

Incremental Eigenanalysis for Classification

Peter M. Hall, David Marshall, and Ralph R. Martin

Department of Computer Science

Cardiff University

Cardiff, CF2 3XF

[peter|dave|ralph]@cs.cf.ac.uk

Abstract

Eigenspace models are a convenient way to represent sets of observations with widespread applications, including classification. In this paper we describe a new constructive method for incrementally adding observations to an eigenspace model. Our contribution is to explicitly account for a change in origin as well as a change in the number of eigenvectors needed in the basis set. No other method we have seen considers change of origin, yet both are needed if an eigenspace model is to be used for classification purposes. We empirically compare our incremental method with two alternatives from the literature and show our method is the more useful for classification because it computes the smaller eigenspace model representing the observations.

1 Introduction

The contribution of this paper is a method for incrementally computing eigenspace models in the context of using them for classification. Eigenspace models are widely used in computer vision. Applications include: face recognition [8] where observations are images which lie in a linear space formed by lexicographically ordered pixels; modelling variable geometry [5] where observations are ordered points from a curve; and estimation of motion parameters [4] where observations comprise matching points in consecutive frames. Only the first two of these applications are examples of classification. Previous workers have also considered incremental eigenspace computation, but their methods have severe limitations when used for classification, namely they do not consider a shift of origin. This is important, as will be described later.

Before continuing, a note on notation is in order. Vectors are columns, and denoted by a single underline. Matrices are denoted by a double underline. The size of a vector, or matrix, where it is important, is denoted with subscripts. Particular column vectors within a matrix are denoted with a superscript, while a superscript on a vector denotes a particular observation from a set of observations, so we treat observations as column vectors of a matrix. As an example, $\underline{\underline{A}}_{mn}^i$ is the i th column vector in an $(m \times n)$ matrix. We denote a column extension to a matrix using square brackets. Thus $[\underline{\underline{A}}_{mn} \underline{b}]$ is an $(m \times (n + 1))$ matrix, with vector \underline{b} appended to $\underline{\underline{A}}_{mn}$, as a last column.

Eigenspace models can be computed using eigenvalue decomposition (EVD) of the covariance matrix, $\underline{\underline{C}}_{nn}$ of a set of N observations $\underline{\underline{X}}_{nN}$, where $\underline{\underline{C}}_{nn} = (1/N) \sum_{i=1}^N (\underline{\underline{x}}^i - \underline{\bar{x}})(\underline{\underline{x}}^i - \underline{\bar{x}})^T$, and $\underline{\bar{x}}$ is the mean of the observations. (This

technique is also referred to as principal component analysis or Karhunen-Loeve expansion.) Such models can be regarded as a p -dimensional hyperellipse in an n -dimensional space, where n is the number of variables per observation, and p is the number of new variables used to represent those observations to within some desired degree of accuracy: thus $p \leq n$. The hyperellipse's centre is at the mean of the observations \bar{x} , its axes are the eigenvectors which are the columns of the matrix \underline{U}_{np} , and the lengths of its axes are the square-roots of the eigenvalues along the diagonal of $\underline{\Lambda}_{pp}$. The full eigensystem involved is $\underline{C}_{nn}\underline{U}_{nn} = \underline{U}_{nn}\underline{\Lambda}_{nn}$; certain eigenvalues are discarded using an appropriate criterion which reduces this system to the approximation $\underline{C}_{nn}\underline{U}_{np} \approx \underline{U}_{np}\underline{\Lambda}_{pp}$.

In practice, "low-dimensional" methods are available that solve eigenproblems no larger than the number of observations, see Murakami and Kumar [9], or Sirovich and Kirby [10] for examples. Note that the rank of the covariance matrix may be less than the number of observations.

An alternative approach to computing the eigenmodel is to use singular value decomposition (SVD).

We must also make clear the difference between *batch* and *incremental* methods for computing eigenspace models. A batch method computes an eigenmodel using all observations simultaneously. An incremental method computes an eigenspace model by successively updating an earlier model as new observations become available. In either case, the observations used to construct the eigenspace model are the *training observations*; that is, they are instances from some class. This model is then used to decide whether further observations belong to the class.

Despite the popularity of eigenspace models, there is little in the vision literature for computing them incrementally [3, 4, 9], although researchers in other fields have addressed the issue. For example, the numerical analysts Bunch *et al.* update eigenmodels using EVD [1], and again using SVD [2]. Their work was built upon by De Groat and Roberts who, working in signal processing, examined error accumulation [6] of such algorithms.

Incremental methods are important to the vision community because of the opportunities they offer. We give two examples: (a) They allow the construction of eigenmodels via procedures that use less storage, and so render feasible some previously inaccessible problems [3]; (b) They could be used as part of a learning system in which observations are added to an eigenmodel. For example, a security system may need to learn new people as part of a classifier system.

The first of these applications requires that the observations are reproducible from their eigenspace representation to a very high fidelity. We define *fidelity* as the reciprocal of the size of the *residue vector*, which in turn is defined by $\underline{h} = (\underline{x} - \bar{x}) - \underline{U}_{np}\underline{U}_{np}^T(\underline{x} - \bar{x})$, where \underline{x} is an observation. The residue vector is that part of the observation which is perpendicular to the eigenspace.

The second application requires that observations are well classified. For this to be so, the eigenspace model describing the training observations must have a hyperellipse no larger than necessary, so that observations are correctly classified whether they belong to the class or not. If the hyperellipse is too large, then too many observations will be classified incorrectly as belonging. Typically the classification is decided by computing a probability measure. It is conventional to use a multi-dimensional Gaussian distribution with eigenspace models, in which case the surface of the hyperellipse can be considered as a contour at one standard deviation. Thus the classification measure for an observation

\underline{x} is $\exp(-0.5(\underline{x} - \bar{\underline{x}})^T (\underline{C}_{nn})^{-1} (\underline{x} - \bar{\underline{x}})) / ((2\pi)^{p/2} \det(\underline{\Lambda}_{pp})^{0.5})$.

Note that in an incremental method, $\bar{\underline{x}}$ is unknown until all the training observations have been input. For reproduction of observations, our experiments showed improved fidelity if we used the true mean of the observations rather than assuming a mean at the origin. In classification methods, our experiments showed that using the wrong value of $\bar{\underline{x}}$ generally leads to too large a hyperellipse and thus poor classification. No previous incremental method in the literature we have found [1, 2, 3, 4, 6, 9] attempt to estimate the mean; and they use the origin in its place. Thus their methods while being useful for reproduction applications, are not appropriate for classifiers. In the rest of this paper we give an incremental method which does update the mean as observations are added to produce a model which is useful for classification. Of course, our models can also be used for reproduction.

Section 2 states the problem more precisely and then derives our new incremental method. In Section 3, we go on to compare it with two alternative incremental methods from the literature [3, 9]. Our experimental results (Section 4) show that our method is consistently the best for classifying, yet loses little (and often gains) in fidelity. Final conclusions are drawn in Section 5.

2 The problem and our method

2.1 The problem

An eigenspace model, Ω , constructed over N observations $\underline{x}^i \in \mathfrak{R}^n$ comprises a mean, a set of eigenvectors, the eigenvalue associated with each eigenvector, and the number of observations. The mean is given by: $\bar{\underline{x}} = \frac{1}{N} \sum_{i=1}^N \underline{x}^i$. The eigenvectors are columns of the matrix \underline{U}_{nn} , and the spread of the observations over each is measured by the corresponding eigenvalue in the diagonal matrix $\underline{\Lambda}_{nn}$. The eigenvectors and their eigenvalues are solutions to the eigenproblem: $\underline{C}_{nn} \underline{U}_{nn} = \underline{U}_{nn} \underline{\Lambda}_{nn}$ in which \underline{C}_{nn} is the covariance matrix defined previously. Typically, only p of the eigenvectors and eigenvalues are kept, and $p \leq \text{rank}(\underline{C}_{nn}) \leq \min(n, N)$. The criteria for keeping eigenvectors and eigenvalues vary, and are application dependent. We give three examples: (a) keep the p largest eigenvectors [9]; (b) keep all eigenvectors whose eigenvalue exceed an absolute threshold [3]; (c) keep the largest eigenvectors such that a specified fraction of energy in the eigenvalue spectrum is retained. In any case, the remaining $n-p$ eigenvalues and their corresponding eigenvectors are discarded. From now on, we use those eigenvectors and eigenvalues that remain as our eigenspace model (or eigenmodel), and denote it by $\Omega = (\bar{\underline{x}}, \underline{U}_{np}, \underline{\Lambda}_{pp}, N)$.

It is instructive to examine discarding eigenvectors and eigenvalues using spectral decomposition and writing matrices in block form, thus:

$$\begin{aligned}
\underline{C}_{nn} &= \underline{U}_{nn} \underline{\Lambda}_{nn} \underline{U}_{nn}^T \\
&= [\underline{U}_{np} \underline{U}_{n(n-p)}] \begin{bmatrix} \underline{\Lambda}_{pp} & \underline{0}_{p(n-p)} \\ \underline{0}_{(n-p)p} & \underline{\Lambda}_{(n-p)(n-p)} \end{bmatrix} [\underline{U}_{np} \underline{U}_{n(n-p)}]^T \\
&= \underline{U}_{np} \underline{\Lambda}_{pp} \underline{U}_{np}^T + \underline{U}_{n(n-p)} \underline{\Lambda}_{(n-p)(n-p)} \underline{U}_{n(n-p)}^T \\
&\approx \underline{U}_{np} \underline{\Lambda}_{pp} \underline{U}_{np}^T
\end{aligned} \tag{1}$$

provided that $\underline{\underline{\Lambda}}_{(n-p)(n-p)} \approx \underline{\underline{0}}_{(n-p)(n-p)}$.

A new observation \underline{y} is projected into the eigenspace model to give a p -dimensional vector, \underline{g} , using the eigenvectors as a basis. This \underline{g} can be projected back into n -dimensional space, but with loss represented by the residue vector \underline{h} :

$$\underline{g} = \underline{\underline{U}}_{np}^T (\underline{y} - \underline{\bar{x}}) \quad (2)$$

$$\underline{h} = (\underline{y} - \underline{\bar{x}}) - \underline{\underline{U}}_{np} \underline{g} \quad (3)$$

The vector $\underline{\bar{x}} + \underline{\underline{U}}_{np} \underline{g}$ lies entirely within the subspace spanned by the eigenvectors. The residue vector is orthogonal to this, and it lies in the complementary space, i.e. \underline{h} is orthogonal to every vector in the eigenmodel. If \underline{h} is large, then the observation \underline{y} is not well represented by the eigenmodel. Therefore, if a general incremental method is to be effective it must be able to include new, orthogonal directions as new observations are included in the model; in short, it must allow the number of dimensions to increase when appropriate.

The problem we address is this: given an eigenspace model $\Omega = (\underline{\bar{x}}, \underline{\underline{U}}_{np}, \underline{\underline{\Lambda}}_{pp}, N)$, but not neither the original observations nor their correlation matrix, and a new observation $\underline{y} \in \mathbb{R}^n$, estimate the eigenspace model $\Omega' = (\underline{\bar{x}}', \underline{\underline{U}}_{nq}, \underline{\underline{\Lambda}}'_{qq}, N+1)$ that would be computed from all previous observations and the new one. Importantly, the incremental method should include an additional eigenvector if necessary, hence $q = p+1$ or $q = p$. Equally importantly it should account for a change of mean, to be effective for classification.

2.2 Derivation of our method

We now show how a new eigenspace model can be incrementally computed using EVD. The eigenproblem, after adding \underline{y} is

$$\underline{\underline{C}}'_{nn} \underline{\underline{U}}'_{nn} = \underline{\underline{U}}'_{nn} \underline{\underline{\Lambda}}'_{nn} \quad (4)$$

It is easy to confirm that the new mean is:

$$\underline{\bar{x}}' = \frac{1}{N+1} (N\underline{\bar{x}} + \underline{y}) \quad (5)$$

and the new covariance matrix is:

$$\underline{\underline{C}}'_{nn} = \frac{N}{N+1} \underline{\underline{C}}_{nn} + \frac{N}{(N+1)^2} \underline{y}' (\underline{y}')^T \quad (6)$$

where $\underline{y}' = \underline{y} - \underline{\bar{x}}$.

We assume initially that $q = p+1$; if it turns out that the additional eigenvalue is small we will discard it and the corresponding eigenvector at a later stage.

The new eigenvectors must be a rotation, $\underline{\underline{R}}_{(p+1)(p+1)}$, of the current eigenvectors plus some new orthogonal unit vector. The unit residue vector is an obvious choice for the additional vector. We form the unit residue vector; however, if the new observation lies exactly within the current eigenspace, then the residue is zero:

$$\hat{\underline{h}} = \begin{cases} \frac{\underline{h}}{\|\underline{h}\|_2} & \text{if } \|\underline{h}\|_2 \neq 0 \\ \underline{0} & \text{otherwise} \end{cases} \quad (7)$$

and setting

$$\underline{\underline{U}}'_{nq} = [\underline{\underline{U}}_{np}, \hat{\underline{h}}] \underline{\underline{R}}_{(p+1)(p+1)} \quad (8)$$

Substitution of Equations 6 and 8 into 4, followed by left multiplication by $([\underline{\underline{U}}_{np}, \hat{\underline{h}}]^T$ gives:

$$\begin{aligned} [\underline{\underline{U}}_{np}, \hat{\underline{h}}]^T \left(\frac{N}{N+1} \underline{\underline{C}}_{nn} + \frac{N}{(N+1)^2} \underline{y}'(\underline{y}')^T \right) [\underline{\underline{U}}_{np}, \hat{\underline{h}}] \underline{\underline{R}}_{(p+1)(p+1)} = \\ \underline{\underline{R}}_{(p+1)(p+1)} \underline{\underline{\Lambda}}'_{(p+1)(p+1)} \end{aligned} \quad (9)$$

which is a $(p+1) \times (p+1)$ eigenproblem with solution eigenvectors $\underline{\underline{R}}_{(p+1)(p+1)}$ and solution eigenvalues $\underline{\underline{\Lambda}}'_{(p+1)(p+1)}$. Note that we have reduced dimensions from n to $p+1$ by discarding eigenvalues in $\underline{\underline{\Lambda}}'_{nn}$ already deemed negligible.

The left hand side comprises two additive terms. Up to a scale factor the first of these terms is

$$\begin{aligned} [\underline{\underline{U}}_{np}, \hat{\underline{h}}]^T \underline{\underline{C}}_{nn} [\underline{\underline{U}}_{np}, \hat{\underline{h}}] &= \begin{bmatrix} \underline{\underline{U}}_{np}^T \underline{\underline{C}}_{nn} \underline{\underline{U}}_{np} & \underline{\underline{U}}_{np}^T \underline{\underline{C}}_{nn} \hat{\underline{h}} \\ \hat{\underline{h}}^T \underline{\underline{C}}_{nn} \underline{\underline{U}}_{np} & \hat{\underline{h}}^T \underline{\underline{C}}_{nn} \hat{\underline{h}} \end{bmatrix} \\ &\approx \begin{bmatrix} \underline{\underline{\Lambda}}_{pp} & \underline{\underline{0}} \\ \underline{\underline{0}}^T & 0 \end{bmatrix} \end{aligned} \quad (10)$$

where $\underline{\underline{0}}$ is a p -dimensional vector of zeros. This uses Equation 1, $\underline{\underline{C}}_{nn} \approx \underline{\underline{U}}_{np} \underline{\underline{\Lambda}}_{pp} \underline{\underline{U}}_{np}^T$ and the fact that $\hat{\underline{h}}$ is orthogonal to every vector in $\underline{\underline{U}}_{np}$.

The second term, up to a scale factor, is

$$\begin{aligned} [\underline{\underline{U}}_{np}, \hat{\underline{h}}]^T \underline{y}'(\underline{y}')^T [\underline{\underline{U}}_{np}, \hat{\underline{h}}] &= \begin{bmatrix} \underline{\underline{U}}_{np}^T \underline{y}'(\underline{y}')^T \underline{\underline{U}}_{np} & \underline{\underline{U}}_{np}^T \underline{y}'(\underline{y}')^T \hat{\underline{h}} \\ \hat{\underline{h}}^T \underline{y}'(\underline{y}')^T \underline{\underline{U}}_{np} & \hat{\underline{h}}^T \underline{y}'(\underline{y}')^T \hat{\underline{h}} \end{bmatrix} \\ &= \begin{bmatrix} \underline{g}\underline{g}^T & \gamma \underline{g} \\ \gamma \underline{g}^T & \gamma^2 \end{bmatrix} \end{aligned} \quad (11)$$

where we have used Equations 2 and 3, and set $\gamma = \hat{\underline{h}}^T \underline{y}'$.

Thus, using these results, to compute the updated eigenspace model we must solve an intermediate eigenproblem of size $(p+1) \times (p+1)$:

$$\begin{aligned} \left(\frac{N}{N+1} \begin{bmatrix} \underline{\underline{\Lambda}} & \underline{\underline{0}} \\ \underline{\underline{0}}^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2} \begin{bmatrix} \underline{g}\underline{g}^T & \gamma \underline{g} \\ \gamma \underline{g}^T & \gamma^2 \end{bmatrix} \right) \underline{\underline{R}}_{(p+1)(p+1)} = \\ \underline{\underline{R}}_{(p+1)(p+1)} \underline{\underline{\Lambda}}'_{(p+1)(p+1)} \end{aligned} \quad (12)$$

A solution to this problem yields the new eigenvalues directly, and the new eigenvectors are then computed from Equation 8. For future reference we call the the matrix whose eigensolution is sought above $\underline{\underline{D}}_{(p+1)(p+1)}$.

We note some properties of our solution. In the case when $\hat{\underline{h}} = \underline{\underline{0}}$, $\underline{\underline{D}}_{pp}$ reduces to $\frac{N}{N+1} \underline{\underline{\Lambda}} + \frac{N}{(N+1)^2} \underline{g}\underline{g}^T$; the effect is simply to rotate the current eigenvectors. Should it happen that $\underline{y} = \underline{\underline{x}}$, then the new eigenvectors are scaled by $\frac{N}{N+1}$. Finally, as $N \mapsto \infty$, so $\frac{N}{N+1} \mapsto 1$ and $\frac{N}{(N+1)^2} \mapsto 0$, which shows our method converges to a stable solution in the limit.

3 Previous work

We fix our attention on solutions in the vision literature that allow for a change in dimension (some do not); we have found two such methods. No method we have found, in either the vision literature or elsewhere, allows for a change in origin. All methods, however presented, use the same basic method as our: they solve an intermediate eigenproblem of the form $\underline{\underline{D}}_{(p+1)(p+1)} \underline{\underline{R}}_{(p+1)(p+1)} = \underline{\underline{R}}_{(p+1)(p+1)} \underline{\underline{\Lambda}}'_{(p+1)(p+1)}$ to compute rotation matrices and eigenvalues, but their $\underline{\underline{D}}_{(p+1)(p+1)}$ takes a different form. Thus, we can compare the methods considered on the basis of the form of the “intermediate matrix”, $\underline{\underline{D}}$ and the kind of decomposition (EVD or SVD) used.

Murakami and Kumar [9] assume a fixed mean at the origin. They compute the EVD of

$$\underline{\underline{D}}_{(p+1)(p+1)} = \frac{N}{N+1} \begin{bmatrix} \underline{\underline{\Lambda}}_{pp} & \underline{\underline{0}}_p \\ \underline{\underline{0}}_p^T & 0 \end{bmatrix} + \frac{N^{1/2}}{N+1} \begin{bmatrix} \underline{\underline{0}}_{pp} & \underline{\underline{\Lambda}}_{pp}^{1/2} \underline{\underline{g}}_p \\ \underline{\underline{g}}_p^T \underline{\underline{\Lambda}}_{pp}^{1/2} & \underline{\underline{y}}'^T \underline{\underline{y}}' \end{bmatrix} \quad (13)$$

where $\underline{\underline{0}}_p$ is a p vector of zeros, and $\underline{\underline{0}}_{pp}$ is a $p \times p$ matrix of zeros, and $\underline{\underline{\Lambda}}_{pp}^{1/2}$ is the diagonal matrix whose entries are the square roots of $\underline{\underline{\Lambda}}_{pp}$. (In the above $\underline{\underline{x}}$ is set to zero, so that $\underline{\underline{y}}'$ and $\underline{\underline{g}}$ have different values than in our method.) The eigenvectors and eigenvalues of the new model are then given as:

$$\underline{\underline{U}}_{n(p+1)} = [\underline{\underline{U}}_{np}, \hat{\underline{\underline{h}}}] \underline{\underline{R}}_{(p+1)(p+1)} ((N+1) \underline{\underline{\Lambda}}'_{(p+1)(p+1)})^{-1/2} \quad (14)$$

The scaling term $((N+1) \underline{\underline{\Lambda}}'_{(p+1)(p+1)})^{-1/2}$ ensures that the new eigenvectors form an orthonormal basis set. We note the risk of division by zero, which in practice means that very small eigenvectors must be removed if the system is not to suffer from numerical problems. In addition, we see they lack a $\underline{\underline{g}}\underline{\underline{g}}^T$ term in the second matrix, which provides rotation and scaling even when the new observation lies in the same eigenspace as the previous observations. They proposed that a stipulated number of eigenvectors should be kept. Later work by Vermeulen and Casasent [11] enhanced this method by providing a way to determine whether a new observation lies in the same eigenspace as the previous observations.

Chandrasekaran *et al.* [3] also use a fixed mean at the origin. Despite the fact they use SVD rather than EVD they still solve an “intermediate problem”. They compute the SVD of:

$$\underline{\underline{D}}_{(p+1)(p+1)} = \begin{bmatrix} \underline{\underline{\Lambda}}_{pp}^{1/2} & \underline{\underline{0}}_p \\ \underline{\underline{0}}_p^T & 0 \end{bmatrix} + \begin{bmatrix} \underline{\underline{0}}_{pp} & \underline{\underline{g}}_p \\ \underline{\underline{0}}_p^T & \gamma \end{bmatrix} \quad (15)$$

Again, note $\underline{\underline{x}}$ has been set to zero. The SVD of this matrix yields left singular vectors $\underline{\underline{R}}_{(p+1)(p+1)}$, singular values $(\underline{\underline{\Lambda}}'_{(p+1)(p+1)})^{1/2}$, and right singular vectors $\underline{\underline{S}}_{(p+1)(p+1)}$. The new left singular vectors are computed exactly as in Equation 8, the new singular values are identically $(\underline{\underline{\Lambda}}'_{(p+1)(p+1)})^{1/2}$, and the right singular vectors are not used. The form of $\underline{\underline{D}}_{(p+1)(p+1)}$ arises because Chandrasekaran *et al.* [3] compute the SVD of $\underline{\underline{X}}_{nN}$, which is the matrix in which every column is an observation, whereas EVD is computed from the covariance matrix of the observations.

Other authors have also used SVD. For example, Chaudhuri *et al.* [4] also use SVD in recursive estimation of motion parameters, but set $\underline{D}_{pp} = \underline{\Lambda}_{pp} + \underline{g}\underline{g}^T$ and consequently make no allowance for a change in the rank of the covariance matrix. Therefore, their work is of interest to this paper only in that they also include a $\underline{g}\underline{g}^T$ term.

4 Experimental Method and Results

Space prevents us from giving here all results from all the experimental tests we have performed. Rather, we summarise some results and present results relating to fidelity and classification in a little more detail. The reader is referred elsewhere for a more detailed report [7].

No matter which of the incremental methods was under consideration, we used a strategy whereby as each new observation was added, the size $p + 1$ new eigenmodel was computed, and then a decision was made as to whether to keep it, or reduce it to size p . We examined the effects of three different methods for reduction: the first kept all eigenvectors whose eigenvalues exceeded an absolute threshold (recommended by [3]), the second kept a stipulated number of the largest eigenvectors (recommended by [9]); the third kept the largest eigenvectors such that a stipulated fraction of the eigenspectrum energy was retained.

We measured general accuracy of incremental eigenmodels compared to batch eigenmodels, ensuring EVD models were compared with EVD models, and SVD models with SVD models. Accuracy was measured in a variety of ways: average alignment of closest eigenvectors; the average difference between eigenvalues; and the energy in the batch model not accounted for by the incremental model (and vice-versa). The most accurate was the SVD method of Chandrasekaran *et al.* [3]. For example, if more than about 95% of the energy was retained, then the average angular deviation of incremental eigenvectors and batch eigenvectors was about 5° . Under the same conditions our method gave eigenvectors whose average angular deviation was about 15° , while the measure for Murakami and Kumar [9] was about 45° . Again, the SVD method was also more accurate in terms of eigenvalues. However, our method proved best in terms of energy measures.

However desirable similarity to the batch model is, the task in hand is to represent incrementally presented observations. High fidelity is important to general applications, while probability measures are of importance for classification (as discussed in Section 1). In both cases, we discarded eigenvectors by stipulating a number to be kept. However, when using our method we kept one vector less than when using either of the comparative methods. This is because we explicitly keep a mean, and doing the former allows the testing to be fair in that each method keeps the same amount of information.

Fidelity measures how well the observations used to construct an eigenmodel are represented by that model. To measure this, for each observation we computed the size of the residue vector with respect to the final model. Here we use the mean value of the residue as a measure of error, fidelity is the reciprocal of this.

Classification is based on the probability that observation \underline{x} belongs to eigenspace model Ω :

$$P(\underline{x}|\Omega) \approx \frac{\exp(-0.5(\underline{x} - \bar{\underline{x}})^T \underline{U}_{np} \underline{\Lambda}_{pp}^{-1} \underline{U}_{np}^T (\underline{x} - \bar{\underline{x}}))}{(2\pi)^{p/2} \det(\underline{\Lambda})^{1/2}} \quad (16)$$

in which $(\mathbf{x} - \bar{\mathbf{x}})^T \underline{\underline{U}}_{np} \underline{\underline{\Lambda}}_{pp}^{-1} \underline{\underline{U}}_{np}^T (\mathbf{x} - \bar{\mathbf{x}})$ is the Mahalanobis distance [8]. We computed the Mahalanobis distance for each observation used to construct the eigenspace model, and then computed the probability of the mean distance as our classification measure.

One prediction we can make regarding classification is that for fixed-mean methods the observations should be less well classified the further the cluster mean is from the origin because the hyperellipse representing the data grows in volume. However, for varying-mean methods the classification value should be invariant because the hyperellipse is centred on the data. To test this we generated N observations in an n dimensional space, for various values of n and N ; $n < N$, $n = N$, and $n > N$ were all tried. The cluster was initially generated such that the mean was at the origin, and there was unit standard deviation in every direction. We then progressively shifted this cluster away from the origin, until the cluster was 10 standard deviations from the origin.

Typical results are presented in Figure 1(a), which shows 10 observations in a 100 dimensional space; in this figure 10 eigenvectors were kept for the comparator methods, our method kept 9 eigenvectors plus the mean. These results bear out our prediction regarding classification. Figure 1(b) shows the mean residue as a function of distance. This rises with distance for other methods, but hardly at all with ours. Thus, when all eigenvectors are kept our method gives better performance.

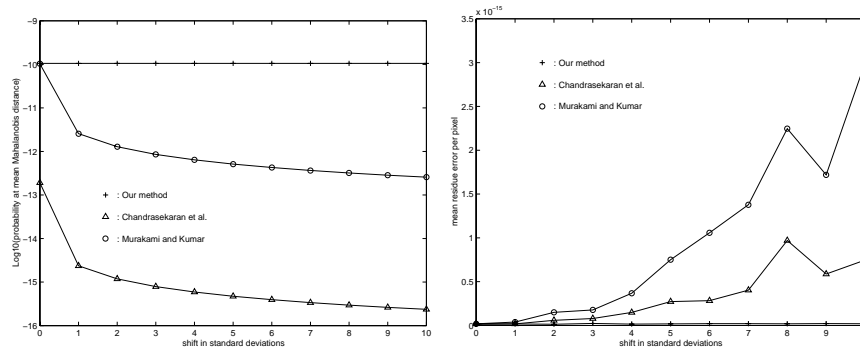


Figure 1: Probability and residue measures as a function of distance: 1(a) \log_{10} of probability at mean Mahalanobis distance v. shift; 1(b) mean residue error v. shift.

A second prediction is that the fixed-mean methods are prone to mis-classification. To show this we again used synthetic data. We computed an eigenmodel using a set of training observations, and then computed the fidelity and classification measures on a set of test observations. The training set was synthesised in the same way, shifted as before. The test set was also generated as a cluster, but always centred on the origin. Clearly, as the distance of the training set from the origin increases, the classification measure should fall as the two sets become ever more distinct.

Typical results are presented in Figure 2(a). Again, the results bear out our prediction, we see that the probability of mis-classification with our method falls with distance, while those for comparator methods remain about constant. Figure 2(b) shows that as the test set becomes increasingly distant from the training set our residue error grows, while it does not for the other methods.

Finally, we present results of experiments using real image data in which we tested the

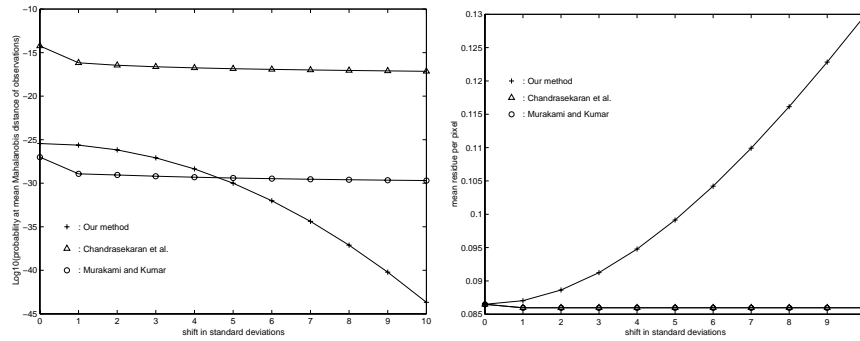


Figure 2: Probability of mis-classification and residue error of observations in set larger than class: 2(a) \log_{10} of probability at mean Mahalanobis distance v. shift; 2(b) mean residue error v. shift .

effect of discarding eigenvectors by keeping a fixed number, as for the synthetic data. We randomly selected 50 images from the Olivetti database of faces¹. We then computed an eigenmodel using each of the three incremental methods, and computed the residue and classification measures. Of interest here was the performance of each method as varying numbers of eigenvectors and eigenvalues were discarded. Results are shown in Figure 3. We see that our method consistently classifies the observations better than the fixed mean methods, and that it is consistently competitive with respect to mean residue error.

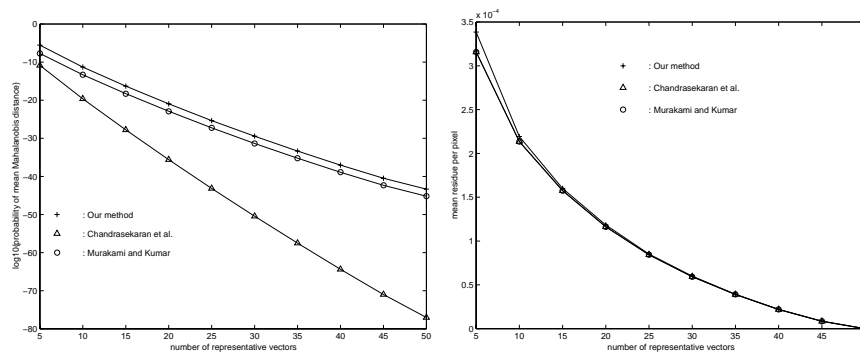


Figure 3: The variation in classification and residue with number of vectors retained, real image data used: 3(a) \log_{10} of probability at mean Mahalanobis distance v. vectors kept; 3(b) mean residue error v. vectors kept.

5 Conclusions

We have presented a new method for incremental eigenspace models that updates the mean. Experimental results show that our method is better suited for classification ap-

¹<http://www.cam-orl.co.uk/facedatabase.html>

plications than either of the other methods we tested. Additionally, our method is competitive in terms of fidelity. We therefore conclude our method gives the best general performance for incremental eigenspace computation. Fixed mean methods cannot be relied upon for classification.

SVD methods are generally regarded as more numerically stable than EVD. We intend to investigate incremental SVD with a shift in mean. However, our early investigations suggest this is not straightforward.

References

- [1] J.R. Bunch, C.P. Nielsen, D.C. Sorenson. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31:31–48, 1978.
- [2] J.R. Bunch, C.P. Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 31:111–129, 1978.
- [3] S. Chandrasekaran, B.S. Manjunath, Y.F. Wang, J. Winkler, H. Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59(5):321–332, 1997.
- [4] S. Chaudhuri, S. Sharma, S. Chatterjee. Recursive estimation of motion parameters. *Computer Vision and Image Understanding*, 64(3):434–442, 1996.
- [5] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, 9–18, 1992.
- [6] R.D. DeGroat, R. Roberts. Efficient, numerically stabilized rank-one eigenstructure updating. *IEEE Trans. acoustics, speech, and signal processing*, 38(2):301–316, 1990.
- [7] P.M. Hall, A.D. Marshall, R.R. Martin. Incrementally computing eigenspace models. Technical report, Dept. Comp. Sci., Cardiff University, 1998.
- [8] B. Moghaddam, A. Pentland. Probabilistic visual learning for object representation. *IEEE PAMI*, 19(7):696–710, 1997.
- [9] H. Murakami, B.V.K.V. Kumar. Efficient calculation of primary images from a set of images. *IEEE PAMI*, 4(5):511–515, 1982.
- [10] L. Sirovich, M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. America, A*, 4(3):519–524, 1987.
- [11] P.J.E. Vermeulen, D.P. Casasent. Karhunen-Loeve techniques for optimal processing of time-sequential imagery. *Optical Engineering*, 30(4):415–423, 1991.