# The Effective VC Dimension of the $n$-tuple Classifier

N.P. Bradshaw

IRIDIA - ULB (CP 194/6), 50, av. F.Roosevelt, 1050-Brussels.

**Abstract.** One family of classifiers which has has considerable experimental success over the last thirty year is that of the $n$-tuple classifier and its descendents. However, the theoretical basis for such classifiers is uncertain despite attempts from time to time to place it in a statistical framework. In particular the most commonly used training algorithms do not even try to minimise recognition error on the training set. In this paper the tools of statistical learning theory are applied to the classifier in an attempt to describe the classifier's effectiveness. In particular the effective VC dimension of the classifier for various input distributions is calculated experimentally, and these results used as the basis for a discussion of the behaviour of the $n$-tuple classifier. As a side-issue an error-minimising algorithm for the $n$-tuple classifier is also proposed and briefly examined.

## 1 Introduction to the $n$-tuple Classifier

The original $n$-tuple classifier was described by Bledsoe and Browning in [3]. It is a pattern recognition system which accepts binary images and outputs a binary "yes/no" response. Modifications to the original design have included allowing the output to be one of a finite number of preset class labels [1], extending the input space to allow real-valued data [2, 9] or extending the output space to solve regression problems [6]. It has also been shown to give good performance on the Statlog data sets [7].In this paper the system considered will accept binary strings as inputs and output 1 or 0. A schematic diagram of this system is given in figure 1.

The architecture of the classifier consists of three layers: a layer of look-up tables, a layer of summing devices (one per class) and a winner-takes-all comparison. The operation of the classifier consists of three stages: a sampling stage, a look-up stage and an output stage. The principle of the classifier is that the image space may be sampled in blocks of $n$ bits known as **$n$-tuples**. For the purposes of this paper it will be assumed that the $n$-tuples are chosen uniformly at random with the condition that no two overlap (ie. no input bit belongs to more than one $n$-tuple) although it is possible to perform a similar analysis with this assumption relaxed. Each class has an associated set of $N$ look-up tables or **nodes**, the whole set being known as a **discriminator**. The nodes in each discriminator are connected to the input space in exactly the same way so that all discriminators

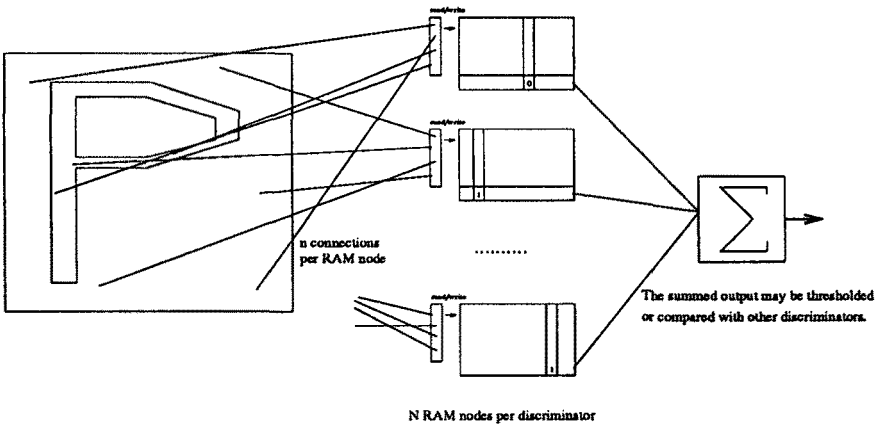


**Fig. 1.** A single n-tuple discriminator (schematically). An $n$-tuple classifier consists of as many discriminators as there are classes. Each discriminator is associated to each class. The classifier returns the class label of the discriminator with the highest output.

are identical apart from the contents of the nodes. The look up tables take binary $n$-bit strings as inputs and generate binary bits, $\{1, 0\}$, as output.

In each discriminator the (binary) output for each $n$-tuple is then read from the corresponding look-up table. The output of the discriminator is obtained by summing the outputs of each node to obtain an integer between 0 and $N$. The outputs of each discriminator are compared and the class-label associated with the highest scoring discriminator is given as the output of the whole system. For this paper we assume that all classifiers contain only two discriminators. We shall also assume that in the case where the two discriminators give identical integer outputs, the class-label **1** is output. The problem for the training algorithm is what information to put in the nodes. In this paper the SMA training algorithm, see [5], is used.

## 2 Background to Statistical Learning Theory

Statistical learning theory is the application of statistical techniques to the problem of learning from examples which can be used to derive minimal required training set sizes to guarantee a given level of generalisation with a certain confidence. These bounds are usually formulated in terms of the *VC dimension*, see for example [10]. Previous work has fixed lower and upper bounds for for the two-discriminator $n$-tuple classifier [4] as $N(2^n - 1)$ and $(\log_2 3)N(2^n - 1)$ respectively.

However, most practical work suggests that the sample sizes required by the VC dimension bounds are in fact much larger than required (see for example the results of the Statlog tests performed by Rohwer and Morciniec, [7]). In [11]

Vapnik, Levin and Le Cun define a quantity called *effective VC dimension* which is based on the learning machine together with an input distribution which yields a new sample size bound which is never greater than the original. Vapnik et al. also give an experimental method for estimating it, a method which they show works well in the case of the linear perceptron. This method is applied in this paper to the $n$-tuple classifier and shown to give plausible results in this case too. Thus the $n$-tuple classifier is analysed further and the experimental method of Vapnik et al. is further validated.

## 3 Calculating the Effective VC Dimension

Vapnik et al. suggest a definition of a VC dimension based not on all input sets but just on those with probability close to one. More formally

**Definition** The effective VC dimension of the learning machine $\mathcal{L}$ for the input distribution $P$ is the minimal VC dimension of $\mathcal{L}$ on those subsets $X^*$ of the input space $X$ whose probability measure according to $P$ is almost 1.

They show that the effective VC dimension can be estimated by measuring the maximal deviation $\xi_l$ between the errors of a trained classifier on two halves of an input sample of length $2l$. The estimation takes the form of a function (with two free parameters $a$ and $b$) denoted by $\Phi(l/h)$, or an approximation to this denoted by $\Phi_1(l/h)$ with one free parameter $d$, whose forms are given in [11]. The theoretical demonstration can be found in [11].

To maximise the error divergence with the $n$-tuple classifier, an error *minimising* algorithm for the $n$-tuple classifier is needed. The *Stochastic Mimimisation Algorithm* (SMA) for the $n$-tuple classifier was proposed in [4] as such an algorithm. The principle of the algorithm is to train as many patterns as possible by looping through the training set and in each case of misclassification changing a minimal number of output values — selected at random — so that the current pattern is trained correctly. The hope is that by making an alteration of minimal size the previously trained responses will not be disturbed and thus that a good approximation to the global minimum of the training error can be found. The loop is repeated enough times so that the minimum training error stops decreasing.

### 3.1 Necessary Assumptions

For an empirical estimate of effective VC dimension to be made by the method described the following assumptions must hold true.

- $E[\xi_l]$ does not depend on the distribution of the classes, only the patterns themselves.
- The expected deviation depends on the learning machine only through the effective VC dimension, $h$.

- $\Phi(l/h)$ or $\Phi_1(l/h)$ is a good approximation to $E[\xi_l]$ for appropriate values of the parameters $a$ and $b$.
- $a$ and $b$ are constant over a large class of related learning machines.

These assumptions were verified for the $n$-tuple classifier in [5].

### 3.2 Varying the Classifier

Values of $\xi_l$ were calculated for various values of $n$ and $N$ as well as $i$, the range of the integers stored in each RAM location. The output values were assigned with 50% probability as was experimentally justified in [5]. The $\xi_l$ are plotted in figure 2 for varying values of $N$ and $n$. To allow comparison, the plot for the basic two discriminator classifier with $n = 4, N = 50$ is included in both graphs. As a guide to interpreting these graphs it is worth noting that the further to the right a curve is, the higher its effective VC dimension.

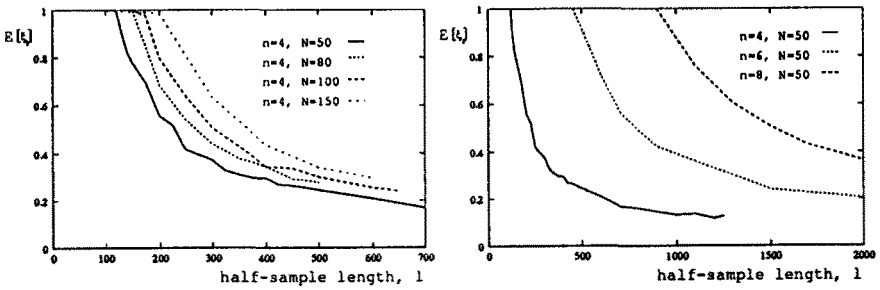


**Fig. 2.** Empirical $E[\xi_l]$ for a range of n-tuple classifiers with varying number of (a) nodes per discriminator, $N$ and (b) support per node. Uniform distribution.

To make an estimate of the effective VC dimension, we must plot $E[\xi_l]$ against $l/h$ and show that some curve $\Phi$ fits the resulting graph. If the same $\Phi$ fits all the $E[\xi_l]$ graphs then our assumption that the parameters $a$ and $b$ are independent of the learning machine will have been justified. Since in almost all cases the ranges of $l$ are such that $l/h < 5$, an approximation by $\Phi_1$ with a single free parameter $d$ is valid, see [11]. Figure 3 shows the results for a range of machines and the best fit curve of type $\Phi_1$, with $d = 0.225$. The same value of $d$ fits all settings of the parameters.

## 4 Results and Conclusions

The known VC dimension values and bounds are shown in table 1 along with the corresponding effective VC dimension values for the uniform distribution.

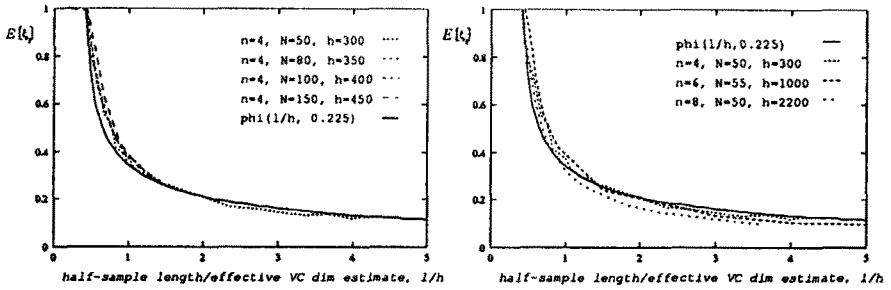


**Fig. 3.** Empirical $E[\xi_l]$ for varying (a) $N$ and (b) $n$ against estimated EVCD over l plotted against a best-fit estimate. Uniform distribution.

| n | N | VC dim min | VC dim max | EVC dim | Ratio 1 | Ratio 2 |
|---|---|---|---|---|---|---|
| 4 | 50 | 750 | 1190 | 300 | 2.5 | 4.0 |
| 4 | 100 | 1500 | 2380 | 400 | 3.8 | 6.0 |
| 4 | 150 | 2250 | 3570 | 450 | 5.1 | 7.9 |
| 6 | 50 | 3150 | 4990 | 1000 | 3.2 | 5.0 |
| 8 | 50 | 12750 | 20,200 | 2200 | 5.8 | 9.2 |

**Table 1.** Effective (uniform distribution) and actual VC dimensions for $n$-tuple classifier. Ratio 1 is the VC dimension lower bound over the effective VC dim, while Ratio 2 is the VC dimension upper bound over the effective VC dim.

This table shows at a glance how pessimistic the VC dimension results are when the patterns are drawn from a uniform distribution.
Table 1 shows clearly that the effective VC dimension of the n-tuple classifier over a uniform distribution of input patterns is significantly less than the actual VC dimension.

### 4.1 Conclusions and Future Work

The aim of the work in this paper has been two-fold. First to use the tools of learning theory to try to explain and predict the performance of the $n$-tuple classifier and second to validate the approach of Vapnik et al. to incorporating information about the input distribution into the VC bounds. Estimates of generalisation error of $n$-tuple classifiers made using the full VC dimension tend to severely over-estimate the error found in most experimental contexts. The current work shows that if the bounds are based on effective VC dimension then the predicted bounds are closer to experimental results. This was shown in [4]. Furthermore the work has validated the hypotheses of Vapnik et al. in 3.1 and given consistent results. Thus both of these goals have been substantially achieved.

Several directions of further work are suggested by this study, some in the domain of the $n$-tuple classifier and others in the domain of learning theory. The

relationship of the various training algorithms to the success of the learning process are brought sharply into focus by this work. Although the aim of the classifier is to classify patterns with minimum error, the training algorithm does not explicitly try and minimise error on the training set. Non-zero error on the training set is often referred to as "saturation" and dealt with by increasing the number of samples (ie. $N$) or the size of the $n$-tuple, thereby increasing the VC dimension of the classifier. On the other hand, the SMA, which does try to explicitly minimise the error on the training set, has been shown to give markedly worse results to the original algorithm on certain data (see [4]). It would be interesting to know how the training algorithm limits the search for an acceptable hypothesis and thus how the classifier is often able to achieve good performance despite apparent over-capacity.

In parallel to this, a practical theory of learning which takes into account more than just the machine capacity and the input distribution is required if theoretical sample size predictions are to become a useful tool for those applying learning machines to different tasks. For instance the "unluckiness" function defined by Shawe-Taylor et al. [8] is one new way of incorporating prior expectations about the data distribution into the VC dimension/sample size bounds calculations.

## References


1. I. Aleksander and T.J. Stonham. Guide to pattern recogntion using random-access memories. *Computers and Digital Techniques*, 2:29–40, 1979.
2. W.W. Bledsoe and C.L. Bisson. Improved memory matrices for the n-tuple recogntion method. In *IRE Joint Computer Conference,11*, pages 414–415, 1962.
3. W.W. Bledsoe and L. Browning. Pattern recognition and reading by machine. In *Proc. Eastern Joint Computer Conf.*, pages 232–255, 1959.
4. N.P. Bradshaw. *An Analysis of Learning in Weightless Neural Systems.* PhD thesis, Imperial College, London., 1996.
5. N.P. Bradshaw. Improving the generalisation of the n-tuple classifier with the effective VC-dimension. Technical report, IRIDIA, Universite Libre de Bruxelles, 1997.
6. A. Kolcz and N.M. Allinson. N-tuple regression network. *Neural Networks*, 9(5):855–869, 1999.
7. R. Rohwer and M. Morciniec. A theoretical and experimental account of n-tuple classifier performance. *Neural Computation*, 8:657–670, 1996.
8. J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, and M. Anthony. A framework for structural risk minimisation. In *Proceedings of the 9th Annual Conference on Computational Learning Theory*, 1996.
9. M.J. Sixsmith, G.D. Tattershall, and J.M. Rollett. Speech recognition using n-tuple techniques. *Br Telecom J*, 8(2), April 1990.
10. V. Vapnik. *The Nature of Statistical Learning Theory.* Spinger-Verlag, 1995.
11. V. Vapnik, E Levin, and Y LeCun. Measuring the VC-dimension of a learning machine. *Neural Computation*, 6:851–876, 1994.