Low Training Strength High Capacity Classifiers for Accurate Ensembles Using Walsh Coefficients

Terry Windeatt and Cemre Zor

Univ Surrey, Guildford, Surrey, Gu2 7XH
t.windeatt'surrey.ac.uk

Abstract. If a binary decision is taken for each classifier in an ensemble, training patterns may be represented as binary vectors. For a two-class supervised learning problem this leads to a partially specified Boolean function that may be analysed in terms of spectral coefficients. In this paper it is shown that a vote which is weighted by the coefficients enables a fast ensemble classifier that achieves performance close to Bayes rate. Experimental evidence shows that effective classifier performance may be achieved with one epoch of training of an MLP using Levenberg-Marquardt with 64 hidden nodes.

Keywords: Ensembles, Multilayer Perceptrons, Boolean Function, Walsh Coefficients.

1 Introduction

For an ensemble of classifiers it is often useful to think of each base classifier as being controlled by two main parameters, the *capacity* and the *training strength* of the learning algorithm [1]. The term *capacity* refers to the flexibility of the classifier boundary. By *training strength* we mean the effort that is put into training the classifier. For an MLP, the capacity is the number of hidden nodes, and training strength is the number of epochs. In this paper we consider the trade-off between these two parameters, and what combination is suitable for a weighted majority vote.

The weighted vote is computed using Walsh coefficients. If each base classifier in an ensemble is given a binary decision, and if the problem is two-class, a Boolean mapping is defined. This mapping may be analysed using Walsh spectral coefficients. First order Walsh coefficients were shown to provide a measure of class separability for selecting optimal base classifiers in [2], in which it is also shown that this does not imply optimality of the ensemble. In contrast, in [3] it was shown that second order Walsh coefficients may be used to determine optimal ensemble performance. The motivation for using Walsh coefficients in ensemble design is fully explored in [4] and [2]. For further understanding of the meaning and applications of Walsh coefficients see [5] and [6].

To understand the computation of the weighted vote, the Tumer-Ghosh model [7] for ensemble classifiers will be described. This model defines Added Classification Error as the difference between classifier error and Bayes error, and provides a framework for understanding the reduction in error due to combining.

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 701–709, 2012.

Section 2 explains the computation of the Walsh coefficients, and Section 3 discusses the relationship with the model of Added Classification Error. In Section 4, the weighted vote using Walsh coefficients is compared as the number of nodes and training epochs of MLP base classifiers are systematically varied.

2 Walsh Coefficients

Consider an ensemble framework, in which there are *N* parallel base classifiers, and X_m is the *N*-dimension vector representing the *mth* training pattern, formed from the decisions of the *N* classifiers. For a two-class supervised learning problem of μ training patterns, the target label given to each pattern X_m is denoted by $\Omega_m = \Phi(X_m)$ where $m = 1 \ldots \mu$, $\Omega_m \in \{1,-1\}$ and Φ is the unknown Boolean function that maps X_m to Ω_m. Thus the binary vector X_m represents the *mth* original training pattern

$$X_m = (X_{m1}, X_{m2}, \ldots, X_{mN}) \tag{1}$$

where $X_{mi} \in \{1,-1\}$ is a vertex in the *N*-dimensional binary hypercube. The Walsh transform of Φ is derived from the mapping T_n and defined recursively as follows

$$T_n = \begin{bmatrix} T_{n-1} & T_{n-1} \\ T_{n-1} & -T_{n-1} \end{bmatrix} T_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

The first and second order spectral coefficients s_i and s_{ij} derived from (2) are defined in [5] as

$$s_i = \sum_{m=1}^{\mu} (X_{mi} \Omega_m) \tag{3}$$

$$s_{ij} = \sum_{m=1}^{\mu} (X_{mi} \oplus X_{mj}) \Omega_m \tag{4}$$

In (3) s_i represents the correlation between $\Phi(X_m)$ and X_{mi} and $s_{ij}(i, j = 1 \ldots N, i \neq j)$ in (4) represents correlation between $\Phi(X_m)$ and $X_{mi} \oplus X_{mj}$, where $\oplus$ is logic exclusive-OR.

Realistic learning problems are ill-posed [8], and therefore Φ may be partially specified, noisy and possibly contradictory. Relationships for computing spectral coefficients for partially specified Boolean functions, are proved in [9], for which the

context is logic circuit design. The relevant ideas are presented here using different terminology, specifically minterms interpreted as patterns.

In [9], the concept of a standard trivial function Ψ is introduced. Each spectral coefficient gives a correlation value between the Boolean function Φ and Ψ. For first order coefficients, Ψ_i is the Boolean variable X_{mi} in (3) while for second order coefficients Ψ_{ij} is $X_{mi} \oplus X_{mj}$ in (4). Note in (4) $X_{mi} \oplus X_{mj} = 1$ implies pair of classifiers *i* and *j* disagree for pattern X_m and $X_{mi} \oplus X_{mj} = 0$ implies classifiers agree. For third order coefficients, Ψ_{ijk} is $X_{mi} \oplus X_{mj} \oplus X_{mk}$ and higher order follows, but in this paper we restrict ourselves to first and second order spectral coefficients.

The equations (3) and (4) require binary variables $\{1,-1\}$ but for computing coefficients it is notationally more convenient to use $\{0,1\}$. For $p, q \in \{0,1\}$ define n_{pq} to be the number of class *p* patterns of Boolean function Φ for which both Φ and Ψ have the logical value *q*. Then n_{11} is the number of class *1* patterns (true minterms in [9]) for which both Φ and Ψ that have the logical value 1. Similarly n_{00} is the number of class *0* patterns (false minterms in [9]) for which both Φ and Ψ have the logical value *0*. Corresponding definitions follow for n_{01} and n_{10}. Now define d_1 and d_0 to be the number of unspecified patterns (don't care minterms) for which Ψ has the logical value 1 and 0 respectively. It is clear that the sum of all patterns of an N-dimensional Boolean function is given by

$$n_{11} + n_{00} + n_{01} + n_{10} + d_1 + d_0 = 2^N \tag{5}$$

According to [9], all spectral coefficients s_l may be computed as

$$s_l = (n_{11} + n_{00}) - (n_{01} + n_{10}) \tag{6}$$

where l may be *i* or *ij*. Substitution of (5) into (6) gives various equivalent formulae, but the advantage of (6) is that it is not necessary to include unspecified patterns d_1, d_0 explicitly in the computation.

3 Added Classification Error Model

Figure 1 shows the two class (ω_1, ω_0) model of Added Classification Error (E darkly shaded region) according to [7], which for simplicity is restricted to one

dimension (x). The optimum (Bayes) boundary in Figure 1 is the loci of all points $\tilde{x} : P(\omega_1 \mid \tilde{x}) = P(\omega_0 \mid \tilde{x})$. The output of the classifier representing class ω_1 is given by $\hat{P}(\omega_1 \mid x) = P(\omega_1 \mid x) + \varepsilon_1(x)$ where $P, \hat{P}$ are the actual and estimated *a posteriori* probability distributions as shown in Figure 1, and $\varepsilon_1(x)$ is the difference between them. A similar equation is obtained for class ω_0 with $P(\omega_0 \mid x), \hat{P}(\omega_0 \mid x)$ and error $\varepsilon_0(x)$. In Figure 1 b is the amount that the *kth* classifier boundary (x_b) differs from the ideal Bayes boundary ($\tilde{x}$), and assuming that b is a Gaussian random variable with mean β and variance σ_b, in [7] it is shown that Added Classification Error for *kth* classifier is given by $E_k = \nabla P(\sigma_b^2 + \beta^2)$ where $\nabla P = 0.5(P'(\omega_1 \mid \tilde{x}) - P'(\omega_0 \mid \tilde{x}))p(\tilde{x})$ and P' indicates differentiation.

Figure 2 shows decision boundaries of *(i,j)th* classifiers for which it is assumed that the complexity is not sufficient to approximate the Bayes boundary, so that both classifiers under-fit. Note in Figure 2 that estimated probabilities $\hat{P}(\omega_0 \mid x)$ and $\hat{P}(\omega_1 \mid x)$ are omitted for clarity. Mutually exclusive areas under the probability distribution are labelled 1 – 8 in Figure 2, and denoting the number of patterns in area y by a_y, the contribution from classifiers i,j according to area is given in Table 1.

The model assumptions are discussed in [3], in which the expression for the difference in Added Classification Error of *ith* and *jth* classifiers $E_{ij} = E_i - E_j$ is derived

$$E_{ij} = E_i - E_j = 0.5(s_{ij} + \gamma) \tag{7}$$

where $\gamma = 1 - 2p_0$ and p_0 is the prior probability of class ω_0.

Averaging over all pairs of classifiers in (7) the mean difference in added error is given by

$$\Delta\overline{E} = \sum_{i,j,i \neq j} E_{ij} \tag{8}$$

Therefore from (7) and (8) we can approximate mean Added Error by subtracting γ and averaging over all pair-wise second order coefficients, call it S2M. In [3] it is shown that S2M is a good predictor of ensemble performance as number of epochs is increased. For the datasets in Section 4, optimal performance for majority vote occurs on average around 2-3 epochs.

The usual idea in weighted voting is to reward individual classifiers that perform accurately [10]. In this paper, a different approach is taken. For classifiers with lower training strength, it is expected that classifiers maybe unevenly spread around the optimal boundary. The idea is to give larger weight to pairs of classifiers with low

Added Error. The classifiers are chosen based on the product of first order coefficients as follows. The first order coefficients in (3) are decomposed into the contributions from the two classes $(n_{11} - n_{01})$ and $(n_{00} - n_{01})$ and the weight is proportional to their product. The weight of the *ith* classifier is given by

$$w_l = (n_{11} - n_{10})(n_{00} - n_{01}) \tag{9}$$

with negative weights in (9) set to zero. Considering Figure **1**, classifiers close to the Bayes boundary will receive larger weight, but as they move further away, weight is decreased and becomes zero as n_{11} approaches n_{10} or as n_{00} approaches n_{01}. When classes are unbalanced, (9) tends to favour classifiers on either side of the Bayes boundary, in contrast to a weighting scheme based on training error. The weighting scheme using (9) is referred to as W1P in Section 4, and shown to reduce the mean Added Error given by (8).

4 Experimental Evidence

Natural two-class benchmark problems selected from [11] and [12] are shown in Table 2.The original features are normalised to mean 0 std 1, and for datasets with missing values the scheme suggested in [11] is used. Random perturbation of the MLP base classifiers is caused by different starting weights on each run. The number of hidden nodes and training epochs of homogenous (same number of nodes and epochs) MLP base classifiers are systematically varied over 1-5 epochs and 2-64 nodes. The experiments are performed with two hundred single hidden-layer MLP base classifiers, using the Levenberg-Marquardt training algorithm with default parameters. Combining uses majority (MAJ) or weighted vote. The random train/test split is 20/80 and experiments are repeated twenty times and averaged. Note that, for each dataset the class with most patterns is assigned ω_0 to give the same sign to γ in (7).

Bias/Variance will refer to 0/1 loss function using Breiman's decomposition [13], for which Bias plus Variance plus Bayes equals the base classifier error rate. Bias is intended to capture the systematic difference with Bayes, and requires Bayes probability. Patterns are divided into two sets, the Bias set containing patterns for which the Bayes classification disagrees with the ensemble classifier and the Unbias set containing the remainder. Bias is computed using the Bias Set and Variance is computed using the Unbias Set, but both Bias and Variance are defined as the difference between the probabilities that the Bayes and base classifier predict the correct class label. The Bayes estimation is performed for 90/10 split using original features, and a Support Vector Classifier (SVC) with polynomial kernel run 100 times. The polynomial degree and regularisation constant are varied, and lowest test error is given in Table 2.

Figure 3 gives mean results over seven datasets, which clearly indicates the overall trend. Figure 3 (a) (b) shows base and ensemble (MAJ) test error rates. Figure 3 (c)-(f) shows difference between MAJ and various weighted vote schemes. Figure 3 (c) uses the first order Walsh coefficient (W1D) in (3), Figure 3 (d) is the proposed scheme (W1P) using (9), Figure 3 (e) uses the logarithmic weighting scheme used in Adaboost (ADA) [14]. Figure 3 (f) uses a trained linear perceptron (LIN) to learn the mapping. All the weighting schemes give a large improvement over MAJ at 1 epoch, the best being W1P, with a 13 percent improvement at 64 nodes. The best MAJ error occurs at 3-4 epochs, and here there is a small improvement W1P over MAJ of between 0.3 percent at 64 nodes and 1 percent at 4 nodes.

Fig. 4 shows various measures to help explain the results. Fig 4 (a) shows the mean second order coefficients (S2M), normalised by the total number of training patterns, and which is an estimate of the mean added error in (8). Figure 4 (b) is similar to (a) but shows coefficients weighted by (9) (for classifier i and j, weight is given by $(w_i + w_j)/2$). Figure 4 (c) – (f) show bias and variance for MAJ (Bias, Var) and W1P (BiasW, VarW). By comparing Figure 4 (a) and (b) the weighted coefficients (S2W) shows that weighted classifiers have reduced the Added Error. The Weighted bias (BiasW) in (d) is reduced in comparison with Bias in (c). For 64 nodes, the best weighted error rate is at 1 epoch, shown in (d), which is within 1 percent of Bayes rate. On the other hand, at 1 epoch Figure 4 (e) (f) show that weighted variance has increased, indicating that more diverse classifiers are weighted.

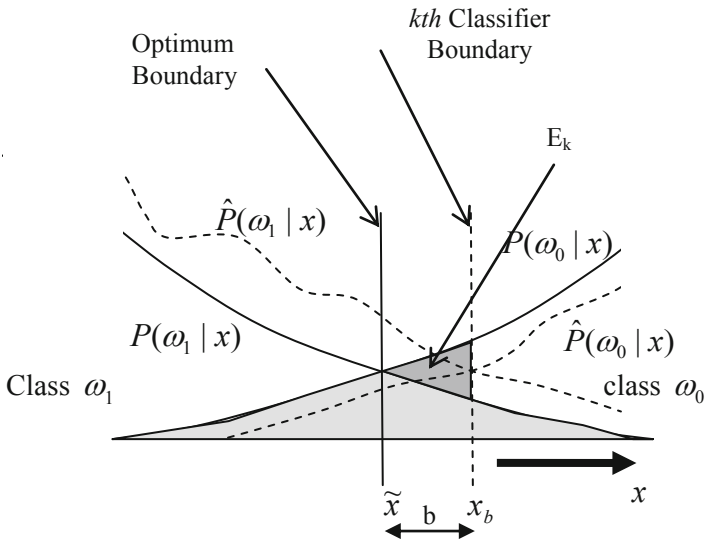

Fig. 1. Model of error region associated with *a posteriori* probabilities showing optimum (Bayes) boundary, kth classifier boundary with Added Classification Error (E_k)

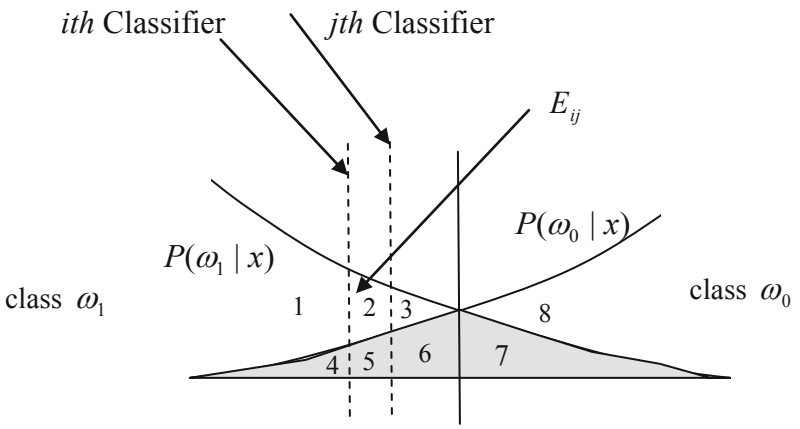

Fig. 2. Model showing pair of classifier boundaries and the difference in Added Classification Error between *ith* and *jth* classifiers E_{ij} (area 2)

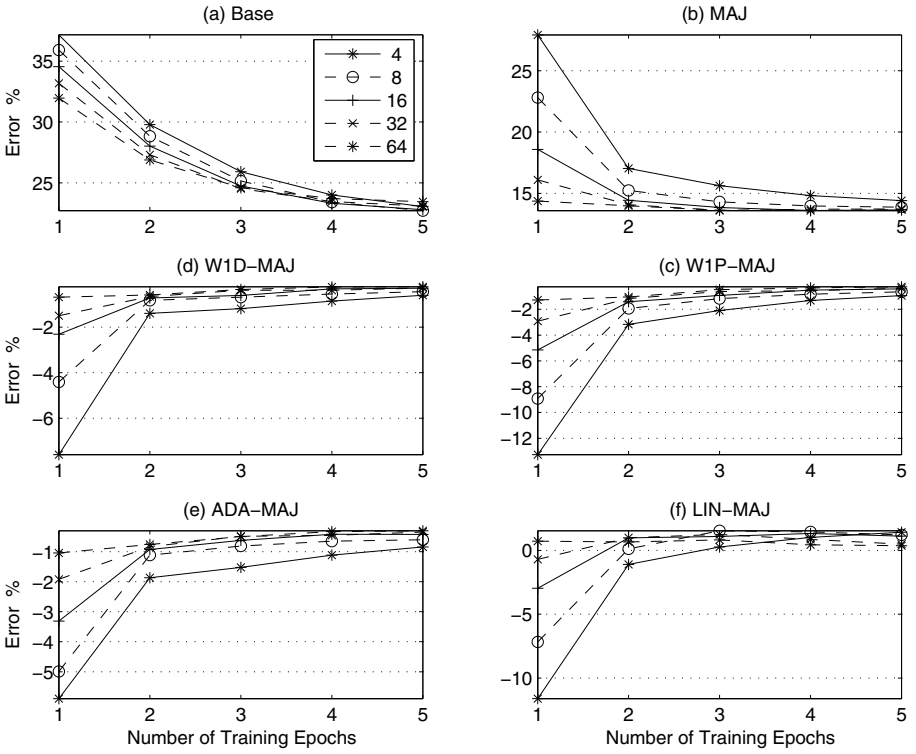

Fig. 3. Mean test errors over 2-class datasets for [4,8,16,32,64] nodes 1-5 epochs (a) Base Classifier (b) Majority Vote (c) –(f) Weighted votes with MAJ subtracted

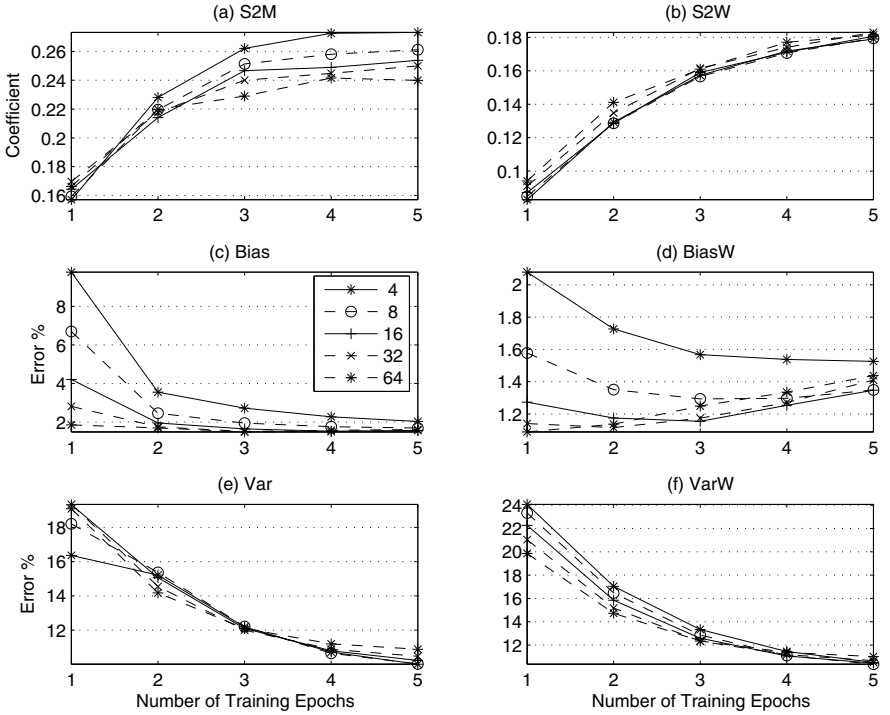

Fig. 4. (a) Mean measures over 2-class datasets for [4,8,16,32,64] nodes 1-5 epochs (a) Second order coefficients (b) Weighted Second order coefficients (c) Bias (d) Bias W1P (e) Variance (f) Variance W1P

Table 1. Areas under Distribution defined in Fig. **2**, showing corresponding number of class ω_1, ω_0 patterns (1st subscript) for which the pair of classifiers agree or disagree (2nd subscript)

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
ω_1	n_{10}	n_{11}	n_{10}	n_{10}	n_{11}	n_{10}	n_{10}	
ω_0				n_{00}	n_{01}	n_{00}	n_{00}	n_{00}

Table 2. Datasets showing # patterns, prior probability ω_0, #continuous and discrete features and estimated Bayes error

DATASET	*#pat*	p_0	*#con*	*#dis*	*%Bay*
cancer	699	.655	0	9	3.1
card	690	.555	6	9	12.8
credita	690	.555	3	11	14.1
diabetes	768	.651	8	0	22.0
heart	920	.553	5	30	16.1
ion	351	.641	31	3	6.8
vote	435	.614	0	16	2.8

5 Conclusion

For two-class supervised learning problems, the spectral representation of the mapping between binary base classifier decisions and target class has been analysed with the help of the Tumer-Ghosh model of Added Classification Error. If the majority vote is weighted by the product of the class-dependent first-order coefficients, the ensemble has error rate that is close to optimal, even with fast inaccurate base classifiers.

References

[1] Smith, R.S., Windeatt, T.: A Bias-Variance Analysis of Bootstrapped Class-Separability Weighting for ECOC Ensembles. In: Proceedings of the 22nd International Conference on Pattern Recognition, Istanbul, Turkey (August 2010)
[2] Windeatt, T.: Vote Counting Measures for Ensemble Classifiers. Pattern Recognition 36(12), 2743–2756 (2003)
[3] Windeatt, T., Zor, C.: Minimising Added Classification Error using Walsh Coefficients. IEEE Trans. Neural Networks 22(8), 1334–1339 (2011)
[4] Windeatt, T.: Accuracy/ Diversity and ensemble classifier design. IEEE Trans. Neural Networks 17, 1194–1211 (2006)
[5] Hurst, L., Miller, D.M., Muzio, J.: Spectral Techniques in Digital Logic. Academic Press (1985)
[6] Beauchamp, K.G.: Walsh Functions and their Applications. Academic Press (1975)
[7] Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science 8(3), 385–404 (1996)
[8] Tikhonov, A.N., Arsenin, V.A.: Solutions of Ill-posed Problems. Winston & Sons, Washington (1977)
[9] Falkowski, B.J., Perkowski, M.A.: Effective Computer Methods for the Calculation of Rademacher-Walsh Spectrum for Completely and Incompletely Specified Boolean Functions. IEEE Trans. on Computer-Aided Design 11(10), 1207–1226 (1992)
[10] Kuncheva, L.I.: Combining Pattern Classifiers. Wiley (2004)
[11] Prechelt, L.: Proben1: A set of neural network Benchmark Problems and Benchmarking Rules. Tech Report 21/94, Univ. Karlsruhe, Germany (September 1994)
[12] Merz, C.J., Murphy, P.M.: UCI repository of ML databases, http://www.ics.uci.edu/~mlearn/MLRepository.html
[13] Breiman, L.: Arcing Classifiers. The Annals of Statistics 26(3), 801–849 (1998)
[14] Freund, Y., Shapire, R.E.: A Decision-Theoretic Generalization of On-line Learning and its Application to Boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)