Then it can be shown that $2\epsilon(t)$ has a power-function distribution with density $2t(1-x)^{2t-1}$ whereas, for the worst learning machine, $2\epsilon(t)$ always has a power-function distribution with density $t(1-x)^{t-1}$. Fig. 2 shows the two distributions for $\epsilon(t)$ in the case $t = 5$.

IV. Nonsymmetric Cases

In the previous section we considered the SVM with symmetric input distributions. Certainly we do not expect $x(i)$ and $-y(i)$ to be identically distributed in practical problems. In this section we therefore assume that each $x(i) \sim U(0, a)$ and each $y(i) \sim U(-b, 0)$ where $a, b > 0$. According to Lemma 1, the limiting distributions of $x(tt)$ and $y(tt)$ now have densities $(t/a)\exp(-tx/a)$ and $(t/b)\exp(ty/b)$, respectively. Correspondingly the distribution of ξ has constant density $1/2a$ on the interval $(0, a)$ and $1/2b$ on the interval $(-b, 0)$.

In the nonsymmetric case it is not obvious where the optimal separating hyperplane should lie. We therefore consider the general case of a threshold

$$z_\lambda(tt) = \lambda x(tt) + \mu y(tt) \qquad (\lambda + \mu = 1). \tag{8}$$

Since this always lies between $y(tt)$ and $x(tt)$, any such threshold will correctly classify the data. The worst learning machines correspond to $\lambda = 0$ and $\lambda = 1$ and the SVM corresponds to $\lambda = 1/2$.

The generalization error (3) is now given by

$$\begin{aligned}
\epsilon_\lambda(t) = &\frac{1}{2a}(\lambda x(tt) + \mu y(tt)) I_{\{\lambda x(tt)+\mu y(tt)>0\}} \\
&- \frac{1}{2b}(\lambda x(tt) + \mu y(tt)) I_{\{\lambda x(tt)+\mu y(tt)<0\}}.
\end{aligned} \tag{9}$$

We can evaluate the expectations of the four components on the right to obtain

$$\left\langle x(tt) I_{\{\lambda x(tt)+\mu y(tt)>0\}} \right\rangle = \frac{a}{t}\left\{1 - \left(\frac{\mu b}{\lambda a + \mu b}\right)^2\right\}$$

$$\left\langle y(tt) I_{\{\lambda x(tt)+\mu y(tt)>0\}} \right\rangle = -\frac{b}{t}\left(\frac{\lambda a}{\lambda a+\mu b}\right)^2$$

$$\left\langle x(tt) I_{\{\lambda x(tt)+\mu y(tt)<0\}} \right\rangle = \frac{a}{t}\left(\frac{\mu b}{\lambda a+\mu b}\right)^2$$

$$\left\langle y(tt) I_{\{\lambda x(tt)+\mu y(tt)<0\}} \right\rangle = -\frac{b}{t}\left\{1-\left(\frac{\lambda a}{\lambda a+\mu b}\right)^2\right\}.$$

Using these expressions with (9) and simplifying we have

$$\langle \epsilon_\lambda(t) \rangle = \frac{1}{2t}\left\{\frac{\lambda^2\alpha+\mu^2\beta}{\lambda\alpha+\mu\beta}\right\} \tag{10}$$

where $\alpha = a/(a+b)$ and $\beta = b/(a+b)$, so that $\alpha+\beta = \lambda+\mu = 1$. Fig. 3 shows the expression in braces, namely $2t\langle\epsilon_\lambda(t)\rangle$, as a function of λ for $\alpha = 0.2$ and $\beta = 0.8$ (for example, $a = 1$, $b = 4$).

We make the following comments, the first three of which follow directly from inspection of (10).

1) For $\lambda = 0$ or $\lambda = 1$ we have the worst learning machine with
$$\langle \epsilon_0(t) \rangle = \langle \epsilon_1(t) \rangle = \frac{1}{2t}.$$
As mentioned before, this is a universal result and hence independent of the relative scaling of the input distributions.
2) For $\lambda = 1/2$ we have the conventional SVM with
$$\langle \epsilon_{1/2}(t) \rangle = \frac{1}{4t}.$$
It is interesting that this result, which was proved in Theorem 1 for the symmetric case, is independent of the scaling of the input distributions. The same independence applies to the variance. We prove a general result independent of input distributions in [8].
3) For $\lambda = \beta$, where the interval between $y(tt)$ and $x(tt)$ is now divided in the inverse ratio of the two scales, we have the same value for the expected generalization error as for the symmetric SVM, namely
$$\langle \epsilon_\beta(t) \rangle = \frac{1}{4t}$$
and again this is independent of the scaling of the input distributions. The variance, however, is increased.[2]
4) The minimum of (10) in fact occurs at
$$\lambda^* = \frac{\sqrt{\beta}}{\sqrt{\alpha}+\sqrt{\beta}} \tag{11}$$
and for $\lambda = \lambda^*$ we have
$$\langle \epsilon_{\lambda^*}(t) \rangle = \frac{1}{4t}\left\{1-\left(\frac{\sqrt{\alpha}-\sqrt{\beta}}{\sqrt{\alpha}+\sqrt{\beta}}\right)^2\right\} \tag{12}$$
which is less than $1/4t$. For example, in the case of Fig. 3 where $\alpha = 1/5$ and $\beta = 4/5$, we have $\lambda^* = 2/3$ and $\langle \epsilon_{\lambda^*}(t) \rangle = 2/9t$. In this case the improvement over the symmetric SVM is more than 10%. The variance is also decreased. At $\lambda = \lambda^*$ we have
$$\mathrm{var}(\epsilon_{\lambda^*}(t)) = \frac{1}{16t^2}\left\{1-\left(\frac{\sqrt{\alpha}-\sqrt{\beta}}{\sqrt{\alpha}+\sqrt{\beta}}\right)^4\right\} \tag{13}$$
which is less than $1/16t^2$. Note that both the mean and variance tend to zero as α or β tend to 0 or 1.

We call the machine corresponding to $\lambda = \lambda^*$ the *scaled* SVM.

A. Implementation of the Scaled SVM

Implementation of the scaled SVM requires an estimate of the ratio α/β of the scales of the two input distributions. A simple implementation is as follows. Assume that A_1 and A_2 are the data to be learned (see Fig. 4).

[2]For $\lambda = \beta$ the variance is $[1+(2\beta-1)^2]/16t^2$ which is greater than $1/16t^2$ unless $\beta = 1/2$.

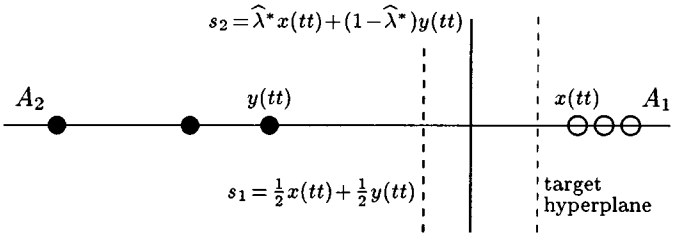

Fig. 4. The scaled SVM s_2 (thick solid line) can further improve the mean of the generalization error of the SVM s_1 (thick dashed line). s_2 is based on an estimator $\widehat{\lambda}^*$ of the quantity λ^* defined by equation (11).

1) Use the usual SVM algorithm to obtain the separating hyperplane s_1.
2) Calculate the mean of the distances from s_1 to the points in A_1, and the mean of the distances from s_1 to the points in A_2. Denote these distances by d_1 and d_2, respectively.
3) In parallel with the hyperplane s_1, find a new hyperplane s_2 so that
$$\frac{c_1}{c_2} = \sqrt{\frac{d_1}{d_2}} \tag{14}$$
where c_1 and c_2 are the distances from s_2 to the nearest points in A_1 and A_2, respectively. We call s_2 the separating hyperplane of the scaled SVM.

Under the assumptions described in the previous section, d_1/d_2 is an asymptotically unbiased estimator of α/β. The algorithm therefore implements the machine based on $\lambda = \lambda^*$ for large samples.[3]

The fact that the scaled SVM s_2 improves the mean of the generalization error of s_1 is easily understood. The SVM only uses information contained in the support vectors, whereas the scaled SVM uses information from the whole data set, since first moments of the two classes are also taken into account.

V. Discussion

We have presented a novel approach to the calculation of the exact mean and variance of the generalization error of the SVM and the worst learning machine. Estimation of upper bounds for the SVM is currently a very active topic. Our results show, for the first time, how much the SVM improves the generalization error, compared with other learning algorithms. Although we have considered a very simple case here, our results may be used as a criterion to check the tightness of estimated upper bounds of general cases (see [3], [4], and references there).

Extreme value theory is somewhat similar to the central limit theorem; it is a powerful and universal theory almost independent of the sample distributions (compare [5]–[7]). We hope the techniques introduced here may help to clarify some issues related to the SVM. Some issues we intend to pursue are the following. 1) We have only considered here the case of one dimension. It will be interesting to consider the models of higher dimension. We shall report results elsewhere [8]. 2) Extreme values are more sensitive to perturbations than other statistical quantities such as the mean or median of samples. It will be interesting to study how the SVM depends on perturbations.

References

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[3]Note that the bias for smaller samples is toward the standard mid-point solution, and it is in this region, between $1/2$ and λ^* that the minimum variance solution in fact lies.

[2] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data using support vector machines," *Proc. Nat. Acad. Sci.*, vol. 97, no. 1, pp. 262–267, 1999.
[3] V. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machines," *Neural Comput.*, vol. 12, no. 9, pp. 2013–2036, 2000.
[4] R. Dietrich, M. Opper, and H. Sompolinsky, "Statistical mechanics of support vector networks," *Phys. Rev. Lett.*, vol. 82, no. 14, pp. 2975–2978, 1999.
[5] M. R. Leadbetter, G. Lindgren, and H. Rootzén, *Extremes and Related Properties of Random Sequences and Processes*. New York: Springer-Verlag, 1983.
[6] J. Feng, "Behaviors of spike output jitter in the integrate-and-fire model," *Phys. Rev. Lett.*, vol. 79, no. 22, pp. 4505–4508, 1997.
[7] ——, "Generalization errors of the simple perceptron," *J. Phys. A*, vol. 31, no. 17, pp. 4037–4048, 1998.
[8] J. Feng and P. Williams, "Support Vector Machines: A Theoretical and Numerical Study,", 2001, submitted for publication.

Relaxation of the Stability Condition of the Complex-Valued Neural Networks

Donq Liang Lee

***Abstract*—Jankowski *et al.* have proposed a complex-valued neural network (CVNN) that is capable of storing and recalling gray-scale images. However, the weight matrix of the CVNN must be Hermitian with nonnegative diagonal entries in order to preserve the stability of the network. The Hermitian assumption poses difficulties in both physical realizations and practical applications of the networks. In this letter, a new stability condition is derived. The obtained result not only permits a little relaxation on the Hermitian assumption of the connection matrix, but also generalizes some existing results.**

***Index Terms*—Asynchronous update mode, complex-valued neural networks, energy function.**

I. Introduction

Conventional neural networks are usually based on two-state neurons, i.e., the states of the networks are usually bipolar (1 and -1) or binary (1 and 0). Although such representations are widely used in engineering applications for their simplicity, multivalued representation [4] is a much relevant and direct approximation to real-world data. In [1] the authors proposed a complex-valued neural network (CVNN) which is capable of storing and recalling gray-scale images. The CVNN is composed of fully connected multistate complex-valued neurons and the information representation is based on amplitude and phase coding. It can be referred to as a modified Hopfield network [2], [3] having complex-signum activation functions and complex weighting connections. Since the weight (connection) matrix is constructed by the generalized Hebb rule, the capacity of original CVNNs [1] is low. By using the gradient descent technique, an improved learning rule has been proposed for CVNNs [5]. However, all previous studies on CVNNs assumed that the weight matrix is Hermitian with nonnegative diagonal entries. The Hermitian assumption poses difficulties in both physical realizations and practical applications of the networks [7], [8]. First, it is almost impossible to implement the hardware network precisely conserving symmetry or antisymmetric connections because this requires that two physical quantites (such as resistances or the gains of amplifiers) be *exactly* equal [9]. Second, asymmetric weight matrices will allow a wider variety of dynamics behaviors in neural networks, such as trajectory attractors and cycles of finite lengths [10]. These raise the importances of relaxing the Hermitian assumption of the CVNNs.

In this letter, the concept of CVNNs is briefly reviewed. Then, a new condition ensuring convergence of the CVNN in asynchronous mode is derived. The validity of the obtained result is demonstrated by an example.

II. Background

CVNN is an autoassociative memory that stores complex-valued prototype vectors X^k, $k = 1, \ldots, m$, where $X^k = (x_1^k, x_2^k, \ldots, x_N^k)^T$ and m is the number of the prototype vectors. The components $x_i^k s$ are all quantization values defined by

$$x_i^k \in \exp[i2\pi v/K]_{v=0}^{K-1} \qquad i = 1, \ldots, N. \tag{1}$$

The resolution factor K divides the complex unit circle into K quantization levels so that $|x_i^k| = 1 \ \forall i, k$. The m prototype vectors are stored in the weight matrix according to the generalized Hebb rule [1]

$$s_{ij} = \frac{1}{N}\sum_{k=1}^{m} x_i^k \overline{x}_j^k \qquad i, j = 1, \ldots, N \tag{2}$$

where $\overline{x}_j^k$ denotes the complex conjugate of x_j^k. Let $X \in \mathbb{C}^N$ denotes the state of the CVNN. The asynchronous recalling process of the CVNN is determined by the following equation:

$$x_i' = \left\{\sum_{j=1}^{N} s_{ij}x_j\right\} \tag{3}$$

in which x_i is the ith component of X; x_i' denotes the next state of x_i. Moreover, $\phi(\cdot)$ is a complex-signum function

$$\phi(Z) = \begin{cases} \exp(i0) & 0 \le \arg\left[Ze^{i(\theta_0/2)}\right] < \theta_0 \\ \exp\left(i\,\dfrac{2\pi}{K}\right) & \theta_0 \le \arg\left[Ze^{i(\theta_0/2)}\right] < 2\theta_0 \\ \vdots & \\ \exp\left[i\,\dfrac{2\pi(K-1)}{K}\right] & (K-1)\theta_0 \le \arg\left[Ze^{i(\theta_0/2)}\right] < K\theta_0 \end{cases} \tag{4}$$

where $\arg(\alpha)$ is the phase angle of α, θ_0 is a phase quantum delimited by K: $\theta_0 = 2\pi/K$. Equation (4) means that $\phi(Z)$ is the quantization value on the complex unit circle closest to Z. The process (3) is a stochastic process that starts with an initial vector X_0 presented to the network. Then neuron states are updated one at a time by following (3) with equal probabilities. In [1] the authors proved that the process (3) converges to one of the fixed points X_f in a finite number of iterations if the matrix $S = [s_{ij}]$ is Hermitian with nonnegative diagonal entries

$$s_{ii} \ge 0, \qquad \forall i. \tag{5}$$

Obviously, a stored vector can be recalled only if it is a fixed point. If this is true the CVNN can be referred to as a multivalued associative memory. Since the connection matrix is constructed by the generalized

Manuscript received February 12, 2001; revised June 10, 2001. This work was supported by the National Science Council, R.O.C., under Grant NSC 90–2213–E-233–001.

The author is with the Department of Electronics Engineering, Ta-Hwa Institute of Technology, Chung-Lin, Hsin-Chu, Taiwan 307, R.O.C.

Publisher Item Identifier S 1045-9227(01)05528-X.