Feature Ranking Ensembles for Facial Action Unit Classification

Terry Windeatt and Kaushala Dias

Centre for Vision, Speech and Signal Proc (CVSSP), University of Surrey,
Guildford, Surrey, United Kingdom GU2 7XH
t.windeatt@surrey.ac.uk

Abstract. Recursive Feature Elimination RFE combined with feature-ranking is an effective technique for eliminating irrelevant features. In this paper, an ensemble of MLP base classifiers with feature-ranking based on the magnitude of MLP weights is proposed. This approach is compared experimentally with other popular feature-ranking methods, and with a Support Vector Classifier SVC. Experimental results on natural benchmark data and on a problem in facial action unit classification demonstrate that the MLP ensemble is relatively insensitive to the feature-ranking method, and simple ranking methods perform as well as more sophisticated schemes. The results are interpreted with the assistance of bias/variance of 0/1 loss function.

1 Introduction

Consider a supervised learning problem, in which many features are suspected to be irrelevant. To ensure good generalisation performance dimensionality needs to be reduced, otherwise there is the danger that the classifier will specialise on features that are not relevant for discrimination, that is the classifier may over-fit the data. It is particularly important to reduce the number of features for small sample size problems, where the number of patterns is less than or of comparable size to the number of features [1]. To reduce dimensionality, features may be extracted (for example Principal Component Analysis PCA) or selected. Feature extraction techniques make use of all the original features when mapping to new features but, compared with feature selection, are difficult to interpret in terms of the importance of original features.

Feature selection has received attention for many years from researchers in the fields of pattern recognition, machine learning and statistics. The aim of feature selection is to find a feature subset from the original set of features such that an induction algorithm that is run on data containing only those features generates a classifier that has the highest possible accuracy [2]. Typically with tens of features in the original set, an exhaustive search is computationally prohibitive. Indeed the problem is known to be NP-hard [2], and a greedy search scheme is required. For problems with hundreds of features, classical feature selection schemes are not greedy enough, and filter, wrapper and embedded approaches have been developed [3].

Although feature-ranking has received much attention in the literature, there has been relatively little work devoted to handling feature-ranking explicitly in the

L. Prevost, S. Marinai, and F. Schwenker (Eds.): ANNPR 2008, LNAI 5064, pp. 267–279, 2008.

context of Multiple Classifier System (MCS). Most previous approaches 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 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 [1], forward feature and random (without replacement) selection methods are used to sequentially determine disjoint optimal subsets. In [4], 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 [5].

In this paper an MLP ensemble using Recursive Feature Elimination RFE [12] is experimentally compared for different feature-ranking methods. Ensemble techniques are discussed in Section 2, and feature-ranking strategies in Section 3. The datasets, which include a problem in face expression recognition, are described in Section 4, with experimental results in Section 5.

2 Ensembles, Bootstrapping and Bias/Variance Analysis

In this paper, we assume a simple parallel Multiple Classifier System (MCS) architecture with homogenous MLP base classifiers and majority vote combiner. A good strategy for improving generalisation performance in MCS is to inject randomness, the most popular strategy being Bootstrapping. An advantage of Bootstrapping is that the Out-of-Bootstrap (OOB) error estimate may be used to tune base classifier parameters, and furthermore, the OOB is a good estimator of when to stop eliminating features [6]. Normally, deciding when to stop eliminating irrelevant features is difficult and requires a validation set or cross-validation techniques.

Bootstrapping is an ensemble technique which implies that if μ training patterns are randomly sampled with replacement, $(1-1/\mu))^{\mu} \cong 37\%$ are removed with remaining patterns occurring one or more times. The base classifier OOB estimate uses the patterns left out of training, and should be distinguished from the ensemble OOB. For the ensemble OOB, all training patterns contribute to the 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). Note that OOB gives a biased estimate of the absolute value of generalisation error [7], but for tuning purposes the estimate of the absolute value is not important [8]. Bagging, that is Bootstrapping with majority vote combiner, and Boosting (Section 3.3) are probably the most popular MCS methods.

The use of Bias and Variance for analysing multiple classifiers is motivated by what appears to be analogous concepts in regression theory. The notion is that averaging a large number of classifiers leads to a smoothing out of error rates. Visualisation of simple two-dimensional problems appears to support the idea that Bias/Variance is a good way of quantifying the difference between the Bayes decision boundary and the ensemble classifier boundary. However, there are difficulties with the various Bias/Variance definitions for 0/1 loss functions. A comparison of Bias/Variance definitions [9] shows that no definition satisfies all properties that would ideally be expected for 0/1 loss function. In particular, it is shown that it is impossible for a single definition to satisfy both zero Bias and Variance for Bayes

classifier, and additive Bias and Variance decomposition of error (as in regression theory).

Also, the effect of bias and variance on error rate cannot be guaranteed. It is easy to think of example probability distributions for which bias and variance are constant but error rate changes with distribution, or for which reduction in variance leads to increase in error rate [9] [11]. Besides these theoretical difficulties, there is the additional consideration that for real problems the Bayes classification needs to be known or estimated. Although some definitions, for example [10], do not require this, the consequence is that the Bayes error is ignored.

In our experiments, we use Breiman's definition [11] which is based on defining Variance as the component of classification error that is eliminated by aggregation. Patterns are divided into two sets, the Bias set B containing patterns for which the Bayes classification disagrees with the aggregate classifier and the Unbias set U containing the remainder. Bias is computed using B patterns and Variance is computed using U patterns, but both Bias and Variance are defined as the difference between the probabilities that the Bayes and base classifier predict the correct class label. Therefore, the reducible error (what we have control over) with respect to a pattern is either assigned to Bias or Variance, an assumption that has been criticised [9]. However, this definition has the nice property that the error of the base classifiers can be decomposed into additive components of Bayes error, Bias and Variance.

3 Feature-Ranking and RFE

RFE is a simple algorithm [12], and operates recursively as follows:

1) Rank the features according to a suitable feature-ranking method
2) Identify and remove the r least ranked features

If $r \geq 2$, which is usually desirable from an efficiency viewpoint, this produces a feature subset ranking. The main advantage of RFE is that the only requirement to be successful is that at each recursion the least ranked subset does not contain a strongly relevant feature [13]. In this paper we use RFE with MLP weights, SVC weights (Section 3.1), and noisy bootstrap (Section 3.2).

The issues in feature-ranking can be quite complex, and feature relevance, redundancy and irrelevance has been explicitly addressed in many papers. As noted in [13] it is possible to think up examples for which two features may appear irrelevant by themselves but be relevant when considered together. Also adding redundant features can provide the desirable effect of noise reduction.

One-dimensional feature-ranking methods consider each feature in isolation and rank the features according to a scoring function $Score(j)$ *where* $j=1\ldots p$ is a feature, for which higher scores usually indicate more influential features. One-dimensional functions ignore all $p-1$ remaining features whereas a multi-dimensional scoring function considers correlations with remaining features. According to [3] one-dimensional methods are disadvantaged by implicit orthogonality assumption, and have been shown to be inferior to multi-dimensional methods that consider all features simultaneously. However, there has not been any systematic comparison of single and multi-dimensional methods in the context of ensembles.

In this paper, the assumption is that all feature-ranking strategies use the training set for computing ranking criterion (but see Section 5 in which the test set is used for best case scenario). In Sections 3.1-3.4 we describe the ranking strategies that are compared in Section 5, denoted as *rfenn, rfesvc (Section 3.1) rfenb (Section 3.2) boost (Section 3.3) and SFFS, 1dim (Section 3.4).* Note that SVC, Boosting and statistical ranking methods are well-known so that the technical details are omitted.

3.1 Ranking by Classifier Weights (*rfenn, rfesvc)*

The equation for the output O of a single output single hidden-layer MLP, assuming sigmoid activation function S is given by

$$O = \sum_j S(\sum_i x_i W_{ij}^1) * W_j^2 \tag{1}$$

where i,j are the input and hidden node indices, x_i 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_i is derived form equation (1)

$$w_i = \sum_j \left| W_{ij}^1 * W_j^2 \right| \tag{2}$$

This product of weights strategy has been found in general not to give a reliable feature-ranking [15]. However, when used with RFE it is only required to find the least relevant features. The ranking using product of weights is performed once for each MLP base classifier. Then individual rankings are summed for each feature, giving an overall ranking that is used for eliminating the set of least relevant features in RFE.

For SVC the weights of the decision function are based on a small subset of patterns, known as support vectors. In this paper we restrict ourselves to the linear SVC in which linear decision function consists of the support vector weights, that is the weights that have not been driven to zero.

3.2 Ranking by Noisy Bootstrap *(rfenb)*

Fisher's criterion measures the separation between two sets of patterns in a direction $\boldsymbol{w}$, and is defined for the projected patterns as the difference in means normalised by the averaged variance. FLD is defined as the linear discriminant function for which $J(\boldsymbol{w})$ is maximized

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

where, S_B is the between-class scatter matrix and S_W is the within-class scatter matrix (Section 3.4). The objective of FLD is to find the transformation matrix $\boldsymbol{w}^*$ that maximises $J(\boldsymbol{w})$ in equation (3) and $\boldsymbol{w}^*$ is known to be the solution of the following eigenvalue problem $S_B - S_W \Lambda = 0$ where Λ is a diagonal matrix whose elements are

the eigenvalues of matrix $S_W^{-1}S_B$. Since in practice S_W is nearly always singular, dimensionality reduction is required. The idea behind the *noisy bootstrap* [16] 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. 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 any correlations between features. Two parameters to tune are the noise added γ and the sample to feature ratio *s2f*. We set for our experiments $\gamma = 0.25$ and *s2f = 1* [17].

3.3 Ranking by Boosting *(boost)*

Boosting, which combines with a fixed weighted vote is more complex than Bagging in that the distribution of the training set is adaptively changed based upon the performance of sequentially constructed classifiers. Each new classifier is used to adaptively filter and re-weight the training set, so that the next classifier in the sequence has increased probability of selecting patterns that have been previously misclassified. The algorithm is well-known and has proved successful as a classification procedure that 'boosts' a weak learner, with the advantage of minimal tuning. More recently, particularly in the Computer Vision community, Boosting has become popular as a feature selection routine, in which a single feature is selected on each Boosting iteration [18]. Specifically, the Boosting algorithm is modified so that, on each iteration, the individual feature is chosen which minimises the classification error on the weighted samples [19]. In our implementation, we use Adaboost with decision stump as weak learner.

3.4 Ranking by Statistical Criteria *(1dim, SFFS)*

Class separability measures are popular for feature-ranking, and many definitions use S_B and S_W (equation (3)) [20]. Recall that S_W is defined as the scatter of samples around respective class expected vectors and S_B as the scatter of the expected vectors around the mixture mean. Although many definitions have been proposed, we use trace($S_W^{-1} * S_B$), a one-dimensional method.

A fast multi-dimensional search method that has been shown to give good results with individual classifiers is Sequential Floating Forward Search (SFFS). It improves on (plus l – take away r) algorithms by introducing dynamic backtracking. After each forward step, a number of backward steps are applied, as long as the resulting subsets are improved compared with previously evaluated subsets at that level. We use the implementation in [21] for our comparative study.

4 Datasets

The first set of experiments use natural benchmark two-class problems selected from [22] and [23] and are shown in Table 1. For datasets with missing values the scheme suggested in [22] is used. The original features are normalised to mean 0 std 1 and the number of features increased to one hundred by adding noisy features (Gaussian std

Table 1. Benchmark Datasets showing numbers of patterns, continuous and discrete features and estimated Bayes error rate

DATASET	#pat	#con	#dis	%error
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

0.25). All experiments use random training/testing splits, and the results are reported as mean over twenty runs. Two-class benchmark problems are split 20/80 (20% training, 80% testing) 10/90, 5/95 and use 100 base classifiers.

The second set of experiments addresses a problem in face expression recognition, which has potential application in many areas including human-computer interaction, talking heads, image retrieval, virtual reality, human emotion analysis, face animation, biometric authentication [24]. The problem is difficult because facial expression depends on age, ethnicity, gender, and occlusions due to cosmetics, hair, glasses. Furthermore, images may be subject to pose and lighting variation. There are two approaches to automating the task, the first concentrating on what meaning is conveyed by facial expression and the second on categorising deformation and motion into visual classes. The latter approach has the advantage that the interpretation of facial expression is decoupled from individual actions. In FACS (facial action coding system) [25], the problem is decomposed into forty-four facial action units (e.g. *au1* inner brow raiser). The coding process requires skilled practitioners and is time-consuming so that typically there are a limited number of training patterns. These characteristics make the problem of face expression classification relevant and suitable to the feature-ranking techniques proposed in this paper.

The database we use is Cohn-Kanade [26], which contains posed (as opposed to the more difficult spontaneous) expression sequences from a frontal camera from 97 university students. Each sequence goes from neutral to target display but only the last image is *au* coded. Facial expressions in general contain combinations of action units (*aus*), and in some cases *aus* are non-additive (one action unit is dependent on another). To automate the task of *au* classification, a number of design decisions need to be made, which relate to the following a) subset of image sequences chosen from the database b) whether or not the neutral image is included in training c) image resolution d) normalisation procedure e) size of window extracted from the image, if at all f) features chosen for discrimination, g) feature selection or feature extraction procedure h) classifier type and parameters, and i) training/testing protocol. Researchers make different decisions in these nine areas, and in some cases are not explicit about which choice has been made. Therefore it is difficult to make a fair comparison with previous results.

We concentrate on the upper face around the eyes, (involving *au1, au2, au4, au5, au6, au7*) and consider the two-class problem of distinguishing images containing

inner brow raised (*au1*), from images not containing *au1*. The design decisions we made were

a) all image sequences of size 640 x 480 chosen from the database
b) last image in sequence (no neutral) chosen giving 424 images, 115 containing *au1*
c) full image resolution, no compression
d) manually located eye centres plus rotation/scaling into 2 common eye coordinates
e) window extracted of size 150 x 75 pixels centred on eye coordinates
f) Forty Gabor filters [18], five special frequencies at five orientations with top 4 principle components for each Gabor filter, giving 160-dimensional feature vector
g) Comparison of feature selection schemes described in Section 3
h) Comparison of MLP ensemble and Support Vector Classifier
i) Random training/test split of 90/10 and 50/50 repeated twenty times and averaged

With reference to b), some studies use only the last image in the sequence but others use the neutral image to increase the numbers of *non-aus*. Furthermore, some researchers consider only images with single *au*, while others use combinations of *aus*. We consider the more difficult problem, in which neutral images are excluded and images contain combinations of *aus*. With reference to d) there are different approaches to normalisation and extraction of the relevant facial region. To ensure that our results are independent of any eye detection software, we manually annotate the eye centres of all images, and subsequently rotate and scale the images to align the eye centres horizontally. A further problem is that some papers only report overall error rate. This may be mis-leading since class distributions are unequal, and it is possible to get an apparently low error rate by a simplistic classifier that classifies all images as *non-au1*. For the reason we report area under ROC curve, similar to [18].

5 Experimental Evidence

The purpose of the experiments is to compare the various feature-ranking schemes described in Section 3, using an MLP ensemble and a Support Vector Classifier. The SVC is generally recognised to give superior results when compared with other single classifiers. A difficulty with both MLPs and SVCs is that parameters need to be tuned. In the case of SVC, this is the kernel and regularisation constant C. For MLP ensemble, it is the number of hidden nodes and number of training epochs. There are other tuning parameters for MLPs, such as learning rate but the ensemble has been shown to be robust to these parameters [8]. 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. In general, this makes it a very 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 the two-class problems in table 1, we compare linear SVC with linear perceptron ensemble. We found that the differences

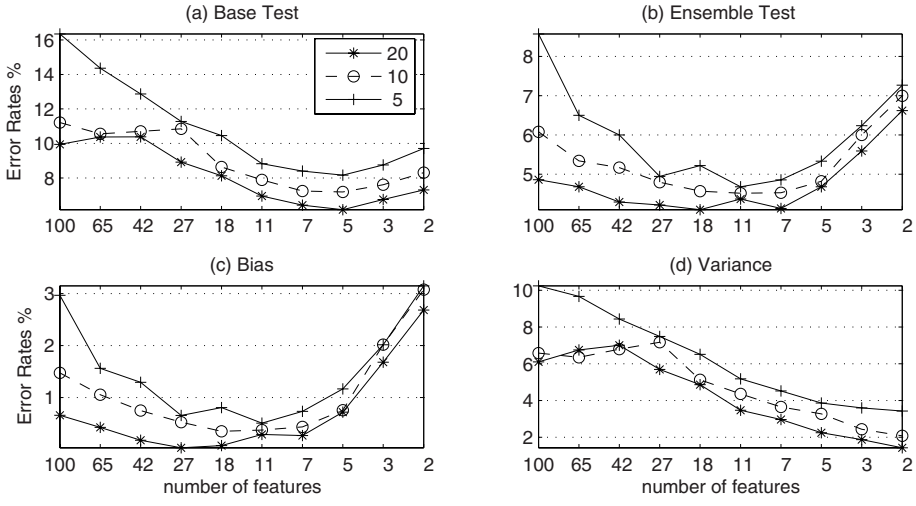

Fig. 1. Mean test error rates, Bias, Variance for RFE perceptron ensemble with Cancer Dataset 80/20, 10/90. 5/95 train/test split

between feature selection schemes were not statistically significant (McNemar test 5% [27]), and we show results graphically and report the mean over all datasets.

Random perturbation of the MLP base classifiers is caused by different starting weights on each run, combined with bootstrapped training patterns, Section 2. The experiment is performed with one hundred single hidden-layer MLP base classifiers, using the Levenberg-Marquardt training algorithm with default parameters. The feature-ranking criterion is given in equ. (2). In our framework, we vary the number of hidden nodes, and use a single node for linear perceptron. We checked that results were consistent for Single layer perceptron (SLP), using absolute value of orientation weights to rank features.

In order to compute bias and variance we need to estimate the Bayes classifier for the 2-class benchmark problems. The estimation is performed for 90/10 split using original features in Table 1, and a SVC with polynomial kernel run 100 times. The polynomial degree is varied as well as the regularisation constant. The lowest test error found is given in Table 1, and the classifications are stored for the bias/variance computation. All datasets achieved minimum with linear SVC, with the exception of 'Ion' (degree 2).

Figure 1 shows RFE linear MLP ensemble results for 'Cancer' 20/80, 10/90, 5/95 which has 140, 70, 35 training patterns respectively. With 100 features the latter two splits give rise to small sample size problem, that is number of patterns less than number of features [1]. The recursive step size for RFE is chosen using a logarithmic scale to start at 100 and finish at 2 features. Figure 1 (a) (b) show base classifier and ensemble test error rates, and (c) (d) the bias and variance as described in Section 2. Consider the 20/80 split for which Figure 1 (a) shows that minimum base classifier error is achieved with 5 features compared with figure (b) 7 features for the ensemble. Notice that the ensemble is more robust than base classifiers with respect to noisy

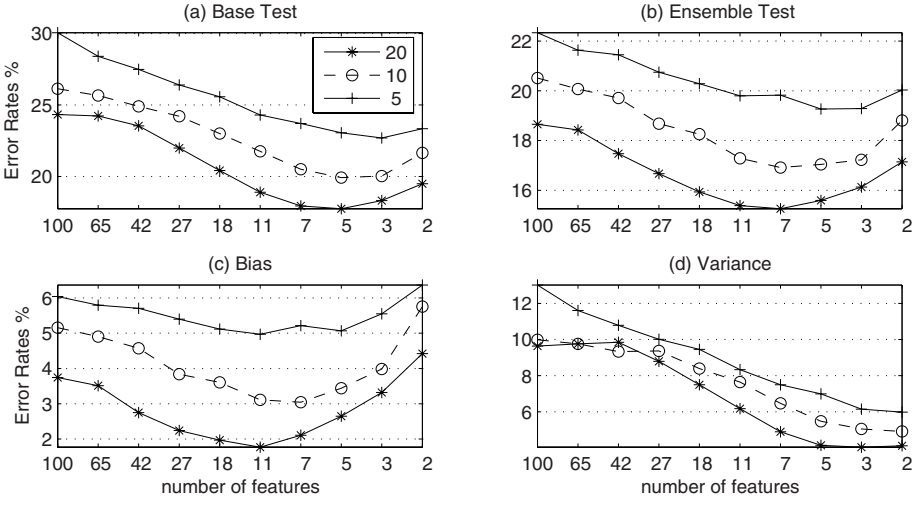

Fig. 2. Mean test error rates, Bias, Variance for RFE MLP ensemble over seven 2-class Datasets 80/20, 10/90. 5/95 train/test split

features. In fact, Figure 1 (c) shows that bias is minimised at 27 features, demonstrating that the linear perceptron with bootstrapping benefits (in bias reduction) from a few extra noisy features. Figure 1 (d) shows that Variance is reduced monotonically as number of features is reduced, and between 27 and 7 features the Variance reduction more than compensates for bias increase. Note also that according to Breiman's decomposition (Section 2), (c) + (d) + 3.1 (Bayes) equals (a).

Figure 2 shows RFE linear MLP ensemble mean test error rates, bias and variance over all seven datasets from table 1. On average, the base classifier achieves minimum error rate at 5 features and the ensemble at 7 features. Bias is minimised at 11 features and Variance at 3 features. For the 5/95 split there appears to be too few patterns to reduce bias, which stays approximately constant as features are reduced. Note that for SVC (not shown) the error is due entirely to bias, since variance is zero.

The comparison for various schemes defined in Section 3 can be found in Table 2. It may be seen that the ensemble is fairly insensitive to the ranking scheme and the linear 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 (Section 3.4) is best on average for 20/80 split, as number of training patterns decreases, performance is slightly worse than RFE methods. We also tried MLP base classifier with 8 nodes 7 epochs which was found to be the best setting without added noisy features [8]. The mean ensemble rate for 20/80, 10/90 5/95 was 14.5%,15.7%, 17.9% respectively the improvement due mostly to 'ion' dataset which has a high bias with respect to Bayes classifier.

To determine the potential effect of using a validation set with a feature selection strategy, we chose SVC plus SFFS with the unrealistic case of full test set for tuning. The mean ensemble rate for 20/80, 10/90 5/95 was 13.3%, 14.0%, 15.0% for SVC

Table 2. Mean best error rates (%)/number of features for seven two-class problems (20/80) with five feature-ranking schemes (Mean 10/90, 5/95 also shown)

	perceptron-ensemble classifier					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

Table 3. Mean best error rates (%)/number of features for au1 classification 90/10 with five feature ranking schemes

MLP-ensemble classifier					SVC-classifier				
rfenn	**rfenb**	**1dim**	**SFFS**	**boost**	**rfesvc**	**rfenb**	**1dim**	**SFFS**	**boost**
10.0/28	10.9/43	10.9/43	12.3/104	11.9/43	11.6/28	12.1/28	11.9/67	13.9/67	12.4/43

and 13.5%, 14.1%, 15.4% for MLP. We also repeated *rfenn* without Bootstrapping, showing that although variance is lower, bias is higher and achieved 15.7%, 17.6%, 20.0% respectively, demonstrating that Bootstrapping has beneficial effect on performance.

Table 3 shows feature-ranking comparison for *au1* classification from the Cohn-Kanade database as described in Section 4. It was found that lower test error was obtained with non-linear base classifier and Figure 3 shows test error rates, using an MLP ensemble with 16 nodes 10 epochs. The minimum base error rate for 90/10 split is 16.5% achieved for 28 features, while the ensemble is 10.0% at 28 features. Note that for 50/50 split there are too few training patterns for feature selection to have much effect. Since class distributions are unbalanced, the overall error rate may be mis-leading, as explained in Section 4. Therefore, we show the true positive rate in Figure 3 c) and area under ROC in Figure d). Note that only 71% of *au1s* are correctly recognised. However, by changing the threshold for calculating the ROC, it is clearly possible to increase the true positive rate at the expense of false negatives. Nevertheless, it is believed that the overall ensemble rate of 10% is among the best for *au1* on this database (recognising the difficulty of making fair comparison as explained in Section 4). We did try SVC for degree 2,3,4 polynomials with C varying,

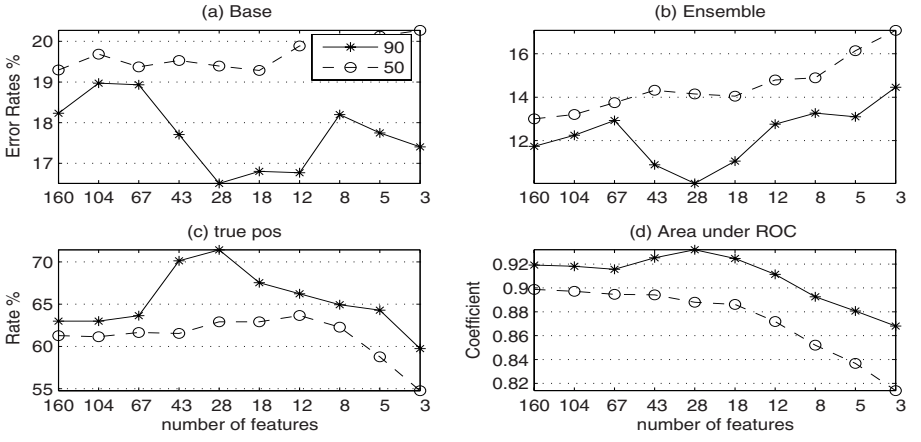

Fig. 3. Mean test error rates, True Positive and area under ROC for RFE MLP ensemble for au1 classification 90/10. 50/50 train/test split

but did not improve on degree 1 results. The results are not presented but the performance of SVC was very sensitive to regularisation constant C, which makes it difficult to tune and we did not try different kernels.

6 Discussion

There is conflicting evidence over whether an SVC ensemble gives superior results compared with single SVC, but in [28] it is claimed that an SVC ensemble with low bias classifiers gives better results. However, it is not possible to be definitive, without searching over all kernels and regularisation constants C. In our experiments, we chose to consider only linear SVC, and found the performance to be sensitive to C. In contrast, the ensemble is relatively insensitive to number of nodes and epochs [8], and this is an advantage of the MLP ensemble. However, we believe it is likely that we could have achieved comparable results to MLP ensemble by searching over different kernels and values of C for SVC.

The feature-ranking approaches have been applied to a two-class problem in facial action unit classification. The problem of detecting action units is naturally a multi-class problem, and the intention is to employ multi-class approaches that decompose the problem into two-class problems, such as Error-Correcting Output Coding (ECOC) [29].

7 Conclusion

A bootstrapped MLP ensemble, combined with RFE and product of weights feature-ranking, is an effective way of eliminating irrelevant features. The accuracy is comparable to SVC but has the advantage that the OOB estimate may be used to tune parameters and to determine when to stop eliminating features. Simple feature-

ranking techniques, such as 1-dimensional class separability measure or product of MLP weights plus RFE, perform at least as well as more sophisticated techniques such as multi-dimensional methods of SFFS and Boosting.

References

1. Skuruchina, M., Duin, R.P.W.: Combining feature subsets in feature selection. In: Oza, N., Polikar, R., Roli, F., Kittler, J. (eds.) Proc. 6th Int. Workshop Multiple Classifier Systems, Seaside, Calif. USA, June 2005. LNCS, pp. 165–174. Springer, Heidelberg (2005)
2. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence Journal, special issueon relevance 97(1-2), 273–324 (1997)
3. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
4. Oza, N., Tumer, K.: Input Decimation ensembles: decorrelation through dimensionality reduction. In: Kittler, J., Roli, F. (eds.) Proc. 2nd Int. Workshop Multiple Classifier Systems, Cambridge, UK. LNCS, pp. 238–247. Springer, Heidelberg (2001)
5. Bryll, R., Gutierrez-Osuna, R., Quek, F.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition 36, 1291–1302 (2003)
6. Windeatt, T., Prior, M.: Stopping Criteria for Ensemble-based Feature Selection. In: Proc. 7th Int. Workshop Multiple Classifier Systems, Prague, May 2007. LNCS, pp. 271–281. Springer, Heidelberg (2007)
7. Bylander, T.: Estimating generalisation error two-class datasets using out-of-bag estimate. Machine Learning 48, 287–297 (2002)
8. Windeatt, T.: Accuracy/ Diversity and Ensemble Classifier Design. IEEE Trans Neural Networks 17(5), 1194–1211 (2006)
9. James, G.: Variance and Bias for General Loss Functions. Machine Learning 51(2), 115–135 (2003)
10. Kong, E.B., Dietterich, T.G.: Error- Correcting Output Coding corrects Bias and Variance. In: 12th Int. Conf. Machine Learning, San Francisco, pp. 313–321 (1995)
11. Breiman, L.: Arcing Classifiers. The Annals of Statistics 26(3), 801–849 (1998)
12. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46(1-3), 389–422 (2002)
13. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224 (2004)
14. Hsu, C., Huang, H., Schuschel, D.: The ANNIGMA-wrapper approach to fast feature selection for neural nets. IEEE Trans. System, Man and Cybernetics-Part B:Cybernetics 32(2), 207–212 (2002)
15. Wang, W., Jones, P., Partridge, D.: Assessing the impact of input features in a feedforward neural network. Neural Computing and Applications 9, 101–112 (2000)
16. Efron, N., Intrator, N.: The effect of noisy bootstrapping on the robustness of supervised classification of gene expression data. In: IEEE Int. Workshop on Machine Learning for Signal Processing, Brazil, pp. 411–420 (2004)
17. Windeatt, T., Prior, M., Effron, N., Intrator, N.: Ensemble-based Feature Selection Criteria. In: Proc. Conference on Machine Learning Data Mining MLDM2007, Leipzig, July 2007, pp. 168–182 (2007) ISBN 978-3-940501-00-4

18. Bartlett, M.S., Littlewort, G., Lainscsek, C., Fasel, I., Movellan, J.: Machine learning methods for fully automatic recognition of facial expressions and facial actions. In: IEEE Conf. Systems, Man and Cybernetics, October 2004, vol. 1, pp. 592–597 (2004)
19. Silapachote, P., Karuppiah, D.R., Hanson, A.R.: Feature Selection using Adaboost for Face Expression Recognition. In: Proc. Conf. on Visualisation, Imaging and Image Processing, Marbella, Spain, September 2004, pp. 84–89 (2004)
20. Fukunaga, K.: Introduction to statistical pattern recognition. Academic Press, London (1990)
21. Heijden, F., Duin, R.P.W., Ridder, D., Tax, D.M.J.: Classification, Parameter Estimation and State Estimation. Wiley, Chichester (2004)
22. Prechelt, L.: Proben1: A set of neural network Benchmark Problems and Benchmarking Rules, Tech Report 21/94, Univ. Karlsruhe, Germany (1994)
23. Merz, C.J., Murphy, P.M.: UCI repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
24. Fasel, B., Luettin, J.: Automatic facial expression analysis: a survey. Pattern Recognition 36, 259–275 (2003)
25. Tian, Y., Kanade, T., Cohn, J.F.: Recognising action units for facial expression analysis. IEEE Trans. PAMI 23(2), 97–115 (2001)
26. Kanade, T., Cohn, J.F., Tian, Y.: Comprehenive Database for facial expression analysis. In: Proc. 4th Int. Conf. automatic face and gesture recognition, Grenoble, France, pp. 46–53 (2000)
27. Dietterich, T.G.: Approx. statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 1895–1923 (1998)
28. Valentini, G., Dietterich, T.G.: Bias-Variance Analysis for Development of SVM-Based Ensemble Methods. Journal of Machine Learning Research 4, 725–775 (2004)
29. Windeatt, T., Ghaderi, R.: Coding and Decoding Strategies for Multi-class Learning Problems. Information Fusion 4(1), 11–21 (2003)