Exploratory Sparse Models for Face Classification

N. P. Costen and M. Brown
The Manchester Metropolitan University
Department of Computing and Mathematics,
John Dalton Building, Chester Street,
Manchester M1 5GD, U.K.

Abstract

In this paper, a class of sparse regularization methods are considered for developing and exploring sparse classifiers for face recognition. The sparse classification method aims to both select the most important features and maximize the classification margin, in a manner similar to support vector machines. An efficient process for directly calculating the complete set of optimal, sparse classifiers is developed. This set can be explored in order to understand the sensitivity of feature selection process to small parametric changes. We show that this method can be used to construct a useful classification hyper-plane for faces represented via an appearance model. In addition, the stability of the classification process is explored and the incorporation of prior knowledge about the importance of individual features is considered.

1 Introduction

One of the fundamental problems with any image processing or computer vision task is the specification of a suitable vocabulary for representing the data. In the case of face recognition, this takes the form of the extraction of a set of parameters or dimensions which will describe the differences between people, rather than differences between images of people. This is true even after the images have been coded with the aim of the exclusion of non-facial differences as within-person variation still needs to be excluded.

One method of classification, which has shown good generalization ability when applied to large vocabularies is Support Vector Machines (SVM) [13]. Once a suitable coding of the data has been achieved, SVMs extract the optimal hyperplane for separating a pair of classes by minimizing the structural risk of miss-classification. This is a regularization, or penalty–based, approach. SVMs have been previously used for face classification [6, 11], demonstrating consistently better recognition performance than standard nearest neighbour methods. It was however necessary to vary the number of dimensions on which the decisions were made, apparently using high-variance eigenvectors. In this paper, a method for performing feature selection using SVMs is described where the optimization occurs in parameter space. By using a 1-norm model complexity function, a sparse set of features is produced because many of the classifier's parameters will be

BMVC 2003 doi:10.5244/C.17.2

estimated to be zero. This has been termed Lasso regression [12], or robust [1] or basis pursuit classification [3]. Other approaches for SVM feature selection are described elsewhere [14, 15].

When performing feature selection, it is important to understand the stability of the feature selection process. This can be achieved by exploring neighbouring sparse classifiers which are slightly more or less complex. In this paper, a new procedure for finding a complete set of sparse SVMs is described, which generates all the globally optimal classifiers as a sequence of linear programming sub-problems. This technique is then applied to the face recognition problem. In addition, this paper also considers whether the eigenvalues associated with the appearance model eigenvectors can act as suitable parametric priors for providing evidence about the relative importance of the different features and thus improving robust classification.

In this paper, a linear SVM is considered which performs 1-of-*m* classification [10]. An alternative approach to multi-class face recognition [6] constructs 2^m decision trees, allowing individual comparisons between faces to be converted into a recognition from a gallery. Similarly, a non-linear SVM has been shown to be more effective at visually determining the gender of faces [7]. The visual division between genders was shown to be notably curved, and this was captured by the support vector margin. While it is expected that other representations for multi-class recognition and more flexible kernels will provide improved recognition accuracy, the main purpose of this paper is to describe how the (weighted) feature selection procedure works and analyze its application to a high dimensional (500+), feature selection, classification problem with weighted priors.

2 Appearance models

Facial coding requires the approximation of a manifold, or high dimensional surface, on which faces lie. This allows accurate coding, recognition and reproduction of previously unseen examples. A number of previous studies [4, 5] have suggested that using a *shape-free* coding provides a ready means of doing this, at least the when the range of pose-angle is relatively small. Here, the correspondence problem between faces is first solved by finding a pre-selected set of distinctive points (corners of eyes or mouths, for example) which are present in all faces. This is typically performed by hand during training. Pixels thus defined as part of the face are warped to a standard shape by standard grey-level interpolation techniques, ensuring that the image-wise and face-wise coordinates of images are equivalent. If a rigid transformation to remove scale, location and orientation effects is performed on the point-locations, they can then be treated in the same way as the grey-levels, as again identical values for corresponding points on different faces will have the same meaning.

Although these operations will linearize the space, allowing interpolation between pairs of faces, they do not give an estimate of the dimensions. In addition, redundancies between feature-point location and grey-level values cannot be described. Both these problems can be addressed by Principal Components Analysis. Given a set of N vectors q_i (either the pixel grey-levels, or the feature-point locations) sampled from the images, the covariance matrix C of the images is calculated,

$$\mathrm{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{q}_i - \overline{\mathrm{q}})(\mathrm{q}_i - \overline{\mathrm{q}})^T, \tag{1}$$

and orthogonal unit eigenvectors Φ and a vector of eigenvalues λ are extracted from C.

Redundancies between configuration and pigmentation are removed by performing separate PCAs upon the shape and grey-levels, providing shape parameters $\mathrm{w}_{\mathrm{s}i}$ and texture parameters $\mathrm{w}_{\mathrm{t}i}$. These are combined to form a single vector for each image on which second PCA is performed [5]. This gives a single feature vectors

$$\mathrm{x} = \Phi_{\mathrm{c}}{}^{T} \left[\begin{array}{c} \mathrm{W_s}\Phi_{\mathrm{s}}{}^{T}(\mathrm{q_s} - \bar{\mathrm{q}}_{\mathrm{s}}) \\ \Phi_{\mathrm{t}}{}^{T}(\mathrm{q_t} - \bar{\mathrm{q}}_{\mathrm{t}}) \end{array} \right] \tag{2}$$

for each image, assuming zero-mean weights and where $\mathrm{W_s}$ is a diagonal vector of weights compensating for the characteristic scales of the shape and texture parameters.

This 'appearance model' allows the description of the face in terms of true, expected variation – the distortions needed to move from one to another [8]. However, it will code the entire variation between the faces which the ensemble, both between and within individuals. The following study aims to select those dimensions most appropriate for describing differences between individuals, so allowing effective recognition. The variation present in the first two dimensions of such a model is provided in Figure 1. Note that these include a mixture of pose, expression and identity variation.

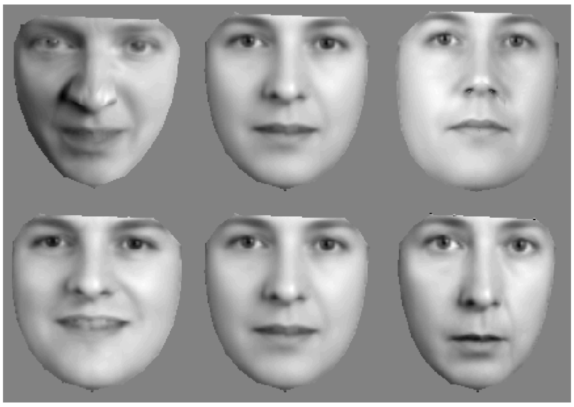

Figure 1: The first two dimensions of the face-space in the appearance model. From the left, $-2s.d.$, the mean $+2s.d.$.

3 Exploratory Basis Pursuit Classification

Basis pursuit classification is a regularization–based approach that can be used to develop sparse, kernel classifiers. The aim is to jointly minimize both a loss function that measures how well the classifier models the data and a penalty function that measures the complexity of the classifier. Specifically, for basis pursuit classification, the regularization function

$$J = \sum_{i=1}^{l} (t_i - y_i)^2 + \theta ||\mathrm{w}||_1 \tag{3}$$

is considered, where θ is the non-negative regularization parameter which controls the relative importance of the loss and penalty functions, t is the target vector of class labels, y is the corresponding class prediction vector and w is the classifier's parameter vector. Basis pursuit classification was introduced as a generalized Robust Linear Programming method [1] and has a corresponding approach for regression [3]. It differs from other regularization approaches by the method used to specify the model's complexity/penalty function, as ridge regression uses a 2-norm.

Exploratory Basis Pursuit Classification (EBPC) generates the complete set of optimal models as a function of the regularization parameter. Instead of specifying a single regularization parameter, θ, and then calculating the classifier's optimal parameters, the aim is to examine how the classifier's parameters and performance are affected by locally altering θ and thus obtain important information about the classifier's sensitivity. A fuller description can be found elsewhere [2].

3.1 Model Sparseness

For simplicity, it is assumed that classifiers are linear in their parameters, so the predictions can be calculated as

$$y = \mathrm{w} \cdot \mathrm{x} + b \tag{4}$$

where x are the input features provided to the classifier and b is the corresponding bias term. In addition, in this study, it is assumed that the basis functions are linear, so parameter selection is equivalent to feature selection (although this is obviously not necessarily the case). The classifier's output is also thresholded outside the margin [-1, 1], as occurs in SVMs and those data points that are correctly classified and lie outside the margin do not directly affect the calculation of the model's parameters and are termed in-active. The objective function is then convex with a global minima.

This regularization function produces a sparse solution with many zero parameters as can be derived from Equation 3. By differentiating J with respect to w and equating to zero

$$\mathrm{X}^T(\mathrm{t} - \mathrm{X}\mathrm{w}) = \theta \mathbf{sgn}(\mathrm{w}) \tag{5}$$

where X is the matrix of features for the training data. This is only valid for parameters that are already non-zero. If a parameter is in-active, it will remain so until a constraint

$$|\mathrm{x}_i^T(\mathrm{t} - \mathrm{y})| = \theta \tag{6}$$

is activated. When this occurs, the non-zero parameter becomes part of the active set and its values can be directly calculated

$$\mathrm{w} = \mathrm{H}^{-1}(\mathrm{X}^T\mathrm{t} - \theta \mathbf{sgn}(\mathrm{w})) \tag{7}$$

where $\mathrm{H} = \mathrm{X}^T\mathrm{X}$ is the local Hessian for the active data and parameter subspaces. It should be noted that in this application, H with not in general be proportional to the identity matrix. Since it may be computationally expensive to invert H each time, more efficient rank 1 inverse Hessian update procedures could be used [9]. As parameters change their values and the active sets of parameters change, so will the set of active data points. These properties and conditions are global because the basis pursuit regularization function described in Equation 3 has a global minimum for each value of θ. In addition, within a local region, w is a linear function of θ. The bias term b can be calculated separately.

3.2 Constructing the Optimal Classifier Space

While these relationships can be used to explain the properties of an EBPC optimal classifier for a particular value of the regularization parameter, it is useful to consider how a sparse classifier changes as the regularization parameter is altered. In this section, an algorithm is proposed that generates the complete set of optimal sparse classifiers as a function of the regularization parameter. In addition, this is achieved in an efficient manner as the algorithm is formulated as a sequence of linear programming sub-problems which have a simple form.

To begin, obviously the model with $\mathrm{w} = 0$ is optimal as it represents the simplest possible model, and is the solution of the regularization function when θ is large. As θ is reduced, at least one parameter will initially become active, see Equation 6, and its value can be calculated according to Equation 7. This direct calculation of the parameter values is valid as long as the set of active parameters and data points remains unchanged. This can be posed as a linear programming sub-problem where the aim is to maximally reduce θ, subject to a set of linear inequality constraints that control the active parameter and data sets [2]. When the linear inequality constraints are violated, this changes the members of the active sets and a new linear programming sub-problem is constructed. This terminates in a finite number of iterations as θ is reduced to zero, and generates every optimal sparse classifier, where the parameters are a piecewise linear function with respect to θ. This is an extension to the classification case of the LAR/LASSO algorithm [12].

This construction process starts with the simplest possible classifier and iteratively includes extra parameters in the active feature set and drops points from the active data set. However, the process is not strictly monotonic, and at some stages, parameters may be removed from the active set and data points re-introduced to the active data set. While this is data dependent, it should be noted that usually the active Hessian is reasonably well-conditioned as the process is effectively a forwards selection process. Also, when the problem is reasonably well–posed, the number of iterations is a near linear function of the number of features plus number of data points.

3.3 Incorporating Prior Knowledge

When data sets are sparse, feature selection is an ill-posed problem. Prior knowledge about the relative importance of the different features can be incorporated into the EBPC process. From Equation 6 if the feature is larger in magnitude, then it will become non-zero sooner in the process and indeed, it is generally recommended that all potential features are scaled to zero mean, unit variance before being used for EBPC.

Once this has been performed, it is possible to re-scale the features in order to reflect the relative prior importance of each feature. If this information is contained in the vector p, then the regularization function becomes

$$J = \sum_{i=1}^{l} (t_i - y_i)^2 + \theta \sum_{i=1}^{n} |p_i w_i| \tag{8}$$

and features or parameters become active when

$$|p_i \mathrm{x}_i^T (\mathrm{t} - \mathrm{y})| = \theta \tag{9}$$

The larger the prior is, the sooner the feature will be included in the active set. In this paper, power transformations of the eigenvalues are proposed as suitable soft priors and it is demonstrated in sections 4.4 and 4.5 that this can both improve the classifier's generalization performance and stabilize the features selected.

4 Results

4.1 Materials

The appearance model was built from an ensemble of 592 facial images, comprising 211 different individuals; this sub-divided into groups varying on facial pose, expression and lighting. Males and females were present in approximately equal proportions, and the individuals were drawn from a range of ages and ethnic groups. All the images had a uniform set of 68 landmarks found manually. A triangulation was applied to the points and bilinear interpolation used to warp the images to a standard shape and size which would yield a fixed number of pixels, here set to 5000. Lighting levels were normalized to equate the mean and variance at each pixel.

Tests were carried out using a disjoint set of 153 images of 22 individuals ($21 \times 7 + 1 \times 6$). These showed a significant level of variation, having previously been used in other experiments. Feature locations were found manually, since the issue at hand was the final representation; the images were then coded on the appearance model. Tests performed on a 10-fold cross-validation basis, with the EBPC process fully pursued for each target. Since this procedure inevitably grouped some of the images together in test or training sets, the whole procedure was repeated with different initial random orderings.

4.2 Exploring Sparseness

Figure 2 shows the evolution as a function of θ of a single classifier separating an individual from the others. As expected, there is a steady increase in the number of features contributing to the classifier (developing non-zero weights). It should also be noted that a number of classifier parameters change sign or show points of inflection when $\theta \leq 5$. This demonstrates the presence of Simpson's paradox in this data-set, and implies that using values much below it may over-fit the data and so qualitative interpretation of the parameter weights may be miss-leading. The presence of Simpson's paradox is somewhat surprising, as the eigenvectors are orthogonal with one another, but may reflect non-linearity in the clouds of identity-specific points, or may be due to the essential non-linearity of faces due to their variable shape.

4.3 Criterion Selection

The effect of varying the parameter θ is shown in Figure 3 with respect to the accuracy of classification of the training and test images, taking the mean for different individuals and cross-validation sets. Clearly while the errors for the training images smoothly decline to zero at a value of $\theta = 4$, the test images show rather more noise and only plateau at a lower value. Thus the selection of a critical value of θ is unclear. Two classifier sets were chosen, those at a knot point with the highest value of θ allowing perfect training classification, and those with the lowest such value. This bracketed the possible values.

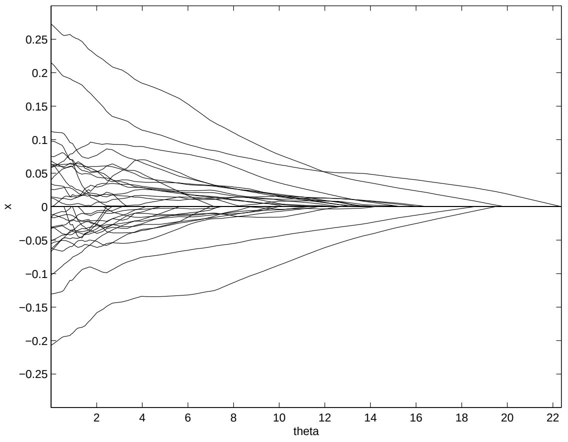

Figure 2: Effects of varying the parameter θ on the classifier's parameters.

4.4 Specifying Eigenvalue Priors

While the probability of inclusion of an eigenvector in the classifiers was determined by the eigenvalues of the combined appearance model eigenvectors, it is not obvious that the relative proportions (as opposed to the orderings) of this distribution was the most appropriate set of values. This was addressed by setting the priors to

$$p_i = \lambda_i^\alpha \tag{10}$$

and then searching with $\alpha = [0,2]$. This is similar to the Box-Cox power transformations in linear regression. Note that the power of 0 corresponds with equal weighting. The results (shown as number of errors out of 3363 tests) are shown in Figure 4.

As well as the obvious main effect of α, with lowest errors for the fourth root of the eigenvalues for both criteria, there is an interesting interaction. While the number of errors rises with increased power, it does so faster for the less sparse classifier, with the $\alpha = 2$ eigenvalues giving more errors when the classifier is made less sparse. As the power factor increases the classifier becomes less representative. This reflects the power transformation decreasing the set of eigenvectors from which parameters can be chosen. While this is advantageous for small powers (where this excludes unreliable low-variance eigenvectors), this becomes less useful as the power increases and the classifier concentrates upon only high-variance eigenvectors. When θ is dropped to its minimum value, dimensions which are significantly less useful in classifying test-images may be included; this effect is sufficiently extreme for the higher α classifiers to increase the number of errors.

4.5 Feature Selection

One way of interpreting the classification functions derived by EBPC is by considering the probability of inclusion of the features, in this case the eigenvectors. This is simply the count of non-zero w_i values across classifiers for different individuals and cross-validation repetitions. This is shown in Figure 5 for both the equal-probability condition ($\alpha = 0$),

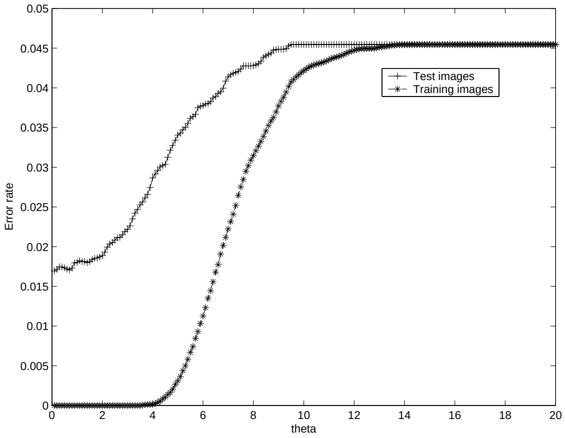

Figure 3: Effects of varying the parameter θ on classification accuracy of test and training images on an equal-weight condition.

and the fourth-root eigenvalues ($\alpha = 0.25$). In both cases the least sparse classifier was used.

Note that while the equal-weighting condition produces a remarkably even spread across the 518 features, with a slight tendency towards higher probabilities for later, lower eigenvalue ones, the weighted case shows a pronounced tendency towards early features. Some low-variance eigenvector have been included. Thus soft selection has been implemented, with a bias away from unreliable parameters, which can still however be used, where they are particularly important.

A second method of weighting the features was also investigated, using the eigenvalues of a Linear Discriminate Analysis of the images to weight the appearance model eigenvectors. Although this did not change classification performance, this did move the point of minimum errors to a power of unity.

5 Discussion and Conclusions

In this paper, a sparse SVM has been used for recognizing individuals based on orthogonal features generated from an appearance model. One property of the appearance model is the large number of candidate features (500+) that were generated from a relatively small training set. Hence, the sparse SVM was used to select important features and a new, novel algorithm for generating a complete set of sparse classifiers was proposed. By exploring all the optimal, sparse classifiers, insights could be gained into the stability of both the feature selection and parameter estimation processes. One slightly surprising property was the interaction between the estimated parameters, when new features were introduced into the active set and data points dropped. This was surprising because the features were constructed to be orthogonal, and it does illustrate how feature selection can be a complex process. In addition, the paper also illustrated how soft priors can be incorporated into the feature selection process and when the prior was a power transformation of the eigenvalues, it was shown to improve both the classifier's generalization performance and the

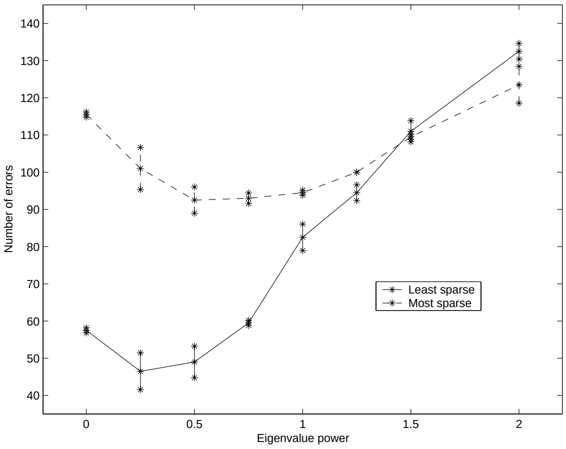

Figure 4: Changes in classification errors with power transformation of the eigenvalues for highest and lowest θ values giving no training errors.

stability of the selected features. While the latter is to be expected, the former property is more interesting, as it allows for the incorporation of expert knowledge about the relative importance of different features in a soft fashion, and this can improve the ”conditioning” of the problem. This allows the integration of the appearance model and SVMs, as the former are now supplying the latter with parameterized dimensions, rather than the usual arbitrary basis.

This paper has concentrated on the description and application of the sparse EBPC method. Future work will consider how best to represent the multi-class nature of the face recognition problem, the extension of the above sparse method to non-linear kernels, comparison with other sparse classification techniques and the use of other sets of priors.

References

[1] P. S. Bradley and O. L. Magasarian. Feature selection via concave minimization and Support Vector Machines. In *International Conference on Machine Learning*, pages 82–90, 1998.

[2] M. Brown. Exploring the set of sparse, optimal classifiers. To appear, International Workshop on Artifical Neural Networks in Pattern Recognition, 2003.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing*, 20(1):33–61, 1998.

[4] N. P. Costen, I. G. Craw, G. J. Robertson, and S. Akamatsu. Automatic face recognition: What representation ? In *European Conference on Computer Vision*, pages 504–513, 1996.

[5] G. J. Edwards, A. Lanitis, C. Taylor, and T. F. Cootes. Modelling the variability in face images. In *IEEE International Conference on Face and Gesture Recognition*, pages 328–333, 1996.

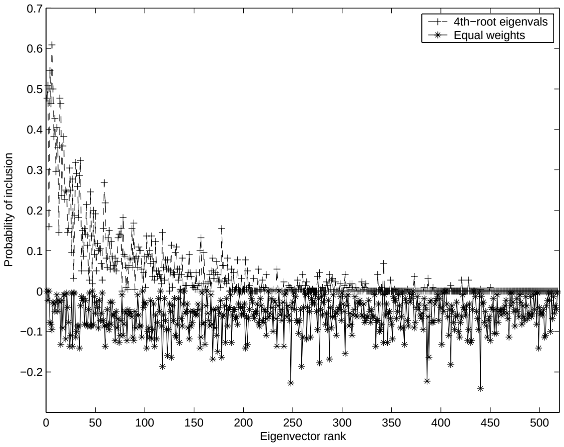

Figure 5: Probabilities of inclusion of features in any of the classifiers for the 22 people, for the equal and 4^{th}-root eigenvalue conditions. Probabilities have been negated for the equal weight condition for display.

[6] G. Guo, S. Z. Li, and K. L. Chan. Support vector machines for face recognition. *Image and Vision Computing*, 19:631–638, 2001.

[7] B. Moghadam and M. Yang. Learning gender with support faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):707–711, 2002.

[8] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: Probabilistic matching for face recogntion. In *IEEE International Conference on Face and Gesture Recognition*, pages 30–35, 1998.

[9] M. Orr. Introduction to radial basis function networks. Technical report, Division of Informatics, Edinburgh University, 1996.

[10] S. Pang, D. Kim, and S. Y. Bang. Membership authentication in the dynamic group by face classification using SVM ensemble. *Pattern Recognition Letters*, pages 215–225, 2003.

[11] P. J. Phillips. Support vector machines applied to face recognition. In *Neural Information Processing Systems*, 1999.

[12] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

[13] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[14] J. Weston, J. Mukherjee, O. Chapelle, T. Poggio, and V. Vapnick. Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13*, pages 668–674, 2002.

[15] J. Zhang and Y. Liu. SVM decision boundary based discriminative subspace induction. Technical Report CMU-RI-TR-02-15, The Robotics Institue, Carnegie Mellon University, Pittsburgh, PA 15213, 2002.