Weighted Decoding ECOC for Facial Action Unit Classification

Terry Windeatt, Raymond S. Smith and Kaushala Dias [1]

Abstract. There are two approaches to automating the task of facial expression recognition, 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 as in FACS (Facial Action Coding System). In this paper, upper face action units (*aus*) are classified using an ensemble of MLP base classifiers with feature ranking based on PCA components. When posed as a multi-class problem using Error-Correcting-Output-Coding (ECOC), experimental results on Cohn-Kanade database demonstrate that error rates comparable to two-class problems (one-versus-rest) may be obtained. Weighted decoding is shown to outperform conventional ECOC decoding. The error rates obtained for six upper face *aus* around the eyes are believed to be among the best for this database.

1 INTRODUCTION

The problem of face expression recognition is difficult because facial expression depends on age, ethnicity, gender, occlusions as well as pose and lighting variation. Facial action unit (*au*) classification is an approach to face expression recognition that decouples the recognition of expression from individual actions. In FACS (facial action coding system) [1] the problem is decomposed into facial action units, that includes six upper face *aus* around the eyes. It has the potential of being applied to a much richer set of applications than an approach that targets facial expression directly. However, the coding process requires skilled practitioners and is time-consuming so that typically there are a limited number of training patterns.

There are various approaches to determining features for discriminating between *aus*. Originally, features were based on geometric measurements of the face that were involved in the *au* of interest [1]. More recently, holistic approaches based on PCA, Gabor [2] and Haar wavelets represent a more general approach to extracting features [3], and have been shown to give comparable results. The difficulty with these latter approaches is the large number of features. When combined with the limited number of patterns, this can lead to the small sample-size problem, that is when the number of patterns is less than or comparable to the number of features. A method of eliminating irrelevant features is therefore required [4] [5]. In this paper the Out-of-Bag error estimate is used to optimise the number of features.

In previous work [6] [7] five feature ranking schemes were compared using Gabor features in an MLP ensemble. The schemes were Recursive Feature Elimination (RFE) [9] combined with MLP weights and noisy bootstrap, boosting (single feature selected each round), one-dimensional class-separability measure and Sequential Floating Forward Search (SFFS). It was shown that ensemble performance is relatively insensitive to the feature-ranking method with simple one-dimensional performing at least as well as multi-dimensional schemes. It was also shown that the ensemble using PCA features with its own inherent ranking outperformed Gabor.

In this paper, PCA features are used with Error-Correcting Output Coding (ECOC) and a weighted decoding strategy based on bootstrapping individual base classifiers is proposed. The principle behind weighted decoding is to reward classifiers that perform well. The weights in this study are fixed in the sense that none change as a function of the particular pattern being classified. Sometimes this is referred to as implicit data-dependence or constant weighting. It is generally recognized that a weighed combination may in principle be superior, but it is not easy to estimate the weights.

The main contribution in this paper is to apply a weighted ECOC decoding strategy to the problem of facial action unit classification. Section 2 discusses ensemble techniques, Bootstrapping and ECOC for weighted decoding. Section 3 describes the database and design decisions for *au* classification, and compares 2-class classification with weighted and conventional ECOC decoding.

2 ENSEMBLES, BOOTSTRAPPING AND ECOC

We assume a simple parallel Multiple Classifier System (MCS) architecture with homogenous MLP base classifiers. 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

[1] University of Surrey, UK email: t.windeatt@surrey.ac.uk

furthermore, the OOB is a good estimator of when to stop eliminating features [8]. 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.

Error-Correcting Output Coding (ECOC) is a well-established method [10] [11] 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-versus-rest code, which can be defined by the diagonal *k x k* code matrix

Many types of coding 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 with near equal split of labels in each column is used with *B=200 and k=12*. It has been shown theoretically and experimentally that a long random code performs almost as well as a pre-defined code, optimised for its error-correcting properties [11].

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

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

where α_{jl} allows for *lth* class and *jth* classifier to be assigned a different weight. If $\alpha=1$ in equ. (1), Hamming decoding uses hard decision and L^1 *norm* decoding uses soft decision.

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 the *pth* pattern is OOB. For the OOB estimate, the summation in equ. (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 this paper we introduce a different weighted decoding scheme, that treats the outputs of the base classifiers as binary features. By using the diagonal matrix $\{Z_{ij} = 1$ if and only if i = j$\}$ the problem is recoded as *k* 2-class problems where each problem is defined by a different binary-to-binary mapping. There are many strategies that may be used to learn this mapping, but we use a weighted vote with weights set by class-separability measure applied to the training data, defined in [12].

Let y_{mj} indicate the binary output of the *jth* classifier applied to the *mth* training pattern, so that the output of base classifiers for the *mth* pattern is given by

$$y_{mj} = (y_{m1}, y_{m2} \cdots y_{mB}) \qquad (2)$$

Assuming in equ. (2) that a value of 1 indicates agreement of the output with target label and 0 disagreement, we can define counts for *jth* classifier as follows

$$N_j^{11} = y_{mj} \wedge y_{nj} \text{ and } N_j^{00} = \bar{y}_{mj} \wedge \bar{y}_{nj} \qquad (3)$$

where the *mth* and *nth* pattern are chosen from different classes.

The weight for the *jth* output is then defined as

$$w_j = \frac{1}{K}\left(\sum_{allpairs} N_j^{11} - \sum_{allpairs} N_j^{00} \right) \qquad (4)$$

where K is a normalization constant and the summation is over all pairs of patterns from different class.

The motivation behind equ. (4) is that the weight is computed as the difference between positive and negative correlation with respect to target class. In [12] this is shown to be a measure of class separability.

3 DATASET & EXPERIMENTAL EVIDENCE

The Cohn-Kanade database [13] contains posed 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 1) subset of image sequences chosen from the database 2) whether or not the neutral image is included in training 3)image resolution 4)normalisation procedure 5)size of window extracted from the image, if at all 6) features chosen for discrimination. Furthermore classifier type/parameters, and training/testing protocol need to be chosen. Researchers choose 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)*. We chose an MLP ensemble and random training/test split of 90/10 repeated twenty times and averaged. Other design decisions we made were:

1) All image sequences of size 640 x 480 chosen
2) Last image in sequence (no neutral) chosen giving 424 images, 115 containing *au1*
3) Full image resolution, no compression
4) Manually located eye centres plus rotation/scaling into 2 common eye coordinates
5) Window extracted of size 150 x 75 pixels centred on eye coordinates
6) PCA applied to raw image with PCA ordering

With reference to 2), 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 4) 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 misleading 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-au*. For the reason we report area under ROC curve, similar to [5].

There are two sets of experiments aimed at 2-class and multi-class formulations of *au* classification. In both sets of experiments, the MLP ensemble uses two hundred single hidden-layer MLP base classifiers, with Levenberg-Marquardt training algorithm and default parameters. Random perturbation of the MLP base classifiers is caused by different starting weights on each run, combined with bootstrapped training patterns. In our framework, we vary the number of hidden nodes, with a single node for linear perceptron, and keep the number of training epochs fixed at 20.

The ultimate goal in *au* classification is to detect combination of *aus*. In the ECOC approach, a random *200x12* code matrix 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 1. 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 1 are interpreted as *au1*, and remaining super-classes as *non-au1*

The first 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*. The MLP ensemble uses majority vote combining rule. The best error rate of 9.4% for *au1* was obtained with 16 nodes and 28 features. The 9.4% 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, area under ROC with number of features and number of nodes for all upper face *aus* are shown in the first two columns of Table 2. 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 second set of experiments uses ECOC method described in Section 2, and figure 1 shows area under ROC for the six *aus*, as number of PCA features is reduced. Columns 3 and 4 in Table 2 show best L^1 *norm* decoding classification error and area under ROC, while last 2 columns show respective weighted decoding. It may be seen that weighted outperforms L^1 *norm* decoding. Also it may be seen from Table 2 that 2-class classification with optimized PCA features (columns 1 and 2) 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 1 is 38.2 %, showing that recognition of combination of *aus* is a difficult problem.

4 DISCUSSION

The results for upper face *aus*, shown in Table 2, are believed to be among the best on this database (recognising the difficulty of making fair comparison as explained in Section 3).There are two possible reasons why the ECOC decoding strategy works well. Firstly, the data is projected into a high-dimensional space and therefore more likely to be linearly separable [14]. Secondly, although the full training set is used to estimate the weights, each base classifier is bootstrapped and therefore is trained on a subset of the data, which guards against over-fitting. As indicated in Section 2, bootstrapping also facilitates the OOB estimate for removing irrelevant features without validation. The effect of bootstrapping can be understood using bias/variance of 0/1 loss

function [15]. In [6] it is shown that a bootstrapped ensemble benefits from reduced bias at the expense of increased variance.

Some preliminary results on other techniques to learn the binary-to-binary mappings defined in Section 2, indicate that the decoding strategy is fairly insensitive to the method of setting the weights. For example, similar results were obtained by using Adaboost logarithmic formula [16].

5 CONCLUSION

For upper face *au* classification, weighted decoding ECOC achieves comparable performance to optimized 2-class classifiers. However, ECOC has the advantage that all *aus* are detected simultaneously, and further work is aimed at determining whether problem-dependent rather than random codes can improve results.

References

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

[2] G Donato, M S Bartlett, J C Hager, P Ekman and T J Sejnowski, Classifying facial actions, IEEE Trans. PAMI 21(10), 1999, 974-989.

[3] 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. **Systems, Man and Cybernetics**, Oct 2004, Vol. 1, 592- 597.

[4] 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, 84-89.

[5] M S Bartlett, G Littlewort, M Frank, C Lainscsek, I Fasel and J Movellan, Fully automatic facial action recognition in spontaneous behavior, Proc 7th Conf. On Automatic Face and Gesture Recognition, 2006, ISBN 0-7695-2503-2, 223-238.

[6] T Windeatt, K Dias, Feature-ranking ensembles for facial action unit classification, IAPR Third Int. Workshop on artificial neural networks in pattern recognition, Paris, July, 2008, accepted.

[7] T. Windeatt., M. Prior, N. Effron, N. Intrator, Ensemble-based Feature Selection Criteria, Proc. Conference on Machine Learning Data Mining MLDM2007, Leipzig, July 2007, ISBN 978-3-940501-00-4, pp 168-182

[8] T Windeatt, M Prior, Stopping Criteria for Ensemble-based Feature Selection, Proc. 7th Int. Workshop Multiple Classifier Systems, Prague May 2007, Lecture notes in computer science, Springer-Verlag, 271-281

[9] Guyon I., Weston J., Barnhill S. and Vapnik V., Gene selection for cancer classification using support vector machines, Machine Learning 46(1-3), 2002, 389-422.

[10] T. G. Dietterich ,G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artificial Intelligence Research 2, 1995, 263-286

[11] T Windeatt and R Ghaderi, Coding and Decoding Strategies for multiclass learning problems, Information Fusion, 4(1), 2003, 11-21.

[12] T Windeatt, Accuracy/Diversity and Ensemble Classifier Design, IEEE Trans. Neural Networks 17(5), 2006, 287-297.

[13] T. Kanade, J. F. Cohn and Y. Tian, Comprehensive Database for facial expression analysis, Proc. 4th Int. Conf. automatic face and gesture recognition, Grenoble, France, 2000, 46-53.

[14] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Trans. Information Theory, vol. EC-14, 1965, 326-334.

[15] G. Valentini, T. G. Dietterich, Bias-variance analysis of Support Vector Machines for the development of SVM-based ensemble methods, *Journal of Machine Learning Research*, 5 , 2004, MIT Press, 725-775.

[16] Y. Freund and R.E. Schapire. A decision-theoretic generalisation of on-line learning and an application to boosting, J. of Computer and System Science, 55(1), 1997, 119-139.

Table 1. 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

Table 2: Mean best test error rates (%) and area under ROC showing #nodes/#features for *au* classification with optimized PCA features and MLP ensemble

	2-class Test Error %	***2-class area under ROC***	***ECOC Test Error %***	***ECOC area under ROC***	***ECOC Weighted Error %***	***ECOC Weighted ROC***
au1	9.4/16/28	0.97/16/36	10.3/1/10	0.92/16/46	9.2/4/36	0.94/16/36
au2	3.5/4/36	0.99/16/22	3.4/1/36	0.96/16/28	2.8/16/22	0.98/1/46
au4	9.1/16/36	0.95/16/46	12.0/16/28	0.92/4/28	9.5/1/28	0.94/4/28
au5	5.5/1/46	0.97/1/46	3.6/16/36	0.99/1/36	3.2/1/36	0.99/1/36
au6	10.5/1/36	0.94/4/28	13.1/1/77	0.88/1/77	12.8/1/77	0.90/1/28
au7	10.3/1/28	0.92/16/60	11.6/1/28	0.89/4/46	10.9/4/46	0.92/1/36
mean	**8.1**	**0.96**	**9.0**	**0.93**	**8.1**	**0.95**

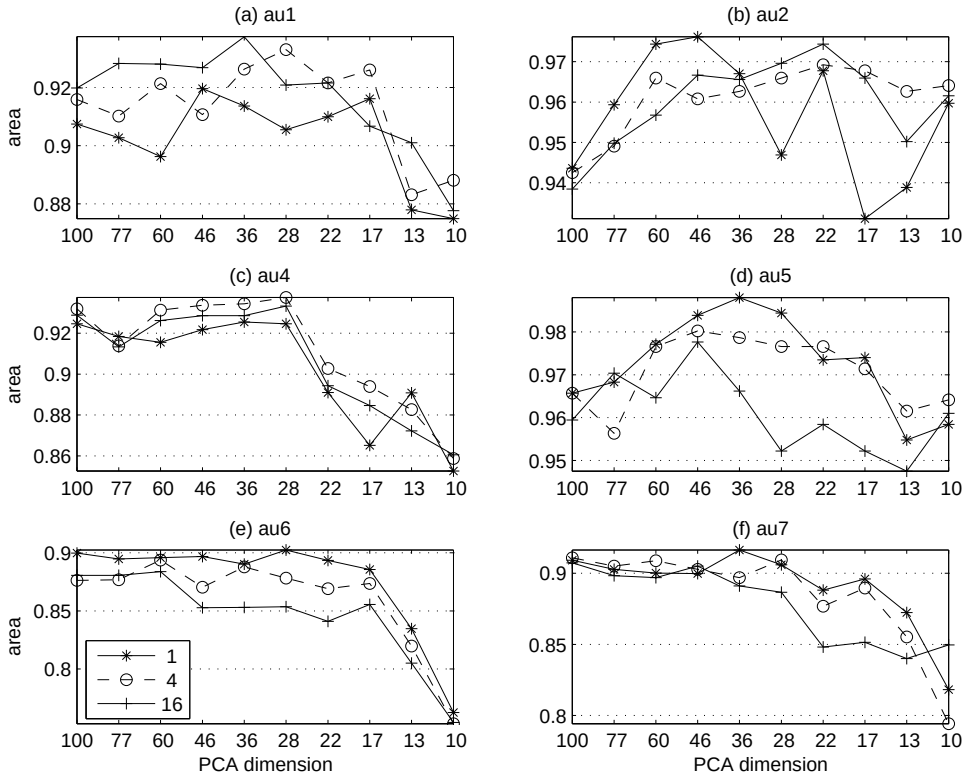

Figure 1: Area under ROC for weighted decoding ECOC MLP ensemble [1,4,16] hidden nodes 20 epochs versus number PCA features (logscale)