Non-Linear Approaches for the Classification of Facial Expressions at Varying Degrees of Intensity

Jane Reilly, John Ghent, John McDonald
Computer Vision and Imaging Laboratory
Department of Computer Science
National University of Ireland Maynooth
{jreilly, jghent, johnmcd} @cs.nuim.ie

Abstract

The research discussed in this paper documents a comparative analysis of two nonlinear dimensionality reduction techniques for the classification of facial expressions at varying degrees of intensity. These nonlinear dimensionality reduction techniques are Kernel Principal Component Analysis (KPCA) and Locally Linear Embedding (LLE). The approaches presented in this paper employ psychological tools, computer vision techniques and machine learning algorithms. In this paper we concentrate on comparing the performance of these two techniques when combined with Support Vector Machines (SVMs) at the task of classifying facial expressions across the full expression intensity range from near-neutral to extreme facial expression. Receiver Operating Characteristic (ROC) curve analysis is employed as a means of comprehensively comparing the results of these techniques.

1. Introduction

Facial expressions play a major role in how people communicate. Therefore a computer that could interpret and react to facial expression would advance human-computer interfaces, providing a basis for communication that could be compared to human-human interaction. As the complexity of computer applications increase in tandem with user expectations, the development of emotionally intelligent interfaces has become both necessary and possible.

This paper builds on preliminary results reported in [12] and [21], where two nonlinear dimensionality reduction techniques were examined in the context of facial expression classification. In those initial investigations we looked at the performance of Kernel Principal Component Analysis (KPCA) and Locally Linear Embedding (LLE) in isolation on a small dataset, while in this paper we perform a comprehensive comparative analysis of these techniques on a more substantial dataset. While previously we concentrated on the number of correct classifications, in this paper we extend this work to examine the performance of our techniques at classifying facial expressions at varying degrees of intensity using Receiver Operating Characteristic (ROC) analysis. By performing ROC analysis we not only look at the number of correct classifications, but we also consider the number of incorrect classifications when deciding which technique performs best (see section 2.5 for more details on ROC analysis).

Since the importance of facial expressions was first established in 1872 [6], many studies have been carried out attempting to interpret their meaning. In more recent years a considerable amount of research has been performed investigating various methodologies for the classification of facial expressions. Techniques range from template based methods [18], to neural network based methods [8], or a combination of the two [20, 11]. Perhaps the most substantial work in this area has been done by Bartlett et al [1]. Bartlett et al. proposed a technique which combines Gabor wavelets and SVMs to classify AUs with 93.3% accuracy [1]. Again in [16], Littlewort and Bartlett propose a similar technique which classifies AUs with 97% accuracy. In [17], Bartlett used SVMs again to successfully distinguish between genuine and fake smiles.

One of the techniques which we appraise in this paper, KPCA is a non-linear dimensionality reduction technique which is closely related to methods applied to SVMs [28, 27]. The principal idea behind KPCA is to transform the input data into a potentially infinite high dimensional space. This is achieved by utilizing the kernel trick to perform feature space operations using dot-products between data points in the feature space (for more information see [10]).

KPCA and more prominently kernel methods have been used extensively in recent years [2]. Several methods have
been proposed for facial expression analysis [31, 30]. More specifically KPCA has been applied successfully for the purpose of de-noising images of hand-written digits [19], and as a pre-processing step in regression problems [26]. On a related note to this paper, KPCA has also been used to analyse facial shape [30]. Here, we use KPCA to classify lower facial expressions by looking at the change in shape of the mouth as an expression forms.

In this paper we also discuss the effectiveness of LLE in conjunction with SVMs to classify expressions. LLE was introduced in 2000 as a non-linear dimensionality reduction technique that computes low dimensional neighborhood preserving embeddings of high dimensional data by unfolding the underlying manifold [24, 25].

The remainder of this paper is structured as follows: Section 2 documents our approach, Section 3 details experiments and results, and in Section 4 we conclude with some final remarks. This paper extends previous work detailed in [10, 21] and [12].

2. Proposed Methodology

Measuring facial expressions at varying expression intensity is a non trivial task as everyone’s face is unique and interpersonal differences exist in how people perform facial expressions. Although numerous methodologies have been proposed to solve this problem, the main focus of our research is to measure facial expressions independent of identity and with varying degrees of intensity in a consistent and robust manner. In this section we discuss the techniques which we use to accurately classify lower facial expressions, namely; the facial action coding system, kernel principal component analysis, locally linear embedding, support vector machines and receiver operating characteristic curve analysis.

2.1. Facial Action Coding System - FACS

In this paper we use the Facial Action Coding System (FACS), which measures facial expressions according to the movement of muscles in the face. The system is based on an anatomical analysis of facial expressions [7]. The FACS allows us to subdivide our data into subsets where the variation in each expression is precisely characterized. The FACS provides an unambiguous, quantitative means of describing all movements of the face in terms of Action Units (AUs). An AU is composed of one or more muscles in the face that causes an atomic change in the faces appearance. All expressions can be described using the AUs defined by the FACS, providing a measurable set of criteria that define whether or not a particular facial expression is present.

However AUs do not always occur with the same expression intensity and for this reason the FACS also includes intensity ranges for the AUs. There are five intensities in total ranging from intensity A, where a trace change in appearance occurs, to intensity E, where an extreme or maximum appearance change occurs. Figure 1 provides a visual representation of the full FACS intensity range, which can be thought of as the spectrum of intensity for each AU. Note that this scale is divided non-uniformly - e.g. C takes up more of the spectrum than A [7].

![Figure 1. The scale of FACS intensity scores](image)

2.2. Kernel Principal Component Analysis - KPCA

Principal Component Analysis (PCA, also known as the Karhunen-Loève transform) is a technique used to lower the dimensionality of a feature space [29, 13]. Kernel Principal Component Analysis (KPCA) is similar to standard PCA except the data is projected into a higher dimensional feature space prior to performing eigenvector decomposition. We project the data into feature space through the use of the kernel trick. This kernel trick permits the computation of dot products in high dimensional feature spaces, using functions defined on pairs of input patterns.

More specifically, mapping from one space to a higher dimensional space involves a mapping from $x_i \rightarrow \phi(x_i)$, however, with an appropriate choice of kernel there exists a mapping $\phi$ such that $(\phi(x_i) \cdot \phi(x_j)) = K(x_i, x_j)$. This means that the inner products of the feature space can be calculated without computing $\phi(x)$ directly. This allows us to work in a potentially infinite high dimensional feature space. The choice of kernel is still a matter of debate, however, in this paper we use a Gaussian kernel. The Gaussian kernel is defined as

$$K(x_i, x_j) = e^{-\frac{(x_i-x_j)^T(x_i-x_j)}{2\sigma^2}}. \quad (1)$$

Where $\sigma$ determines the width of the kernel.

2.3. Local Linear Embedding - LLE

The LLE algorithm was introduced by Saul and Roweis in 2000 as an unsupervised learning algorithm that computes low dimensional, neighborhood preserving embeddings of high dimensional data [24, 25]. Many extensions and adjustments to this core algorithm have been proposed, ranging from Robust-LLE [3, 14] to supervised [23, 15] and semi-supervised versions of LLE [22], however in this paper we are using the original algorithm as defined in [24, 25].
The LLE algorithm is based on simple geometric intuitions where the algorithm attempts to compute a low dimensional embedding with the property that nearby points in the high dimensional space remain nearby and similarly co-located with respect to one another in the low dimensional space. The LLE algorithm takes a dataset of \( N \) real valued vectors \( \mathbf{X}_i \), each of dimensionality \( D \), sampled from some smooth underlying manifold as its input. Provided there is sufficient data such that the manifold is well sampled, we can expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold [25].

There are three main steps in the LLE algorithm. First, the manifold is sampled and the \( K \) nearest neighbors per data point are identified. Second, each point \( \mathbf{X}_i \) is approximated as a linear combination of its neighbors \( \mathbf{X}_j \), these linear combinations are then used to construct the sparse weight matrix \( \mathbf{W}_{ij} \). Reconstruction errors are then measured by the cost function Equation 2, which adds up the squared distances between all the data points and their reconstructions.

\[
\sum \mathbf{W} = \sum_i |\mathbf{X}_i - \sum_j \mathbf{W}_{ij} \mathbf{X}_j|^2 \tag{2}
\]

In the final step of the LLE algorithm, each point \( \mathbf{X}_i \) in the high dimensional space is mapped to a point \( \mathbf{Y}_i \) in the low dimensional space which best preserves the sparse structure and geometry of \( \mathbf{X}_i \)’s neighborhood. This geometry and structure is represented by the weight matrix \( \mathbf{W}_{ij} \). The mapping from \( \mathbf{X}_i \) to \( \mathbf{Y}_i \) is achieved by fixing the weights \( \mathbf{W}_{ij} \) and selecting the bottom \( d \) non-zero coordinates of each output \( \mathbf{Y}_i \) to minimize Equation 3.

\[
\phi \mathbf{Y} = \sum_i |\mathbf{Y}_i - \sum_j \mathbf{W}_{ij} \mathbf{Y}_j|^2. \tag{3}
\]

### 2.4. Support Vector Machines - SVMs

SVMs are used to classify an expression using the outputs from both the KPCA model and the LLE model as inputs. The SVM algorithm can be separated into two distinct procedures, the kernel trick, which we have already discussed, and the base algorithm. Suppose we have a dataset \((x_1, y_1), \ldots, (x_m, y_m) \in \mathbf{X} \times \{\pm 1\}\) where \( \mathbf{X} \) is some space from which the \( x_i \) have been sampled. We can construct a dual Lagrangian of the form

\[
W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{4}
\]

which are subject to the constraints \( \alpha_i \geq 0 \forall i \) and \( \sum_{i=1}^m \alpha_i y_i = 0 \). The solution to Equation 4 is a set of \( \alpha \) values which defines a hyperplane that is positioned in an optimal location between the two classes. Further details of the construction of this equation can be found in [10].

### 2.5. Receiver Operating Characteristic (ROC) Curves

**Receiver Operating Characteristic (ROC) Curves**

ROC graphs are two dimensional graphs with the true positive rate plotted on the \( y \)-axis, and the false positive rate plotted on the \( x \)-axis. A ROC graph depicts relative trade-offs between the benefits and costs of a particular classifier. The most frequently used performance measure in ROC analysis is the area under the ROC curve, more commonly referred to as the AUC. The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. The AUC ranges from 0 - 1, however as the random classifier has an AUC of 0.5, no realistic classifier should have an AUC less than 0.5.
3. Experiments and Results

In our research to date we have used the Cohn-Kanade AU-Coded Facial Expression Database [4]. This database contains approximately 2000 image sequences from over 200 subjects. The subjects came from a cross-cultural background and were aged approximately 18 to 30. This database contains full AU coding and partial intensity coding of facial images and while it is not ideal in that it isn’t completely intensity coded, it is the most comprehensive database currently available.

In this paper we classify four lower facial expressions; AU20+25, AU25+27, AU10+20+25, and AU12. The effect that these have on the mouth is illustrated in Table 2. For a detailed description of each expression refer to [7].

Table 2. This table illustrates the effect of portraying four different expressions. The AUs portrayed are: I=AU20+25; II=AU25+27; III=AU10+20+25; IV=AU12

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our training dataset consisted of 73 images of one subject performing 4 sequences from neutral to extreme expression. Our test dataset consisted of 522 images of multiple subjects from multi-cultural backgrounds performing the four lower facial expressions as shown in Table 2. In our test dataset we sampled the sequences at each intensity rating including neutral, 6 in total, while in our training set we labeled each frame in the sequence with an intensity score. Prior to experimentation we manually placed 24 landmark points placed at 24 specific locations, shown in Figure 2. The optimal position of these landmark points is defined by Cootes in [5].

3.1. Locally Linear Embedding Results

To analyze the variance of the points that describe the shape of the mouth it is necessary that the mouth shapes in the training and test sets are as closely aligned as possible. Therefore, before applying the LLE and KPCA algorithms we align our data by firstly, centering the data and then performing Generalized Procrustes Alignment (GPA) [13]. GPA aligns two shapes with respect to position, rotation and scale by minimizing the weighted sum of the squared distances between the corresponding landmark points. More information on this technique can be found in [10].

Once GPA has been performed, we can either apply KPCA and LLE directly or we can further reduce the variance in the dataset by performing shape differencing, whereby we subtract the neutral shape from each subsequent shape in the sequence. The reason being is that we are interested in the difference between the two shapes and not the actual shapes themselves. Once this preprocessing has been completed we calculate our shape models by performing LLE and KPCA on the aligned training data, the outputs of which are used to train the SVM classifiers. New unseen data is projected into both shape spaces and the outputs are used as inputs to the previously trained SVM classifiers.

Figure 3 shows the LLE expression shape space created as a result of applying LLE to our training set without using shape differencing and Figure 4 displays the LLE shape space after performing the additional step of Shape Differencing. It is clear from these diagrams that LLE has successfully separated out the input dataset into the four lower facial expressions which we wish to classify, however, as an improved separation of the expressions has occurred due to the using the additional shape differencing step, we use this expression space as the basis for performing SVM classification.

![Figure 3. LLE low dimensional expression space when shape differencing is not used. From the legend the expression are: I=AU20+25; II=AU25+27; III=AU10+20+25; IV=AU12](image)
The aim of this experiment was to establish how our LLE-SVM based technique performs at classifying expressions at varying degrees of intensity, therefore for each of our expressions we separated our testing and training data into intensity five groupings: ABCDE; BCDE; CDE; DE; E, all of which contain the neutral expression (for example the first grouping contained the neutral along with intensities A, B, C, D and E). As we had four expressions to classify we need four one-against-all nonlinear SVM classifiers, the optimal SVM for each facial expression was selected based on the results of the classifier-test cases defined by the intensity groupings, an example of these groupings for expression 1 are shown in Table 3.

<table>
<thead>
<tr>
<th>Expression 1 - test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDE</td>
</tr>
<tr>
<td>ABCDE</td>
</tr>
</tbody>
</table>

Table 3. Expression I Intensity groupings, which are used to evaluate the SVM. For each expression we have 20 SVM training - testing pairs.

Once our classifiers were trained we tested them using the various test groupings to determine at what level the classifier broke down. For example we trained our classifier using the neutral and extreme shapes and tested it on data containing the entire intensity range to ascertain if the intervening intensities were correctly classified as belonging to the expression.

Using the outputs of our various SVMs we performed ROC curve analysis, the optimal ROC curve for each of our four expressions is shown in Table 4, and the corresponding mean AUC for the expressions are displayed in Table 6, extracts from the confusion matrices are shown in Table 5, with the complete classifier results in Table 10. We can clearly see from these results that the overall average AUC achieved by our LLE-SVM based technique is 0.81. As expected we achieved lower AUCs for the two similar facial expressions I and III, while our other two expressions II and IV performed significantly better with mean AUCs of 0.92 and 0.98 respectively. This drop in AUC for expressions I and III can be explained by looking at the detailed classifier results in Table 10, where we can see a higher number of reported false positives.

<table>
<thead>
<tr>
<th>Expression 1 - test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDE</td>
</tr>
<tr>
<td>ABCDE</td>
</tr>
</tbody>
</table>

Table 4. LLE-SVM Optimal ROC curves. The four facial expressions are: top left AU20+25; top right AU25+27; bottom left AU10+20+25; bottom right AU12

<table>
<thead>
<tr>
<th>Y</th>
<th>N</th>
<th>Y</th>
<th>N</th>
<th>Y</th>
<th>N</th>
<th>Y</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>52</td>
<td>93</td>
<td>69</td>
<td>1</td>
<td>25</td>
<td>31</td>
<td>62</td>
</tr>
<tr>
<td>n</td>
<td>48</td>
<td>329</td>
<td>71</td>
<td>381</td>
<td>65</td>
<td>401</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 5. LLE: Extract from confusion matrices of the four expressions. From left to right AU20+25, AU25+27, AU10+20+25 and AU12. Correct outcomes lie on the major diagonals and the numbers off this diagonal represent the errors.

3.2. Kernel Principal Component Analysis Results

Figure 5 shows the output space established as a result of applying KPCA to the training dataset when shape differencing is not used, and Figure 6 shows the results achieved when the additional shape differencing step is performed.
While there is not a large variation in how KPCA separated out these datasets, in order to compare the two techniques, we used the latter expression space in our experimentation. As with the previous experiment, the aim was to determine how our KPCA-SVM technique performs at classifying the four lower facial expressions at varying degrees of intensity therefore we used the same intensity groupings as for the LLE experiments as shown in Table 3.

Once our data was separated out into intensity groupings we performed KPCA on the data, and used the resulting shape spaces to train one-against-all SVMs. However, unlike the previous LLE based experiment, as the kernel trick has already been performed inside the KPCA algorithm, we use linear SVMs to classify the facial expressions.

Due to the fact that our KPCA based technique failed to separate out expressions I and III in the lower dimensional space, it is not surprising that the ROC curves for these expressions are concave. In order to to correctly interpret the results of such a seemingly weak classifier we look at extracts from the confusion matrices in Table 8 and also the mean AUC in Table 9. Here we can see that the AUC of expression I and III is very low, and when we look at the detailed results in Table 11 we also see that for these two expressions there are no true positive classifications.

### Table 6. Mean AUC for LLE-SVM based technique.

<table>
<thead>
<tr>
<th>Group</th>
<th>AUC</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-v-all</td>
<td>0.7250</td>
<td>0.0666</td>
</tr>
<tr>
<td>II-v-all</td>
<td>0.9164</td>
<td>0.0712</td>
</tr>
<tr>
<td>III-v-all</td>
<td>0.6236</td>
<td>0.0516</td>
</tr>
<tr>
<td>IV-v-all</td>
<td>0.9798</td>
<td>0.0202</td>
</tr>
<tr>
<td>Average AUC</td>
<td>0.8110</td>
<td></td>
</tr>
</tbody>
</table>

The accurate classification of facial expressions is a growing problem within several domains. The solution described in this paper takes a multidisciplinary approach drawing together psychological tools, statistical models and machine learning techniques.

The two shape models were calculated using KPCA and LLE to lower the dimensionality of the problem. We trained one-against-all SVMs to classify the four expressions (AU20+25, AU25+27, AU10+20+25 and AU12) at varying degrees of intensity. We found that overall the LLE-SVM based technique consistently performed better than the KPCA-SVM based technique. We can see from the LLE ROC curves in Table 4 that even though there is a drop in the AUCs for the two similar facial expressions when compared with the AUCs of the other expressions, this is still a satisfactory result as the differentiation of two such similar expressions is a non trivial task.

### 4. Conclusion

The accurate classification of facial expressions is a growing problem within several domains. The solution described in this paper takes a multidisciplinary approach drawing together psychological tools, statistical models and machine learning techniques.

The two shape models were calculated using KPCA and LLE to lower the dimensionality of the problem. We trained one-against-all SVMs to classify the four expressions (AU20+25, AU25+27, AU10+20+25 and AU12) at varying degrees of intensity. We found that overall the LLE-SVM based technique consistently performed better than the KPCA-SVM based technique. We can see from the LLE ROC curves in Table 4 that even though there is a drop in the AUCs for the two similar facial expressions when compared with the AUCs of the other expressions, this is still a satisfactory result as the differentiation of two such similar expressions is a non trivial task.

### 5. Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

### References

Table 7. KPCA Optimal ROC curves showing results of the application of KPCA and linear SVMs. The four facial expressions are: top left AU20+25; top right AU25+27; bottom left AU10+20+25; bottom right AU12

Table 8. KPCA extract from confusion matrices of the four expressions. From left to right AU20+25, AU25+27, AU10+20+25 and AU12. Correct outcomes lie on the major diagonals and the numbers off this diagonal represent the errors.

Table 9. KPCA mean AUC In the table above I=AU20+25, II=AU25+27, III=AU10+20+25 and IV=AU12.
Table 10. LLE-SVM detailed classifier results and ROC analysis. Expression I=AU20+25; II=AU25+27; III=AU10+20+25 and IV=AU12.

Table 11. KPCA-SVM detailed classifier results and ROC analysis. Expression I=AU20+25; II=AU25+27; III=AU10+20+25 and IV=AU12.


