Ensemble Approaches to Facial Action Unit Classification

Terry Windeatt, 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. Facial action unit (*au*) classification is an approach to face expression recognition that decouples the recognition of expression from individual actions. In this paper, upper face *aus* are classified using an ensemble of MLP (Multi-layer perceptron) base classifiers with feature ranking based on PCA components. This approach is compared experimentally with other popular feature-ranking methods applied to Gabor features. Experimental results on Cohn-Kanade database demonstrate that the MLP ensemble is relatively insensitive to the feature-ranking method but optimized PCA features achieve lowest error rate. When posed as a multi-class problem using Error-Correcting-Output-Coding (ECOC), error rates are comparable to two-class problems (one-versus-rest) when the number of features and base classifier are optimized.

Keywords: Ensembles, ECOC, FACS, Feature-ranking

1 Introduction

Face expression recognition has potential application in many areas including human-computer interaction, talking heads, image retrieval, virtual reality, human emotion analysis, face animation, biometric authentication. 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) [1], the problem is decomposed into facial action units, including six upper face *aus* around the eyes (e.g. *au1* inner brow raised).

There are various approaches to determining features for discriminating between *aus*. Originally, features were based on measuring parts of the face that were involved in the *au* of interest [1]. However, it was found that comparable or better results could be obtained by a holistic approach that represents a more general approach to extracting features, such as Gabor wavelets [2]. The difficulty with methods like PCA and Gabor is the large number of features, and a method of eliminating irrelevant features is required. In this paper various feature-ranking schemes are compared, and the Out-of-Bag error estimate is used to optimise the number of features. In previous work [3] [14], it was shown that ensemble performance over seven benchmark problems is relatively insensitive to the feature-ranking method with simple one-dimensional performing at least as well as multi-dimensional schemes. In this paper, the main contribution is to show that PCA features outperform Gabor when using an ensemble. Furthermore, he Error-Correcting Output Coding (ECOC) method is applied to the problem of detecting combinations of *aus*.

2 Ensembles, Bootstrapping and ECOC

We assume a simple parallel Multiple Classifier System (MCS) architecture with homogenous MLP base classifiers, and for 2-class problems the combining rule is majority vote. 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 [4]. 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, but for tuning purposes the estimate of the absolute value is not important [5].

Error-Correcting Output Coding (ECOC) is a well-established method [6] [7] for solving multi-class problems by decomposition into complementary two-class problems. It is a two-stage process, coding followed by decoding. The coding step is defined by the binary *k x B* 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 labeling. Assuming each element of Z is a binary variable z, a training pattern with target class ω_l *(l = 1... k)* is re-labeled as class Ω_1 if $Z_{ij} = z$ and as class Ω_2 if $Z_{ij} = \bar{z}$. 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 $[0\ 1\ 0\ 0\ 1]^T$, this would naturally be interpreted as patterns from class 2 and 5 being assigned to Ω_1 with remaining patterns assigned to Ω_2. This is in contrast to the conventional One-per-class (OPC) code, which can be defined by the diagonal *k x k* code matrix $\{Z_{ij} = 1$ if and only if i = j$\}$.

In the test phase, the *jth* classifier produces an estimated probability $\hat{q}_j$ that a test pattern comes from the super-class defined by the *jth* decomposition. The *pth* test pattern is assigned to the class that is represented by the closest code word, where distance of the *pth* pattern to the *ith* code word is defined as

$$D_{pi} = \sum_{j=1}^{B} \alpha_{jl} \left| Z_{ij} - \hat{q}_{pj} \right| \qquad l = 1,...k \tag{1}$$

where α_{jl} allows for *lth* class and *jth* classifier to be assigned a different weight. If $\alpha=1$ in (1), Hamming decoding uses hard decision and L^1 norm decoding uses soft decision. Many types of decoding are possible, but theoretical and experimental evidence indicates that, providing a problem-independent code is long enough and base classifier is powerful enough, performance is not much affected. In this paper, a random code is used with *B=200 and k=12*, which is shown to perform almost as well as a pre-defined code, optimised for its error-correcting properties [7]. In Section 4, weighted coding uses Adaboost logarithmic formula to set the weights α in equation (1) [8].

To obtain the OOB estimate, the *pth* pattern is classified using only those classifiers that are in the set OOB_m, defined as the set of classifiers for which is the *pth* pattern is OOB. For the OOB estimate, the summation in equation (1) is therefore modified to $\sum_{j \in OOB_m}$. In other words columns of Z are removed if they correspond to classifiers that used the *pth* pattern for training.

In the experiments Section 4, random perturbation of the MLP base classifiers is caused by different starting weights on each run, combined with bootstrapped training patterns. The MLP ensemble uses two hundred single hidden-layer MLP base classifiers, with Levenberg-Marquardt training algorithm and default parameters. In our framework, we vary the number of hidden nodes, with a single node for linear perceptron.

3 Feature-ranking

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 [9]. Although feature-ranking has received much attention in the literature, there has been relatively little work devoted to handling feature-ranking explicitly in the 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, for which 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 [9], forward feature and random (without replacement) selection methods are used to sequentially determine disjoint optimal subsets. In [10], 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 [11].

The following five feature-ranking methods are used in this paper

1) rfenn, ***rfesvc*** In [**12**], a local feature selection gain w_i is given by

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

where *i*,*j* are the input and hidden node indices of an MLP classifier, x_i is input feature, W^1 is the first layer weight matrix and W^2 is the output weight vector.

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.

2) rfenb Fisher's criterion ***J(w)*** measures the separation between two sets of patterns in a direction ***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(w)*** is maximized. The idea behind the *noisy bootstrap* [13] 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. For further details see [14]

3) boost Boosting is a well-known algorithm 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 [15]. 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 [16]. In our implementation, we use Adaboost [8] with decision stump as weak learner.

4)1dim Class separability measures are popular for feature-ranking, and many definitions use S_B and S_W. 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.

5) sffs A fast multi-dimensional search method that has been shown to give good results with individual classifiers is Sequential Floating Forward Search. 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 [17] for our comparative study.

RFE is a simple algorithm [18], 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 [19]. Therefore RFE boosts performance of simple feature ranking strategies, and in this paper we use RFE with MLP weights (***rfenn***), SVC weights (**rfesvc**), and noisy bootstrap (***rfenb***).

4 Dataset and Experimental Evidence

The database we use is Cohn-Kanade [20], 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 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(inner brow raised), au2(outer brow raised), au4(brow lowered), au5(upper eyelid raised), au6(cheek raised),* and *au7(lower eyelid tightened).* 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) PCA applied either to raw image or after filtering with forty Gabor filters [15], five special frequencies at five orientations with top 4 principle components for each Gabor filter, giving 160-dimensional feature vector
g) PCA ordering or feature ranking schemes described in Section 3
h) MLP ensemble and Support Vector Classifier
i) Random training/test split of 90/10 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 for *au1* by a simplistic classifier that classifies all images as *non-au1*. For the reason we report area under ROC curve, similar to [21].

In the first set of experiments, *au1* classification with Gabor features is compared for the different feature ranking schemes described in Section 3. Table 1 shows feature-ranking comparison for *au1* classification for MLP ensemble and linear SVC classifier. It may be seen that the ensemble is fairly insensitive to the ranking scheme, and the more sophisticated schemes of SFFS and Boosting are slightly worse on average than the simpler schemes. It was found that lower test error was obtained with non-linear base classifier having 16 nodes and 20 epochs. The minimum base error rate is 16.5% achieved for 28 features, while the ensemble is 10.0% at 28 features. By comparison the linear SVC achieves slightly worse results on average, although the differences were found not to be statistically significant (McNemar 5%). We did not try different SVC kernels with varying regularization constant since tuning was difficult [13].

The second set of experiments detects *au1, au2, au4, au5, au6, au7* using six different 2-class classification problems, where the second class contains all patterns not containing respective *au*. Figure 1 shows *au1* classification train/test error rates and ensemble area under ROC for MLP ensemble as number of PCA features is reduced. The best error rate of 8% was obtained with 16 nodes and 36 features, which is an improvement of 2% over the best result in Table 1. It is believed that the overall ensemble rate of 8% is among the best for *au1* on this database (recognising the difficulty of making fair comparison). The 8% error rate for *au1* is equivalent to 73% of *au1s* 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 overall error rate. The best ensemble error rate, number of features and number of nodes for all upper face *aus* are shown in the first two columns of

Table 3. Note that number of nodes for best area under ROC is generally higher than for best error rate, indicating that error rate is more likely to be susceptible to over-fitting.

The third set of experiments uses ECOC method described in Section 2. The ultimate goal in *au* classification is to detect combination of aus. In the ECOC approach, a random 200x12 code matrix Z is used to consider each *au* combination as a different class. After removing classes with less than four patterns this gives a 12-class problem with *au* combinations as shown in Table 2. To compare the results with 2-class classification , we compute test error by interpreting super-classes as 2-class problems, defined as either containing or not containing respective *au*. For example, sc2, sc3, sc6, sc11, sc12 in Table 2 are interpreted as *au1*, and remaining super-classes as *non-au1*. The last two columns of Table 3 show ECOC classification error and area under ROC. It may be seen that 2-class classification with optimized PCA features on average slightly outperforms ECOC. However, the advantage of ECOC is that all problems are solved simultaneously with 200 classifiers, and furthermore the combination of *aus* is recognized. As a 12-class problem, the mean best error rate over the twelve classes defined in Table 2 is 38.2 %, achieved at 60 features with 1 node, showing that recognition of combination of *aus* is a difficult problem.

5 Conclusion

For a bootstrapped MLP ensemble, the OOB estimate may be used to tune classifier parameters and to determine when to stop eliminating features. For *au1* classification, PCA features outperform Gabor, and for upper face *aus* optimized 2-class classifiers give slightly lower mean error rates than ECOC. However, ECOC can detect combinations of *aus* and further work is aimed at determining whether problem-dependent rather than random ECOC codes can give better results.

Table 1: Mean best error rates (%)/number of Gabor 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

Table 2: ECOC super-classes of action units and number of patterns

ID	**sc1**	**sc2**	**sc3**	**sc4**	**sc5**	**sc6**	**sc7**	**sc8**	**sc9**	**sc10**	**sc11**	**sc12**
superclass	{}	1,2	1,2,5	4	6	1,4	1,4,7	4,7	4,6,7	6,7	1	1,2,4
#patterns	149	21	44	26	64	18	10	39	16	7	6	4

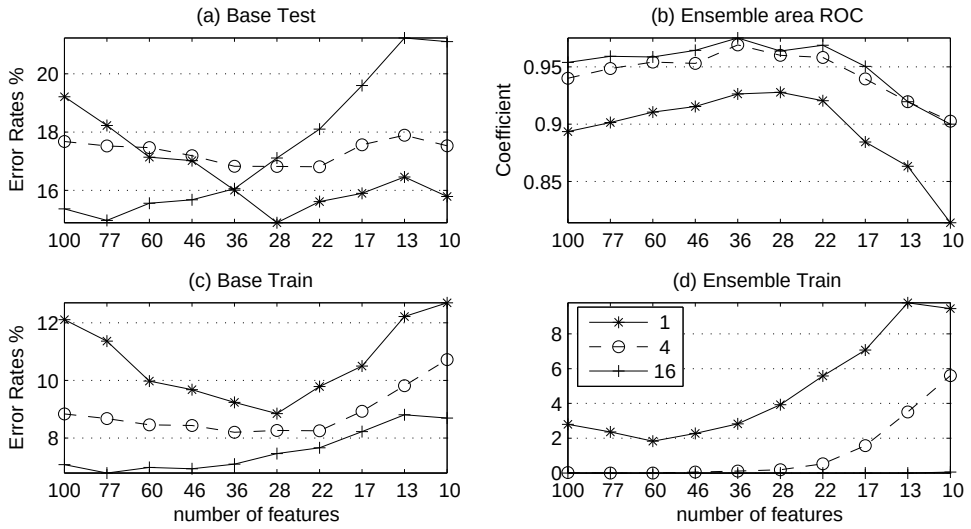

Figure 1: Train and test error rates, Ensemble area under ROC for RFE MLP ensemble, au1 classification, [1,4,16] hidden nodes 20 epochs

Table 3: Mean best error rates (%) and area under ROC showing #nodes /#features for au classification 90/10 with optimized PCA features and MLP ensemble

	2-class Error %	**2-class ROC**	**ECOC Error %**	**ECOC ROC**
au1	8.0/16/28	0.97/16/36	9.0/4/36	0.94/4/17
au2	2.9/1/22	0.99/16/36	3.2/16/22	0.97/1/46
au4	8.5/16/36	0.95//16/28	9.0/1/28	0.95/4/36
au5	5.5/1/46	0.97/1/46	3.5/1/36	0.98/1/36
au6	10.3/4/36	0.94/4/28	12.5/4/28	0.92/1/28
au7	10.3/1/28	0.92/16/60	11.6/4/46	0.92/1/36
mean	7.6	0.96	8.1	0.95

References

1 Tian Y., Kanade T., Cohn J. F: Recognising action units for facial expression analysis, IEEE Trans. PAMI 23(2), 97-115, (2001).

2 Donato G., Bartlett M. S., Hager J. C., Ekman P, Sejnowski T. J.: Classifying facial actions, IEEE Trans. PAMI 21(10), 974-989, (1999).
3 Windeatt T., Dias K.: Feature-ranking ensembles for facial action unit classification, IAPR Third Int. Workshop on artificial neural networks in pattern recognition, Paris,, accepted (2008).
4 Windeatt T., Prior M: Stopping Criteria for Ensemble-based Feature Selection, LNCS vol. 4472, pp. 271-281, Springer, Heidelberg (2007).
5 Windeatt T.: Accuracy/Diversity and Ensemble Classifier Design, IEEE Trans. Neural Networks 17(5), 287-297, (2006).
6 Dietterich T. G., Bakiri G.: Solving multiclass learning problems via error-correcting output codes, J. Artificial Intelligence Research 2,, 263-286, (1995).
7 Windeatt T., Ghaderi R.: Coding and Decoding Strategies for multiclass learning problems, Information Fusion, 4(1), 11-21, (2003).
8 Freund Y., Schapire R.E.: A decision-theoretic generalisation of on-line learning and an application to boosting, J. of Computer and System Science, 55(1), 119-139, (1997).
9 Skuruchina M., Duin R. P. W.: Combining feature subsets in feature selection, Proc. 6th Int. Workshop Multiple Classifier Systems, LNCS vol. 3541, pp. 165-174, Springer, Heidelberg (2005).
10 Oza N.,, Tumer K.: Input Decimation ensembles: decorrelation through dimensionality reduction, Proc. 2nd Int. Workshop Multiple Classifier Systems, LNCS vol. 2096, pp. 238-247, Springer, Heidelberg (2001).
11 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).
12 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).
13 Efron N. and Intrator N., The effect of noisy bootstrapping on the robustness of supervised classification of gene expression data, IEEE Int. Workshop on Machine Learning for Signal Processing, Brazil, pp. 411-420, (2004).
14 Windeatt T., Prior M., Effron N., Intrator N.: Ensemble-based Feature Selection Criteria, Proc. Conference on Machine Learning Data Mining MLDM2007, Leipzig, ISBN 978-3-940501-00-4, pp 168-182, (2007).
15 Bartlett, M.S. Littlewort, G. Lainscsek, C. Fasel, I. Movellan : J. Machine learning methods for fully automatic recognition of facial expressions and facial actions, IEEE Conf. System, Man and Cybernetics, Vol. 1, 592- 597, (2004).
16 Silapachote P., Karuppiah D. R., Hanson A. R.: Feature Selection using Adaboost for Face Expression Recognition, Proc. Conf. on Visualisation, Imaging and Image Processing, Marbella, Spain, pp. 84-89, (2004).
17 Heijden F., Duin R. P.W., Ridder D., Tax D. M .J: Classification, Parameter Estimation and State Estimation, Wiley, (2004).
18 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).
19 Yu L. and Liu H.: Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5, 1205-1224, (2004).
20 Kanade T.. Cohn J. F., Tian Y.: Comprehensive Database for facial expression analysis, Proc. 4th Int. Conf. automatic face and gesture recognition, Grenoble, France, pp. 46-53, (2000).
21 Bartlett M. S., Littlewort G., Frank M., Lainscsek C., Fasel I.., Movellan J.: Fully automatic facial action recognition in spontaneous behavior, Proc 7th Conf. On Automatic Face and Gesture Recognition, ISBN 0-7695-2503-2, pp. 223-238, (2006).