Embedded Feature Ranking for Ensemble MLP Classifiers

Terry Windeatt, Rakkrit Duangsoithong, Raymond Smith

Abstract-- A feature ranking scheme for MLP ensembles is proposed, along with a stopping criterion based upon the out-of-bootstrap (OOB) estimate. To solve multi-class problems feature ranking is combined with modified Error-Correcting Output Coding (ECOC). Experimental results on benchmark data demonstrate the versatility of the MLP base classifier in removing irrelevant features.

Index terms—Classification, Multilayer Perceptrons, Pattern Analysis, Pattern Recognition.

1 INTRODUCTION

Whether an individual classifier or an ensemble of classifiers is employed to solve a supervised learning problem, finding relevant features for discrimination is important. Most previous research on feature relevancy has focussed on individual classifiers, but in this paper the issue is addressed for an ensemble of Multi-layer perceptron (MLP) classifiers. The extension of feature relevancy to classifier ensembles is not straightforward, because of the inherent trade-off between accuracy and diversity [1]. The trade-off has long been recognised, and arises because diversity must decrease as base classifiers approach the highest levels of accuracy. There is no consensus on the best way to measure ensemble diversity, and the relationship between irrelevant features and diversity is not known.

Feature relevancy is particularly important for small sample size problems, that is when the number of patterns is fewer than the number of features [2]. With tens of features in the original set, feature selection using an exhaustive search is computationally prohibitive. Since the problem is known to be NP-hard [3], a greedy search scheme is required, and filter, wrapper and embedded approaches have been developed [4]. The advantage of an embedded method is that feature selection is inherent in the classifier itself, and there is no reliance upon a measure that is independent of the classifier.

Feature ranking is conceptually one of the simplest search schemes for feature selection, and has the advantage of scaling up to hundreds of features. Uni-dimensional feature-ranking methods consider each feature in isolation, but are disadvantaged by the implicit orthogonality assumption [4], whereas multi-dimensional methods consider correlations with remaining features. In this paper, we propose an ensemble of MLP classifiers that incorporates multi-dimensional feature ranking based on MLP weights. The ensemble contains a simple parallel Multiple Classifier System (MCS) architecture with homogenous MLP base classifiers.

Manuscript received April 23, 2009. This work was supported in part by UK Government, EPSRC under grant number E061664/1

T. Windeatt, R. Duangsoithong, R.S. Smith are with the Centre for Vision Speech and Signal Processing, Faculty of Electronics and Physical Sciences, University of Surrey, Guildford Surrey, GU2 7XH, UK (phone: +44-1483-9286; e-mail: t.windeatt@surrey.ac.uk)

It is generally believed that MLP weights in a single classifier are not suitable for identifying relevant features [5]. However, in this paper it is shown that Ensemble MLP weights, when combined with Recursive Feature Elimination (RFE), are effective for eliminating irrelevant features. An important issue for RFE is to determine when to stop eliminating features. In Section 2.1, the Ensemble Out-of-Bootstrap (OOB) estimate is proposed for the stopping criterion [6].

There has not been any systematic comparison of feature ranking methods in the context of MCS. Most previous approaches to feature selection with ensembles have focused on determining feature subsets to combine, but differ in the way the subsets are chosen. The Random Subspace Method (RSM) is the best-known method, and it was shown in [7] that a random choice of feature subset (allowing a single feature to be in more than one subset), improves performance for high-dimensional problems. In [2], forward feature and random (without replacement) selection methods are used to sequentially determine disjoint optimal subsets. In [8], feature subsets are chosen based on how well a feature correlates with a particular class. Ranking subsets of randomly chosen features before combining was reported in [9]. Bootstrap feature selection for ensembles was proposed in [10].

The main contributions are 1) feature ranking using ensemble MLP weights combined with RFE 2) OOB stopping criterion for optimal feature selection 3) extension to multi-class problems by combing RFE with weighted ECOC decoding strategy, and 4) incorporation of OOB estimate into ECOC decoding.

The paper is organised as follows. In Section 2, six feature ranking strategies are described. RFE is applied to three weight ranking strategies MLP, SVC (Support Vector Classifier) and FLD (Fisher's Linear Discriminant). The other three strategies are ranking by modified Boosting, and ranking by two statistical methods. Section 2.1 explains the criterion used to stop eliminating features, which is based on the OOB error estimate. In Section 3, multi-class problems are solved using Error-Correcting Output Coding (ECOC), modified to include problem-dependent decoding. The experimental results in Section 4 show the effectiveness of the embedded feature ranking method for two-class and multi-class problems.

2 FEATURE RANKING

In [11] feature ranking using single SVC was shown to give excellent results when combined with RFE, which operates recursively in two steps. First rank the features according to a suitable feature-ranking method and then identify and remove the r least ranked features. For efficiency reasons, usually $r \geq 2$, which produces a feature subset ranking. RFE only requires that, at each recursion, the least ranked subset does not contain a strongly relevant feature. Definitions of

redundancy, weak and strong relevance can be found in [12].

Feature selection using MLP weights was recently experimentally investigated in [13], but the emphasis was on retraining a single classifier, after each feature reduction. In contrast, we use ensemble feature ranking by MLP weights combined with RFE (*rfenn*). The output *O* of a single output single hidden-layer MLP, assuming sigmoid activation function *S* is given by

$$O = \sum_q S(\sum_p x_p W_{pq}^1) * W_q^2 \qquad (1)$$

where *p,q* are the input and hidden node indices, x_p is input feature, W^1 is the first layer weight matrix and W^2 is the output weight vector. In [14], a local feature selection gain w_p is derived from (1)

$$w_p = \sum_q \left| W_{pq}^1 * W_q^2 \right| \qquad (2)$$

The weight w_p in (2) is the sum over hidden nodes of the product of two weights connected via each hidden node to the *pth* feature, but has been found in general not to give a reliable feature-ranking [5]. However, when used with RFE it is only required to find the least relevant features. The ranking using product of weights in (2) is performed once for each MLP base classifier. Then individual rankings are summed for each feature, giving an overall ranking, which is used for eliminating the set of least relevant features at each recursive step.

For SVC the weights of the decision function are based on a small subset of patterns, the support vectors. In this paper, RFE incorporates linear SVC (*rfesvc)* in which linear decision function consists of the support vector weights, that is the weights that have not been driven to zero [11].

Fisher's criterion measures the separation between two sets of patterns in a direction *w*, and is defined for the projected patterns as the difference in means normalized by the averaged variance

$$J(w) = \frac{\left| w^T S_B w \right|}{\left| w^T S_W w \right|} \qquad (3)$$

where S_B is the between-class scatter matrix and S_W is the within-class scatter matrix. The objective of FLD is to find the transformation that maximises *J(w)* in (3). The optimal transformation ***w**** is known to be the solution of the following eigenvalue problem $S_B W - S_W W \Lambda = 0$, where Λ is a diagonal matrix whose elements are the eigenvalues of matrix $S_W^{-1} S_B$.

The idea behind the *noisy bootstrap* [15] (details of bootstrapping in Section 2.1) is to estimate the noise in the data and extend the training set by re-sampling with simulated noise. Therefore, the number of patterns may be increased by using a re-sampling rate greater than 100 percent, thus solving the small sample size problem. The noise model assumes a multi-variate Gaussian distribution with zero mean and diagonal covariance matrix, since there are generally insufficient number of patterns to make a reliable estimate of correlations between features. For each class, the standard deviation of each feature is used for the diagonal entry. The standard deviation of the noise added to normalised features is set to 0.25 and the ratio of number of samples to the number of features is set to 10. In *rfenb*, RFE incorporates the weight ranking defined by ***w**** in (3).

Boosting has become popular as a feature selection routine, in which a single feature on each Boosting iteration is selected that minimises the classification error on the weighted samples [16]. In our implementation, we use Adaboost with decision stump as weak learner.

Class separability measures are popular statistical feature ranking methods [17]. The one-dimensional method (*1dim*) chosen here is defined as $trace(S_W^{-1} S_B)$, where S_B and S_W are defined in (3). A fast multi-dimensional search method that has been shown to give good results with individual classifiers is Sequential Forward Floating Search (SFFS). SFFS improves on (plus *l* – take away *r*) algorithms by introducing dynamic backtracking [18].

2.1 OOB Stopping Criterion

Bootstrapping is applied to each base classifier in the ensemble, so that if μ training patterns are randomly sampled with replacement, approximately $(1-1/\mu))^{\mu} \cong 37\%$ are not seen and therefore in the OOB set. Let B be the set of classifiers, O_j the set of OOB patterns for *jth* classifier (*j=1...b*) and $E_m(\mathrm{A})$ the error estimate for ensemble applied to *mth* pattern over classifier subset $\mathrm{A} \subseteq \mathrm{B}$. The *jth* base classifier OOB error estimate $BCOOB_j$ is computed over patterns in O_j and should be distinguished from the ensemble classifier OOB

$$ECOOB = \sum_{m=1}^{\mu} E_m\left(\left\{ j \middle| j \in \mathrm{B}, m \in O_j \right\}\right) \qquad (4)$$

where E_m is 1 if majority vote disagrees with target class ω_m, otherwise 0. In (4) all training patterns contribute to the ECOOB estimate, but the only participating classifiers for each pattern are those that have not been used with that pattern for training (that is, approximately thirty-seven percent of classifiers).

The proposed procedure for selecting optimal set of features for the *nth* recursive step is as follows

while ECOOB(n) < ECOOB(n-1)
- rank features for *b* MLP base classifier using (2)
-sum rankings of *b* classifiers to produce overall ranking
-identify and remove *r* least relevant features

3 **MODIFIED ECOC**

Multi-class problems are solved using Error-Correcting Output Coding (ECOC) [19] [20], which is a two-stage process, coding followed by decoding. The coding step is defined by the binary *k xb* code word matrix Z that has one row (code word) for each of *k* classes, with each column defining one of *b* sub-problems that use a different labelling. If each element Z_{ij} $(i = 1 \ldots k, j = 1 \ldots b)$ is a binary variable z, a training pattern with target class *i* is re-labelled as class Ω_1 if $Z_{ij} = z$ and as class Ω_2 if $Z_{ij} = \bar{z}$, the complement of z. The two super-classes Ω_1 and Ω_2 represent, for each column, a different decomposition of the original problem. For example, if a column of Z is given by $[0\ 1\ 0\ 0\ 1\ 1]^T$, this would naturally be interpreted as a six-class problem in which patterns from classes 2,5,6 are assigned to Ω_1 with remaining patterns assigned to Ω_2. This

is in contrast to the conventional One-versus-rest code, defined by the diagonal code matrix. Usually codes are problem-independent, and theoretical and experimental evidence indicates that a long random code performs almost as well as a pre-defined code, optimised for its error-correcting properties [20]. In this paper, the code is random with near equal split of labels in each column [21].

An MLP base classifier is applied to each of the *b* sub-problems defined by Z, and the feature ranking scheme *rfenn*, described in Section 2, is used to eliminate irrelevant features. Therefore, at each recursive step, there are fewer features available for solving two-class decompositions. As described below, the ECOC decoding stage is made problem-dependent, so that it is able to adapt to the changing number of features.

Let the *jth* classifier produce an estimated probability $\hat{q}_{mj}$ that the *mth* pattern comes from the super-class defined by the *jth* decomposition. In the decoding step of ECOC, the *mth* pattern is assigned to the class $\hat{\omega}_m$ represented by the closest code word

$$\hat{\omega}_m = \arg\min_i \sum_{j=1}^{b} w_{ij} \left| Z_{ij} - \hat{q}_{mj} \right| \qquad (5)$$

where w_{ij} introduces problem-dependence into the decoding stage by allowing for *ith* class and *jth* classifier to be assigned a different weight. Conventional ECOC decoding is un-weighted with $w_{ij}=1$ in (5), L_1 norm decoding using soft decision $\hat{q}_{mj}$ and Hamming decoding using binarised hard decision. To facilitate ECOOB estimate for multiclass, (5) is modified by removing columns of Z if they correspond to classifiers that used the *mth* pattern for training, that is the summation is over the subset $\left\{ j \middle| j \in B, m \in O_j \right\}$ as in (4).

The weights w_{ij} in (5) are estimated using Walsh coefficients of a Boolean (binary-to-binary) mapping. The first order coefficients were derived from this mapping and used in [22] to define a measure of class separability, which is computed in Section 4 for experimental comparison. Let $y_{mj} \in \{0,1\}$ be the *jth* classifier binary output of the *mth* pattern with target class $\omega_m = t$. Define $y_{mj}=1$ if and only if the classifier assigns the correct super-class Z_{tj}. For target class *t*, the *jth* weight is defined as

$$w_{tj} = \sum_{\{m,n \mid \omega_m = t, \omega_n \neq t\}} (y_{mj} \wedge y_{nj} - \bar{y}_{mj} \wedge \bar{y}_{nj}) \qquad (6)$$

where $\wedge$ is logical AND. The summation in (6) is over all pairs of patterns, *mth* pattern chosen from class *t* and *nth* pattern chosen from remaining classes. The motivation is that the weight is computed as the difference between positive and negative correlations of class *t* versus the rest. Negative weights are set to zero and for each class $\sum_{j=1}^{b} w_j = 1$.

4 EXPERIMENTAL EVIDENCE

Experiments on two-class datasets compare the feature ranking schemes described in Section 2, and are designed to show that the OOB estimate in Section 2.1 may be used as a criterion for determining when to stop eliminating features. For multi-class datasets the proposed embedded feature ranking strategy is combined with problem-dependent ECOC decoding, which includes the OOB estimate.

Natural benchmark problems [23] and [24] are shown in Table 1 and Table 2. Noisy (mean 0 std 1) features are added after normalisation so that each dataset has a total of one hundred. Databases with one hundred features are chosen to facilitate comparison with a complex feature selection method such as SFFS. The experiments are performed with one hundred single hidden-layer MLP base classifiers, using the Levenberg-Marquardt training algorithm with default parameters. However, for ECOC experiments the number of ECOC columns is set to 200, except where otherwise stated. The random train/test split is [20/80, 10/90, 5/95]. The reason for using few patterns for training is to determine the small sample size performance. Random perturbation of MLP base classifiers is caused by different starting weights combined with bootstrapping, as described in Section 2.1. For non-linear MLP, the number of nodes and epochs is selected as an optimal choice on average over two-class and multi-class datasets using ECOOB [6] (8 nodes with 7 epochs for 2-class and 20 epochs for multi-class). Experiments are repeated twenty times and averaged, and we denote Ensemble and Base classifier test error by ECTE and BCTE respectively.

To assist in understanding results, Bias and Variance of 0/1 loss function according to Breiman's definition [25] are reported. The required estimate of the Bayes classifier is performed for 90/10 split using original features, and a Support Vector Classifier (SVC) with polynomial kernel run hundred times. The polynomial degree is varied as well as the regularisation constant. The lowest test error found is given in Table 1, and the classification for each pattern is stored for the bias/variance computation. All datasets achieved minimum with linear SVC, with the exception of *Ion* (degree 2).

The various feature-ranking schemes described in Section 2 are compared using MLP and SVC, with ranking criteria computed on the training set. When the number of features is reduced, the ratio of the number of patterns to features is changing, so that optimal classifier parameters will be varying. This makes it a complex problem, since theoretically an optimisation needs to be carried out after each feature reduction. To make a full comparison between MLP and SVC, we would need to search over the full parameter space, which is not feasible. For this reason we compare linear SVC with linear perceptron ensemble. Table 3 shows that the ensemble is fairly insensitive to the ranking scheme and the perceptron ensemble performs similarly to SVC. In particular, the more sophisticated schemes of SFFS and Boosting are slightly worse on average than the simpler schemes. Although the 1-dimensional method (*1dim*) is best on average for 20/80 split, as number of training patterns decreases, performance is slightly worse than RFE methods. Since the differences between feature selection schemes were in general not statistically significant (McNemar test 95% [26]), we show results graphically as the mean over all datasets, which clearly indicate the overall trend, despite small differences on individual datasets

The recursive step size for RFE is chosen using a logarithmic scale to start at 100 and finish at 2 features. Fig. 1 shows linear *rfenn* mean test error rates, BCOOB,

ECOOB, bias and variance over all seven two-class datasets. For the 20/80 split Fig. 1 (a) shows that minimum base classifier error is achieved with 5 features compared with Fig. 1 (b) 7 features for the ensemble. Fig. 1f) shows that bias is minimised at 11 features, demonstrating that the linear perceptron with bootstrapping benefits (in bias reduction) from a few extra noisy features. Fig. 1 (e) shows that Variance is reduced monotonically as number of features is reduced. Note also that according to Breiman's decomposition Fig. 1 (e) + (f) + 11.1 (mean Bayes) equals (a). Fig 1 (c) and (d) show that while BCOOB, ECOOB do not accurately predict the absolute value of BCTE, ECTE they are good predictors of optimal number of features.

Mean correlation coefficients between row/column pairs with respect to features for 2-class 20/80 linear and non-linear MLP ensemble are shown in Table 4. For comparison two additional measures are included, the pair-wise diversity Q [27] and class separability σ' [1]. Table 4 also shows the number of datasets that are significantly correlated at 95% confidence, when compared with random chance. The non-linear ensemble is better correlated (than linear ensemble) between ECOOB and ECTE, and the only dataset not significantly correlated is *cancer*. Both σ' and Q are correlated with BCTE, but not as highly as BCOOB.

For non-linear MLP base classifier with *rfenn*, mean ECTE over 2-class for [20/80, 10/90 5/95]% train/test splits was [13.9,15.7,17.9]% respectively, the improvement due mostly to *ion* dataset which has a high bias with respect to Bayes classifier. To determine an artificial performance limit for feature selection, we chose SFFS with the unrealistic case of full test set for tuning. The mean ECTE was [13.5, 14.1, 15.4]% showing that *rfenn* effectively eliminates irrelevant features, particularly for 20/80 split. Finally, *rfenn* without Bootstrapping showed that although variance is lower, bias is higher giving ECTE [15.7, 17.6, 20.0]%, demonstrating that Bootstrapping has beneficial effect on performance.

Fig. 2 shows weighted and un-weighted Decoding ECOC with *rfenn* and non-linear MLP base classifier as number of classifiers is reduced. Fig. 2 (c) and 2 (d) demonstrate that BCOOB, ECOOB are good predictors of the optimal number of features. Fig. 2 (e) shows that weighted decoding test error is smaller, when the number of features is greater than optimal. Below the optimal number, weighted decoding is inferior. It may be seen from Fig. 2 (d) that the ECOOB estimate gives quite reliable indication of optimal number of features down to 5 classifiers. The correlation with respect to features is shown in Table 5, from which it may be seen that ECOOB is highly correlated with ECTE, while σ' and BCOOB are highly correlated with BCTE.

5 DISCUSSION & CONCLUSION

An embedded feature ranking strategy based on MLP weights combined with Recursive Feature Elimination (RFE) is proposed, along with a stopping criterion based on Out-of-Bootstrap (OOB) estimate. The techniques work well for two-class problems, as well as for multi-class using modified decoding strategy for Error-Correcting Output Coding (ECOC). In [28] embedded feature ranking is applied to the Cohn-Kanade face expression database for detecting upper face *action units*, giving detection rates comparable with the best currently attainable.

Table 1: Two-class Datasets showing numbers of patterns, features and estimated Bayes error rate

DATASET	#pat	#con	#dis	%bayes
cancer	699	0	9	3.1
card	690	6	9	12.8
credita	690	3	11	14.1
diabetes	768	8	0	22.0
heart	920	5	30	16.1
ion	351	31	3	6.8
vote	435	0	16	2.8

Table 2: **Multi-class datasets showing numbers of patterns, classes, features**

DATASET	#pat	#class	#con	#dis
dermatology	366	6	1	33
ecoli	336	8	5	2
glass	214	6	9	0
iris	150	3	4	0
segment	2310	7	19	0
soybean	683	19	0	35
vehicle	846	4	18	0
vowel	990	11	10	1
wave	3000	3	21	0
yeast	1484	10	7	1

Table 3: Mean best error rates ECTE%/number of features for two-class problems (20/80) with five feature-ranking schemes (Mean 10/90, 5/95 also shown)

	Linear perceptron-ensemble classifier					**Linear SVC-classifier**				
	rfenn	**rfenb**	**1dim**	**SFFS**	**boost**	**rfesvc**	**rfenb**	**1dim**	**SFFS**	**boost**
diab	24.9/2	25.3/2	25.3/2	25.8/2	25.6/2	24.5/3	24.8/5	24.9/2	25.3/2	25.3/2
credita	16.5/5	15.7/3	14.6/2	15.6/2	15.5/2	15.7/2	15.1/2	14.6/2	15.4/2	15.1/2
cancer	4/7	4/5	4.1/5	4.4/3	4.9/7	3.7/7	3.7/7	3.8/11	4.2/5	4.5/7
heart	21/27	21/18	21/11	23/5	23/18	20/18	20/11	20/18	22/7	24/18
vote	5.5/5	5.3/7	5.6/18	5.7/2	5.5/2	4.8/2	4.8/2	4.7/2	4.3/3	4.7/2
ion	18/11	16.7/3	14.8/3	15.8/3	18.1/2	15/11	15.9/7	15.3/5	17.9/5	19.5/5
card	15.7/7	15/2	14.7/2	16.9/2	14.8/2	15.5/2	14.8/2	14.5/2	16.6/2	14.5/2
Mean20/80	15.1	14.6	14.2	15.4	15.4	14.2	14.2	13.9	15.1	15.3
Mean10/90	16.3	16.3	16.6	18.0	17.6	15.5	15.7	15.8	17.5	17.3
Mean5/95	18.4	18.5	20.0	21.3	21.3	17.0	17.7	18.4	20.3	20.7

Table4: Mean Correlation coefficient/number of significant correlations over seven two-class datasets 20/80 for linear and non-linear *rfenn*

	ECOOB	BCOOB	Q	σ′
ECTE(lin)	0.77/5	0.51/5	-0.17/3	-0.13/2
BCTE(lin)	0.81/5	0.97/7	-0.70/5	-0.72/6
ECTE(nlin)	0.85/6	0.46/2	-0.04/1	0.04/4
BCTE(nlin)	0.76/5	0.98/7	-0.79/6	-0.73/6

Table 5: Mean Correlation coefficient/ number of significant correlations over ten multi-class datasets 20/80 for non-linear *rfenn*

	ECOOB	BCOOB	Q	σ′
ECTE	0.99/10	0.81/9	-0.03/4	-0.45/4
BCTE	0.88/9	1.0/10	-0.42/4	-0.81/8

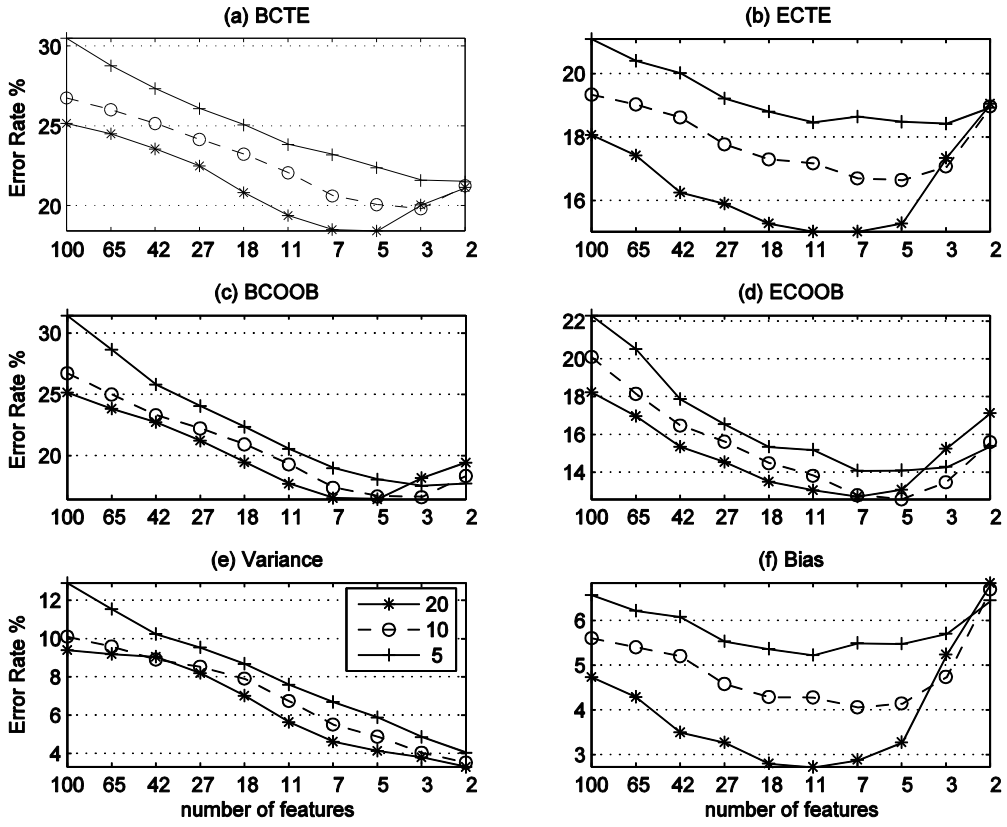

Figure 1: Mean test error rates, OOB estimates, Bias, Variance for *rfenn* over 2-classs Datasets [20/80, 10/90, 5/95] train/test split

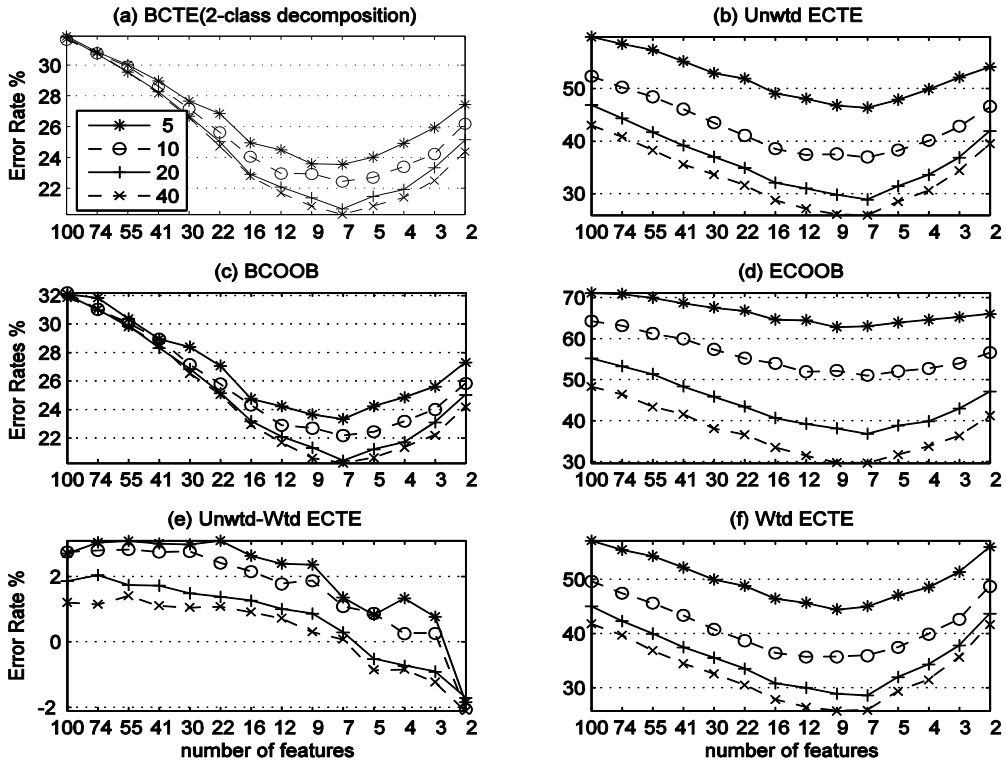

Figure 2: Mean test error rates, OOB estimates, difference between Un-weighted and Weighted ECOC Decoding for *rfenn* over multi-class datasets 20/80 train/test split with [5,10,20,40] base classifiers

REFERENCES

[1] T. Windeatt, "Diversity measures for Multiple Classifier System analysis and design," *Information Fusion*, vol. 6, no. 1, pp. 21-36, 2004

[2] M. Skuruchina and R. P. W Duin, "Combining feature subsets in feature selection," *Proc. 6th Int. Workshop Multiple Classifier Systems*, Seaside, Calif. June, 2005, pp. 165-174.

[3] R. Kohavi and G. H. John," Wrappers for feature subset selection," *Artificial Intelligence Journal, special issue on relevance*, vol. 97, no. 1-2, pp. 273-324, 1997.

[4] I Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research* vol.3, pp. 1157-1182, 2003.

[5] W. Wang, P. Jones, and D. Partridge, "Assessing the impact of input features in a feedforward neural network," *Neural Computing and Applications*, vol. 9, no. 2, pp. 101-112, 2000.

[6] T. Windeatt and M. Prior, "Stopping criteria for ensemble-based feature selection," *Proc. 7th Int. Workshop Multiple Classifier Systems*, Prague, Czech Republic, May , 2007, pp. 271-281.

[7] T. K. Ho**, "**The Random subspace method for constructing decision forests," *IEEE Trans. PAMI*, pp. 832 – 844, Aug. 1998.

[8] N.Oza and K. Tumer," Input Decimation ensembles," *Proc. 2nd Int. Workshop Multiple Classifier Systems*, Cambridge, UK, July, 2001, pp. 238-247.

[9] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291-1302, 2003.

[10] R. Duangsoithong and T. Windeatt, "Bootstrap feature selection for ensemble classifiers," *Advances in Data Mining, Lecture Notes in Computer Science*, vol. 6171, pp. 28-41, 2010.

[11] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik," Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389-422, 2002.

[12] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, vol. 5, pp. 1205-1224, 2004.

[13] E. Romero and J. M. Sopena, "Performing feature selection with MLP," *IEEE Trans. Neural Networks*, vol. 19, pp. 431-441, Mar. 2008.

[14] C. Hsu, H. Huang, and D. Schuschel, "The ANNIGMA-wrapper approach to fast feature selection for neural nets," *IEEE Trans. System, Man and Cybernetics*-Part B:Cybernetics, vol. 32, pp. 207-212, Apr. 2002.

[15] T. Windeatt, M. Prior, N. Efron, and N. Intrator, "Ensemble-based feature selection criteria," *Proc. Conf. on Machine Learning Data Mining* , Leipzig, Germany, July, 2007, pp 168-182.

[16] P. Silapachote, D. R. Karuppiah,, and A. R. Hanson, "Feature selection using Adaboost for face expression recognition," *Proc. Conf. on Visualisation, Imaging and Image Processing*, Marbella, Spain, Sept., 2004, pp. 84-89.

[17] K. Fukunaga , *Introduction to statistical pattern recognition*, Academic Press, 1990.

[18] F. Heijden, R. P.W. Duin, D. Ridder, and D. M.J. Tax, *Classification, Parameter Estimation and State Estimation*, Wiley, 2004.

[19] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research* vol. 2, pp. 263-286, 1995.

[20] T Windeatt and R Ghaderi, "Coding and decoding strategies for multiclass learning problems," *Information Fusion*, vol. 4, no. 1, pp. 11-21, 2003.

[21] E. L. Allwein, R. E. Schapire, and Y. Singer," Reducing multi-class to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, pp. 113-141, 2000.

[22] T. Windeatt, "Accuracy/ Diversity and ensemble classifier design," *IEEE Trans Neural Networks*, vol. 17, pp. 1194- 1211, Sept. 2006.

[23] L. Prechelt, Proben1: A set of neural network Benchmark Problems and Benchmarking Rules, *Tech Report 21/94*, Univ. Karlsruhe, Germany, Sept., 1994.

[24] C. J. Merz and P. M. Murphy, UCI repository of ML databases, [Online], http://www.ics.uci.edu/~mlearn/MLRepository.html.

[25] L. Breiman, "Arcing classifiers," *The Annals of Statistics*, vol. 26, no. 3, pp. 801-849, 1998.

[26] T. G. Dietterich, "Approx. statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895-1923, 1998.

[27] L. I. Kuncheva and C.J. Whitaker," Measures of diversity in classifier ensembles," *Machine Learning* vol. 51, no. 2, pp. 181-207, 2003.

[28] T. Windeatt, "Weighted decoding ECOC for facial action unit classification," *Proc. Workshop on Supervised and Unsupervised Ensemble Methods and their Applications*, Patras, Greece, July, 2008, pp. 26-30.