Lecture 16:

Small Sample Size Problems

(Covariance Estimation)

Many thanks to Carlos Thomaz who authored the original version of these slides

Statistical Pattern Recognition

The Gaussian distribution is written:

$$p(\boldsymbol{x}) = \frac{exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}))}{(2\pi)^{n/2}\sqrt{|\Sigma|}}$$

we can use it to determine the probability of membership of a class given Σ and μ. For a given class we may also have a prior probability, using π_i for class *i*

$$p(class = \pi_i|\boldsymbol{x}) = p(\pi_i)\frac{exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})^T \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu_i}))}{(2\pi)^{n/2}\sqrt{|\Sigma_i|}}$$

The Bayes Plug-in Classifier (parametric)

Taking logs so that we do not have an infinite space the rule becomes:

Assign pattern *x* to class π_i if:

$$d_i(\boldsymbol{x}) = \max_{1\le j\le g}[-\frac{1}{2}ln|\Sigma_j| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_j})^T\Sigma_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu_j}) + ln(p(\pi_j))]$$

We can rearrange the constants to get:

$$d_i(\boldsymbol{x}) = \min_{1\le j\le g}[ln|\Sigma_j| + (\boldsymbol{x} - \boldsymbol{\mu_j})^T\Sigma_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu_j}) - 2ln(p(\pi_j))]$$

The Bayes Plug-in Classifier (parametric)

Taking logs so that we do not have an infinite space the rule becomes:

Assign pattern *x* to class π_i if:

$$d_i(\boldsymbol{x}) = \max_{1\leq j\leq g}[-\frac{1}{2}ln|\Sigma_j| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_j})^T\Sigma_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu_j}) + ln(p(\pi_j))]$$

We can rearrange the constants to get:

$$d_i(\boldsymbol{x}) = \min_{1\leq j\leq g}[ln|\Sigma_j| + (\boldsymbol{x} - \boldsymbol{\mu_j})^T\Sigma_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu_j}) - 2ln(p(\pi_j))]$$

The classification depends strongly on Σ_j

The Bayes Plug-in Classifier (parametric)

$$d_i(\boldsymbol{x}) = \min_{1 \le j \le g} [ln|\Sigma_j| + (\boldsymbol{x} - \boldsymbol{\mu_j})^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu_j}) - 2ln(p(\pi_j))]$$

Notice the relationship between the plug in Bayesian classifier and the Mahalanobis distance.

The equation has a term for the prior probability of a class and for the determinant of the covariance matrix (ie the total variance in the class), but is otherwise the same.

Problems with the Bayes Plug-in Classifier

1. The distribution of each class is assumed to be normal.

2. A separate co-variance estimate is needed for each class.

3. The inversion of the co-variance matrix is needed - computationally expensive for large variable problems

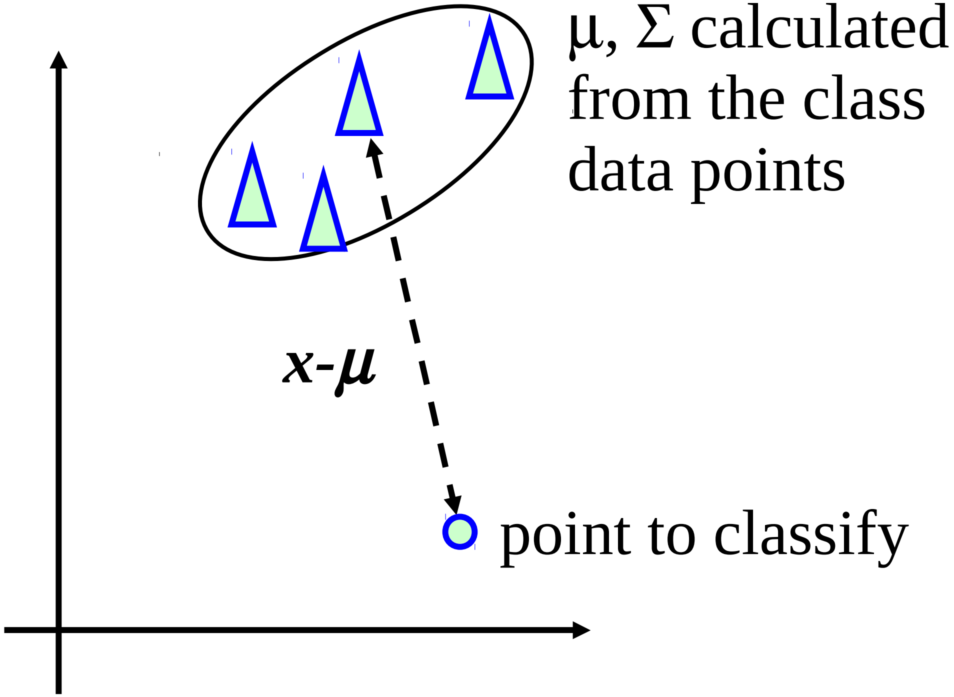
Parametric vs Non-Parametric classifiers

The parameters referred to in the Bayes plug-in classifier are the mean μ and covariance Σ which are estimated from the data points belonging to the class.

A non parametric classifier makes its probability of membership estimates from each individual point rather than from parameters calculated from them.

However, it is still desirable to have a “kernel” function to transform distance from a point to probability of membership.

Parametric Classifier illustration



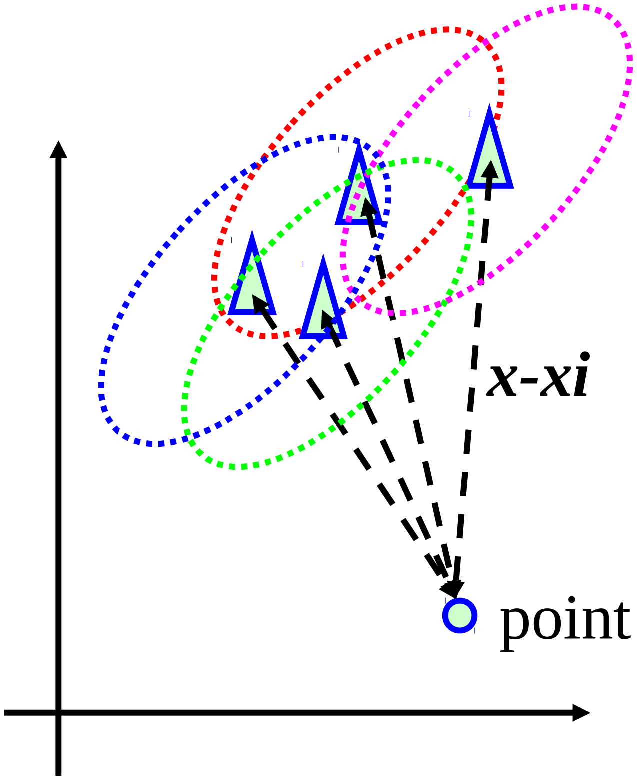
μ, Σ calculated from the class data points

x*-*μ

point to classify

The probability of membership is calculated from a class distribution with parameters μ and Σ

Non-Parametric Classifiers



kernel Σ used to calculate a probability from each point in the class.

point to classify

Probability of membership based on the average probability using each class member as mean and a given kernel covariance.

The Parzen Window Classifier (non-parametric)

$$p(\boldsymbol{x}|\pi_i) = \frac{1}{N_i}\sum_{j=1}^{N_i}\left[\frac{1}{(2\pi)^{N_i/2}\,|\Sigma_i|^{1/2}\,h_i^{N_i}}exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_{\boldsymbol{i},\boldsymbol{j}})^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{x}_{\boldsymbol{i},\boldsymbol{j}})}{2h_i^2}\right)\right]$$

h is a class specific variable called the “window” which acts a little like the total variance in the parametric classifier. A smaller *h* implies a lower variance kernel.

The Parzen Window Classifier (non-parametric)

$$p(\boldsymbol{x}|\pi_i) = \frac{1}{N_i}\sum_{j=1}^{N_i}\left[\frac{1}{(2\pi)^{N_i/2}\boxed{|\Sigma_i|}^{1/2}\;h_i^{N_i}}exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x_{i,j}})^T\boxed{\Sigma_i^{-1}}(\boldsymbol{x}-\boldsymbol{x_{i,j}})}{2h_i^2}\right)\right]$$

h is a class specific variable called the “window” which acts a little like the total variance in the parametric classifier

We need to be able to estimate Σ_i accurately to use either parametric or non-parametric classifiers

In summary: Statistical Pattern Recognition

Information about class membership is contained in the set of class conditional probability density functions (pdfs) which could be:

specified (parametric) or

calculated from data (non-parametric).

In practice, pdfs are usually based on Gaussian distributions, and calculation of the probability of membership involves the inverse of the sample group covariance matrix.

Small sample size problems

In many pattern recognition applications there is a large number of features (n) and the number of training patterns (N_i) per class may be significantly less than the dimension of the feature space.

$$N_i << n\ !$$

This means that the covariance matrix will be singular and cannot be inverted.

Examples of small sample size problems

1. Biometric identification

In face recognition we have many thousands of variables (pixels). Using PCA we can reduce these to a small number of principal component scores (possibly 50).

However, the number of training samples defining a class (person) is usually small (usually less than 10).

Examples of small sample size problems

2. Microarray Statistics

Microarray experiments make simultaneous measurements on the activity of several thousand genes - up to 500,000 probes.

Unfortunately they are expensive and it is unusual for there to be more than 50-500 repeats (data points)

Small Sample Size Problem

Poor estimates of the covariance means that the performance of classical statistical pattern recognition techniques deteriorate in small sample size settings.

$$\Sigma_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (\boldsymbol{x}_{i,j} - \bar{\boldsymbol{x}_i})(\boldsymbol{x}_{i,j} - \bar{\boldsymbol{x}_i})^T$$

→ Σ_i has n variables and N_i data points

→ Σ_i is singular when $N_i < n$

→ Σ_i is poorly estimated when N_i is not >> n

Small sample size problems

So, we need to find some method of estimating the co-variance matrix in small sample size problems.

The problem was first addressed by Fisher and has been the subject of research ever since.

Pooled Covariance Estimation

Fisher’s solution is to use a pooled covariance estimate. In the LDA method this is represented in the S_w scatter matrix, in general for g classes:

$$\begin{aligned}\Sigma_p &= \frac{1}{N-g}\sum_{i=1}^{g}(N_i-1)\Sigma_i \\ &= \frac{(N_1-1)\Sigma_1+(N_2-1)\Sigma_2+\ldots+(N_g-1)\Sigma_g}{N-g}\end{aligned}$$

Assumes equal covariance for all groups

Covariance estimators: van Ness (1980)

If we take a small sample size covariance estimate and set all the non diagonal elements to zero then the resulting matrix will generally be full rank.

$$\Sigma_i^{vn}(\alpha) = \alpha \times diag(\Sigma_i)$$

This approach retains the variance information for use in a non-parametric kernel.

α is a scalar smoothing parameter selected to maximise the classification accuracy

Diagonal Covariance Matrices

Note that:

$$diag(\Sigma_i)$$

Is almost certain to be full rank in any practical example.

If it is not this implies that there is a zero diagonal element which means that the corresponding variable does not change throughout the data. In this case it can be removed.

Shrinkage (Regularisation)

The general idea of shrinkage is to stabilise a poor matrix estimate by blending it with a stable known matrix.

For example given a singular small sample size covariance matrix Σ_i we can make it full rank by forming the estimate:

$$\Sigma_i^{id}(\alpha) = (1 - \alpha)\Sigma_i + \alpha\sigma I$$

Where α is the shrinkage parameter and σ the average variance.

As α is increased from 0 to 1 the inverse of the estimate can more readily be calculated, but the covariance information is gradually destroyed.

Shrinkage towards the pooled estimate

If a pooled estimate can be made (as it can in biometrics) then a better strategy is to shrink towards it rather than the identity matrix.

The problem in regularisation is how to choose the shrinkage parameter α.

One solution is to maximise the classification accuracy

Covariance Estimators: Friedman's RDA (1989)

Friedman proposed a composite shrinkage method *(regularised discriminant analysis)* that blends the class covariance estimate with both the pooled and the identity matrix.

Shrinkage towards the pooled matrix is done by one scalar parameter λ

$$\Sigma_i^{pool}(\lambda) = (1 - \lambda)\Sigma_i + \lambda\Sigma_p$$

Covariance Estimators: Friedman's RDA (1989)

Shrinkage towards the identity matrix is then calculated using a second scalar parameter γ.

$$\Sigma_i^{rda}(\lambda, \gamma) = (1-\gamma)\Sigma_i^{pool}(\lambda) + \gamma\sigma I$$

where σ is a scaling constant chosen to make the magnitude of the second term comparable to the variance in the data using:

$$\sigma = \frac{trace(\Sigma_i^{pool})}{n}$$

The trace of a matrix is the sum of its diagonal elements, so σ represents the average variance.

Covariance Estimators: Friedman’s RDA (1989)

We need to determine the best values for the shrinkage parameters, but this is data dependent.

The method adopted is to use an optimisation grid. We choose as large a set of values of (λ,γ) covering their range [0..1], and use hold out methods to calculate the classification accuracy at each value.

The process is very computationally intensive.

Covariance Estimators: Friedman's RDA (1989)

Notice that shrinkage towards the diagonal destroys covariance information, however:

Given a good pooled estimate we expect the shrinkage towards the identity to be small.

With a poor sample group and pooled estimate the shrinkage towards the identity matrix at least provides a full rank co-variance estimate.

Covariance Estimators: Hoffbeck (1996)

A more computationally effective approach is the leave one out covariance estimate (LOOC) of Hoffbeck:

$$\Sigma_i^{looc}(\alpha) = \begin{cases} (1-\alpha)diag(\Sigma_i) + \alpha\Sigma_i & 0 \leq \alpha \leq 1 \\ (2-\alpha)\Sigma_i + (\alpha - 1)\Sigma_p & 1 \leq \alpha \leq 2 \\ (3-\alpha)\Sigma_p + (\alpha - 2)diag(\Sigma_p) & 3 \leq \alpha \leq 3 \end{cases}$$

This is a piecewise solution. An optimisation grid is calculated, this time in one dimension, to find α

A Maximum Entropy Covariance Estimate

A new solution by Carlos Thomaz (2004)

It is based on the idea that

“When we make inferences based on incomplete information, we should draw them from that probability distribution that has the maximum entropy permitted by the information we do have.”

[E.T.Jaynes 1982]

Loss of Covariance Information

In all the methods we have seen so far there is a trade off between the amount of covariance information retained and the stabilisation of the matrix.

If we use the van Ness technique and replace

$$\Sigma_i \text{ by } diag(\Sigma_i)$$

we make the matrix full rank but we remove all the covariance information

Loss of Covariance Information

Similarly if we use shrinkage to the diagonal:

$$\Sigma_i^{diag}(\alpha) = (1-\alpha)\Sigma_i + \alpha\, diag(\Sigma_i)$$

or (worse still) to the identity

$$\Sigma_i^{id}(\alpha) = (1-\alpha)\Sigma_i + \alpha\sigma I$$

we loose some of the covariance information

Loss of Covariance Information

Even shrinking towards the pooled estimate looses covariance information.

We are averaging class specific information with pooled information hence we are loosing some of the defining characteristics of the class

Loss of Covariance Information

To understand the process in more depth let us consider mixing the class and pooled covariance matrices (similar to the first step of RDA).

$$\Sigma_i^{mix}(\alpha) = \alpha\Sigma_i + \beta\Sigma_p \quad \text{where } \beta = 1 - \alpha$$

In this approach we are looking for an optimal convex combination of α and β

Loss of Covariance Information

Now let’s consider what will happen if we diagonalise the mixture co-variance (ie find the orthonormal eigenvectors).

$$\begin{aligned}
\Phi^T \Sigma_i^{mix} \Phi &= [\lambda_1^{mix}, \lambda_2^{mix}, ..., \lambda_n^{mix}]I \\
&= \Phi^T (\alpha \Sigma_i + \beta \Sigma_p) \Phi \\
&= \alpha \Phi^T \Sigma_i \Phi + \beta \Phi^T \Sigma_p \Phi \\
&\simeq [(\alpha \lambda_1^i + \beta \lambda_1^p), (\alpha \lambda_2^i + \beta \lambda_2^p), ..., (\alpha \lambda_n^i + \beta \lambda_n^p)]I
\end{aligned}$$

Loss of Covariance Information

Think about the terms $\alpha\lambda_j^i + \beta\lambda_j^p$

The problem is that α and β are the same for all variables and therefore cause loss of information.

If λ^i is close to zero then we are simply reducing the variance contribution from the pooled matrix.

If λ^i is large then we are changing its value from the class specific value to the pooled matrix value.

A Maximum Entropy Covariance

Let an *n*-dimensional variable X_i be normally distributed with true covariance matrix Σ_i. Its entropy *h* can be written as:

$$h(X_i) = \frac{n}{2} ln(2\pi) + \frac{1}{2} ln|\Sigma_i| + \frac{n}{2}$$

which is simply a function of the determinant of Σ_i and is invariant under any orthonormal tranformation.

A Maximum Entropy Covariance

In particular we can choose the diagonal form of the covariance matrix

$$ln|\Phi^T \Sigma_i \Phi| = ln|\Lambda_i| = \sum_{k=1}^{n} ln\lambda_k$$

Thus to maximise the entropy we must select the covariance estimation of Σ_i that gives the largest eigenvalues.

A Maximum Entropy Covariance (cont.)

Considering linear combinations of S_i and S_p

$$\begin{aligned}
ln|\Sigma| &= ln|\Phi^T \Sigma_i^{mix} \Phi| \\
&= ln|\Phi^T (\alpha\Sigma_i + \beta\Sigma_p)\Phi| \\
&\simeq ln|diag[(\alpha\lambda_1^i + \beta\lambda_1^p), (\alpha\lambda_2^i + \beta\lambda_2^p), ..., (\alpha\lambda_n^i + \beta\lambda_n^p)]| \\
&= ln \prod_{k=1}^{n} (\alpha\lambda_k^i + \beta\lambda_k^p) \\
&= \sum_{k=1}^{n} ln(\alpha\lambda_k^i + \beta\lambda_k^p)
\end{aligned}$$

A Maximum Entropy Covariance (cont.)

Moreover, as the natural log is a monotonic increasing function, we can maximise

$$\sum_{k=1}^{n}(\alpha\lambda_k^i + \beta\lambda_k^p)$$

However,

$$(\alpha\lambda_k^i + \beta\lambda_k^p) \leq \max(\lambda_k^i, \lambda_k^p)$$

Therefore, we do not need to choose the best parameters α and β but simply select the maximum variances of the corresponding matrices.

A Maximum Entropy Covariance (cont.)

The Maximum Entropy Covariance Selection (**MECS**) method is given by the following procedure:

1. Find the eigenvectors Φ of $\Sigma_i + \Sigma_p$
2. Calculate the variance contribution of each matrix

$$diag(\Phi^T \Sigma_p \Phi) = [\lambda_1^p, \lambda_2^p, ..., \lambda_n^p]I$$

$$diag(\Phi^T \Sigma_i \Phi) = [\lambda_1^i, \lambda_2^i, ..., \lambda_n^i]I$$

3. Form a new variance matrix based on the largest values

$$Z_i^{max} = [max(\lambda_1^i, \lambda_1^p), max(\lambda_2^i, \lambda_2^p), ..., max(\lambda_n^i, \lambda_n^p)]I$$

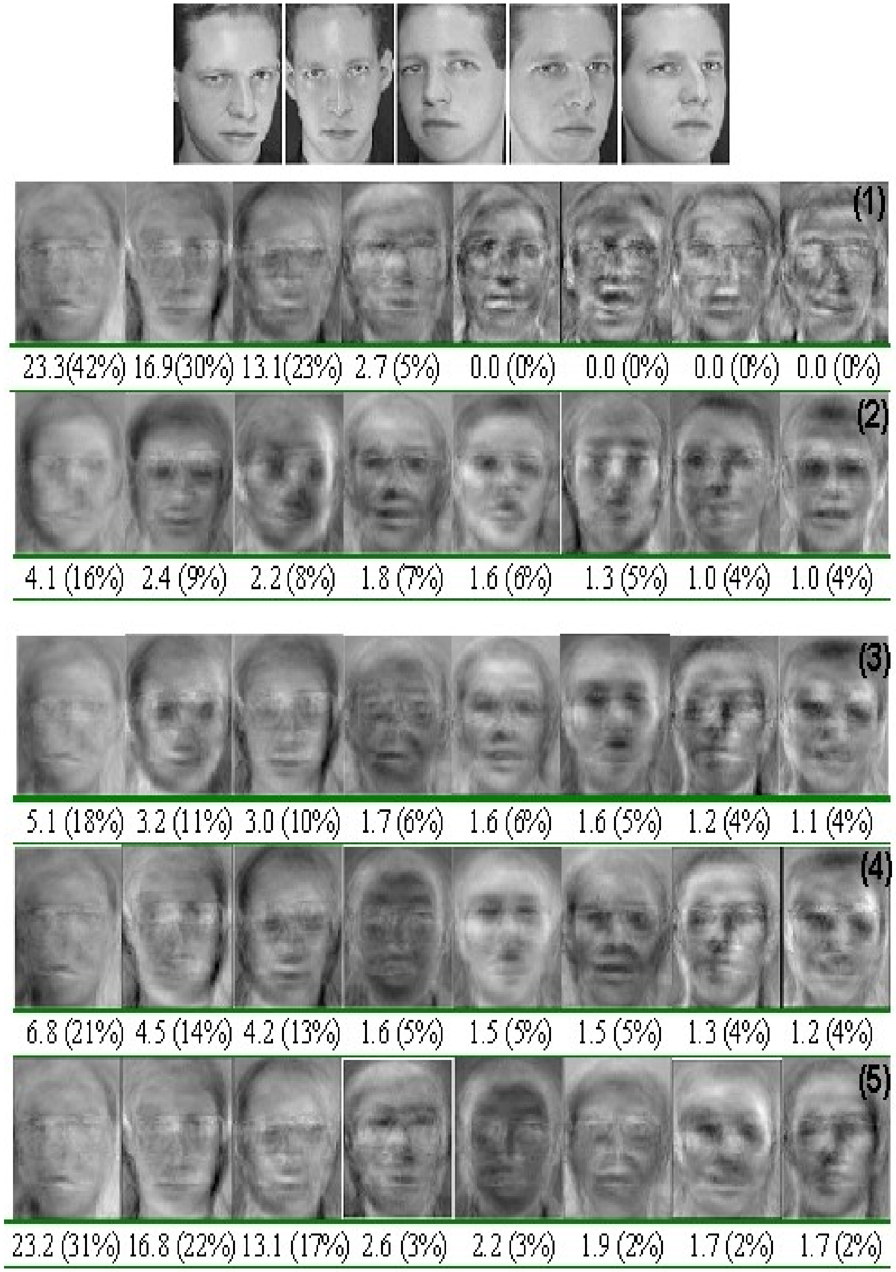
4. Inverse project the matrix to find the MECS covaraince estimate:

$$\Sigma_i^{mecs} = \Phi Z_i^{max} \Phi^T$$

Visual Analysis

The top row shows the 5 image training examples of a subject and the subsequent rows show the image eigenvectors (with the corresponding eigenvalues below) of the following covariance matrices:

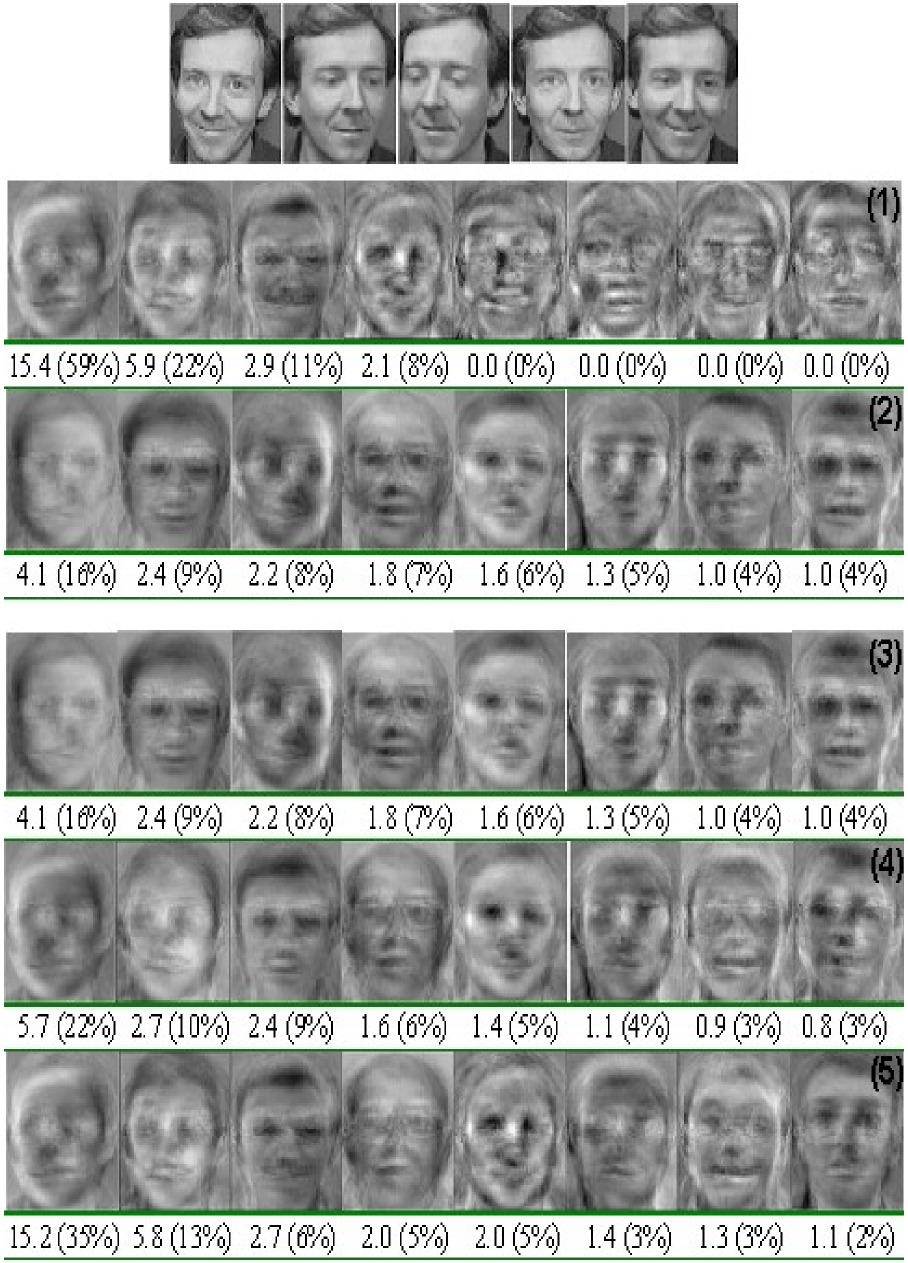
(1) sample group;
(2) pooled;
(3) maximum likelihood (rda);
(4) maximum classification mixture (looc)
(5) maximum entropy mixture.



Visual Analysis (cont.)

The top row shows the 5 image training examples of a subject and the subsequent rows show the image eigenvectors (with the corresponding eigenvalues below) of the following covariance matrices:

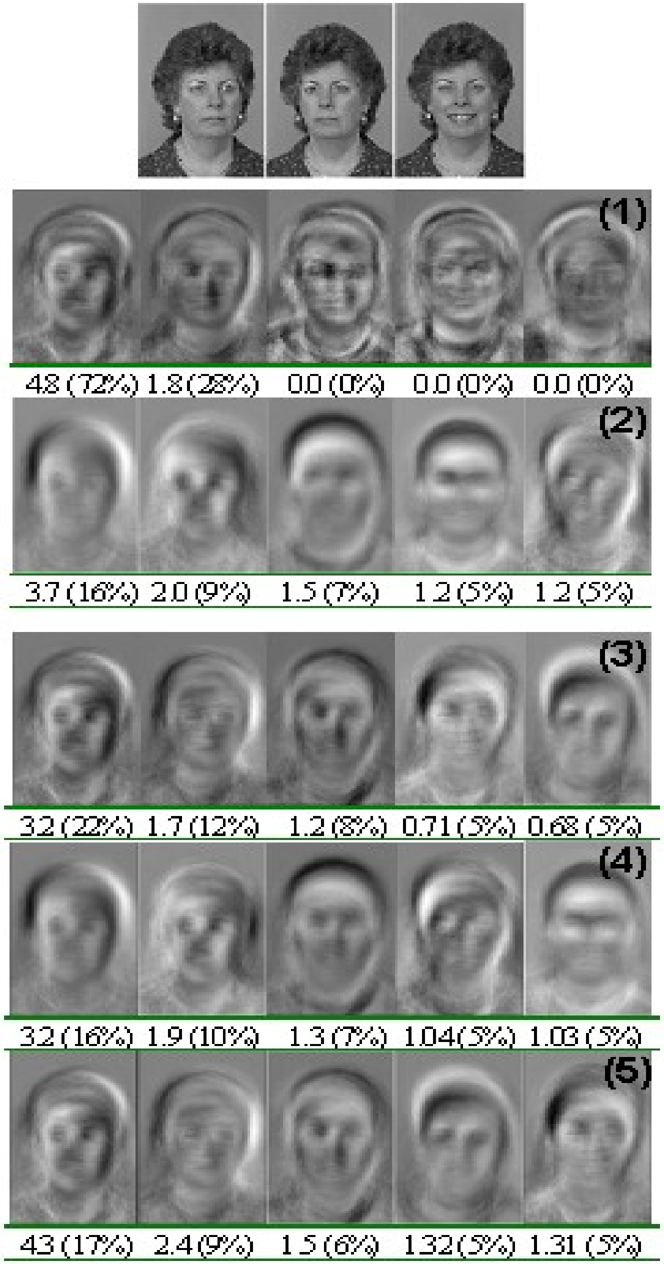
(1) sample group;
(2) pooled;
(3) maximum likelihood (rda);
(4) maximum classification mixture (looc);
(5) maximum entropy mixture.



Visual Analysis (cont.)

The top row shows the 3 image training examples of a subject and the subsequent rows show the image eigenvectors (with the corresponding eigenvalues below) of the following covariance matrices:

(1) sample group;
(2) pooled;
(3) maximum likelihood (rda);
(4) maximum classification mixture (looc);
(5) maximum entropy mixture.



Visual Analysis (cont.)

The top row shows the 3 image training examples of a subject and the subsequent rows show the image eigenvectors (with the corresponding eigenvalues below) of the following covariance matrices:

(1) sample group;
(2) pooled;
(3) maximum likelihood (rda);
(4) maximum classification mixture (looc);
(5) maximum entropy mixture.