# Lecture 16: Small Sample Size Problems and Covariance Estimation

## The Paramteric Bayes "plug-in" classifier.

The most commonly used method is statistical pattern recognition is the Bayes "plug-in" classifier. The usual kernal that is plugged in is the multivariate Gaussian distribution is written:

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

We can use it to determine the probability of membership of a class given $\Sigma$ and $\mu$. It is called "plug-in" to indicate that it is one possible kernel that could represent likelihood information, and it is plugged into the normal Bayes formalism. For a given class denoted $\pi_i$, with prior probability $p(\pi_i)$:

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

In a classification example it is common to estimate a distance for a unclassified point to each possible class, and then choose the nearest. If we take logs so the rule becomes: Assign pattern $x$ to class $\pi_i$ if:

$$d_i(\boldsymbol{x}) = \max_{1\le j\le g}[-\frac{1}{2}ln|\Sigma_j| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_j})^T\Sigma_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu_j}) + ln(p(\pi_j))] \tag{3}$$

If we change the sign to make the result always positive and re-arrange the constants we get a distance metric and assign the pattern $x$ to class $\pi_i$ if:

$$d_i(\boldsymbol{x}) = \min_{1\le j\le g}[ln|\Sigma_j| + (\boldsymbol{x}-\boldsymbol{\mu_j})^T\Sigma_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu_j}) - 2ln(p(\pi_j))] \tag{4}$$

Notice that the second term is the Mahalanobis distance. In many applications, such as face recognition the first term (total class variance) and the last (prior probability) are unlikely to vary much with different classes.

More importantly, notice that the classification depends strongly on $\Sigma$. This dependence immediatly creates two difficulties. Firstly, a separate co-variance estimate is needed for each class. This may be difficult to estimate with any accuracy if the class has only a few examples in it - as is the case for face recognition. Secondly, the inversion of the co-variance matrix is needed. In many problems we have a large number of variables and this means that the inversion is at best computationally expensive. At worst it may be impossible to compute if the co-variance matrix is not of full rank.

## Parametric vs Non-Parametric classifiers

The parameters referred to in the Bayes plug-in classifier are the mean $\mu$ and covariance $\Sigma$ which are estimated from the points known to belong to the class (Figure 1). A non parametric classifier makes its probability of

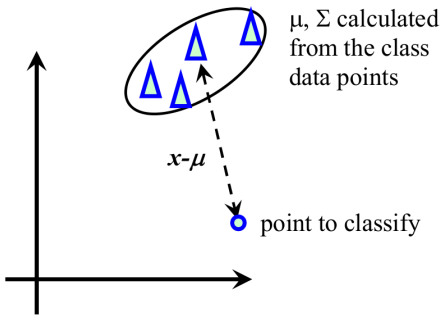


Figure 1: Parametric Classification

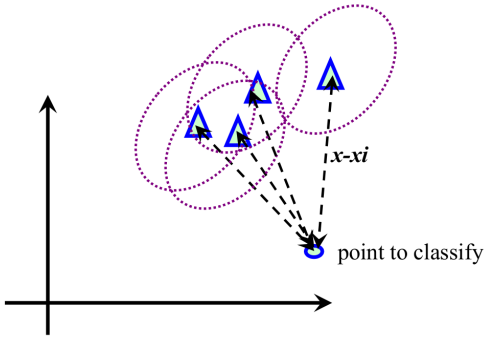


Figure 2: Non Parametric Classification

membership estimates from each individual point rather than from parameters calculated from them. However, it is still necessary to have a kernel function to transform distance from a point to probability of membership.

The probability of membership is calculated using the average distance from each class member given a kernel covariance (Figure 2).

The (non-parametric) Parzen Window Classifier is defined by the equation:

$$p(\boldsymbol{x}|\pi_i) = \frac{1}{N_i}\sum_{j=1}^{N_i}\left[\frac{1}{(2\pi)^{N_i/2}\ |\Sigma_i|^{1/2}\ h_i^{N_i}} exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x_{i,j}})^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{x_{i,j}})}{2h_i^2}\right)\right] \tag{5}$$

The variable $h$ is a class specific variable called the "window" which acts a little like the total variance in the parametric classifier. The covariance in the equation is the plug in kernel, but in most applications of interest we will estimate it from the data points of the class. Thus the classifier is non parametric in the sense that it does not use the class mean. We see that it is necessary to be able to estimate $\Sigma_i$ accurately for both the parametric and non-parametric classifiers.

In summary, information about class membership is contained in the set of class conditional probability density functions (pdfs). In practice, pdfs are usually based on Gaussian distributions, and calculation of the probability of membership involves the inverse of 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. This means that the covariance matrix will be singular and cannot be inverted. There are many important examples of small sample size problems. For example, 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). Microarray experiments have recently been developed and can make simultaneous measurements on the activity of several thousand genes. These are crucial in the understanding of gene regulatory networks, and the development of therapies for diseases caused by genetic defects, such as cancer. Unfortunately microarray experiments are expensive and it is unusual for there to be more than 50 repeats ( data points) for several thousand variables (genes). Poor estimates of the covariance means that the performance of classical statistical pattern recognition techniques deteriorates in these important small sample size settings.

If the number of data points is less than the number of variables, as it is in the examples above, the covariance matrix will be singular, and therefore cannot be inverted. In these cases we need to find some method of estimating a full rank covariance matrix to calculate an inverse. The problem was first addressed by Fisher and has been the subject of research ever since. Fisher noted that in many cases it would be reasonable to pool the data of all classes, and use the pooled estimate in place of the individual class covariances. A good example where this is appropriate is face recognition. A face data base may contain only ten examples per subject (class) but in total may have 500 subjects, giving a total of 5000 images. If the number of variables is reduced to 50 using PCA then a full rank covariance matrix can be found from the entire data base, and it is not unreasonable to use this for each individual class. In the LDA method this technique is adopted in the $S_w$ scatter matrix. In general for $g$ classes the pooled estimate is:

$$\Sigma_p = \frac{1}{N-g}\sum_{i=1}^{g}(N_i-1)\Sigma_i \tag{6}$$

$$= \frac{(N_1-1)\Sigma_1 + (N_2-1)\Sigma_2 + ... + (N_g-1)\Sigma_g}{N-g} \tag{7}$$

A pooled estimate may not be appropriate for all problems - microarray statistics is one - and may not solve the problem if the total number of data points is small. For these cases other approaches are needed.

### Diagonal 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. In practice if there are zeros on the diagonal, then this would indicate a variable having zero variance. This would mean that it does not change throughout the data set,

and therefore can be eliminated as it has no discriminatory power. The van Ness estimator, intended for non-parametric classifiers, therefore simply set all the off diagonal elements to zero. We write this as follows:

$$\Sigma_i^{vn}(\alpha) = \alpha \times diag(\Sigma_i) \tag{8}$$

$\alpha$ is a scalar smoothing parameter selected to maximise the classification accuracy. The approach preserves the size of variance, but clearly will only work well if the original co-variance is low.

**Regularisation (or shrinkage) methods**

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 $\Sigma_i$ we can make it full rank by forming the estimate:

$$\Sigma_i^{id}(\alpha) = (1-\alpha)\Sigma_i + \alpha\sigma I \tag{9}$$

Where $\alpha$ is the shrinkage parameter and $\sigma$ the average variance. As $\alpha$ is increased from 0 to 1 the inverse of the estimate can more readily be calculated, but the more the covariance information is destroyed. If a pooled estimate can be made (as it can in biometrics) then a better strategy is to shrink the sample group covariance towards the pooled estimate rather than the identity matrix. The problem in regularisation is how to choose the shrinkage parameter $\alpha$. One solution is to use statistical testing to maximise the classification accuracy.

Friedman (1989) proposed a composite shrinkage method called regularised discriminant analysis (RDA) that blends the class covariance estimate with both the pooled and the identity matrix. Shrinkage towards the pooled matrix is done (in essence) by one scalar parameter $\lambda$.

$$\Sigma_i^{pool}(\lambda) = (1-\lambda)\Sigma_i + \lambda\Sigma_p \tag{10}$$

Shrinkage towards the identity matrix is then calculated using a second scalar parameter $\gamma$.

$$\Sigma_i^{rda}(\lambda,\gamma) = (1-\gamma)\Sigma_i^{pool}(\lambda) + \gamma\sigma I \tag{11}$$

where $\sigma$ is a scaling constant that makes the magnitude of the second term comparable to the variance in the data using:

$$\sigma = \frac{trace(\Sigma_i^{pool})}{n} \tag{12}$$

The trace of a matrix is the sum of its diagonal elements, so $\sigma$ represents the average variance. 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 $(\lambda,\gamma)$ covering their range [0..1], and use hold out methods to calculate the classification accuracy at each value. The process is very computationally intensive. The RDA is designed to give the best results in all cases. Given a good pooled estimate we expect the shrinkage towards the identity to be small. However, with poor estimates for both the sample group and pooled covariance, the shrinkage towards the identity matrix at least provides a full rank co-variance estimate.

The above formulation is simplified a little to clarify the process. Friedmans original formulation used a more complex normalisation for the first shrinkage.

$$\Sigma_i^{pool}(\lambda) = \frac{(1-\lambda)(N_i-1)\Sigma_i + \lambda(N-g)\Sigma_p}{(1-\lambda)N_i + \lambda N}$$

First, the co-variances are converted to scatter matrices by multiplying by $(N_i-1)$ and $(N-g)$ respectively. After blending the re-normalisation is done using the same combination of $N_i$ and $N$.

A more computationally effective regularisation approach is the leave one out covariance estimate (LOOC) proposed by Hoffbeck.

$$\Sigma_i^{looc}(\alpha) = \begin{cases} (1-\alpha)diag(\Sigma_i) + \alpha\Sigma_i & 0 \le \alpha \le 1 \\ (2-\alpha)\Sigma_i + (\alpha-1)\Sigma_p & 1 < \alpha \le 2 \\ (3-\alpha)\Sigma_p + (\alpha-2)diag(\Sigma_p) & 2 < \alpha \le 3 \end{cases} \tag{13}$$

This is a piecewise solution covering several possibilities. In the most usual case $(1 < \alpha \le 2)$ we shrink the class covariance matrix towards the pooled estimate. However for cases where both the pooled and the class covariance matrix are not full rank we allow the optimisation to choose between shrinking the class covariance towards its diagonal $(0 \le \alpha \le 1)$ or shrinking the pooled estimate towards its diagonal $(2 < \alpha \le 3)$. An optimisation grid is calculated, this time in one dimension, to find the optimal $\alpha$.

## Loss of Covariance Information

A new solution to the problem of finding the best estimate of covariance for small sample size problems was proposed by Carlos Thomaz (2004). It is based on the ideas of E.T.Jaynes who pioneered the theory of entropy in the 1950s. Quoting Jaynes:

*"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."*

In all the methods we have seen so far there is a trade off between the amount of covaraince information retained and the stabilisation of the matrix. In every case we are loosing covariance information, and therefore violating the principal of Jaynes. For example, if we use the van Ness technique and replace $\Sigma_i$ by $diag(\Sigma_i)$ we ensure that the matrix full rank but we destroy all the covariance information. Similarly if we use shrinkage and either to the diagonal or (worse still) to the identity we dilute the covariance information, making the variance larger in proportion. Even shrinking towards the pooled estimate looses covariance information. To see why this is so, consider the following. Suppose we are mixing two matrices, as is done in the RDA and LOOC methods discussed above.

$$\Sigma_i^{mix}(\alpha) = \alpha\Sigma_i + \beta\Sigma_p \quad \text{where } \beta = 1 - \alpha \tag{14}$$

Consider what happens if we diagonalise the resulting mixture covariance matrix (as we do in the PCA process by finding the orthonormal eigenvectors of the matrix).

$$\begin{aligned}
\Phi^T\Sigma_i^{mix}\Phi &= [\lambda_1^{mix}, \lambda_2^{mix}, ..., \lambda_n^{mix}]I && (15)\\
&= \Phi^T(\alpha\Sigma_i + \beta\Sigma_p)\Phi && (16)\\
&= \alpha\Phi^T\Sigma_i\Phi + \beta\Phi^T\Sigma_p\Phi && (17)\\
&\simeq [(\alpha\lambda_1^i + \beta\lambda_1^p), (\alpha\lambda_2^i + \beta\lambda_2^p), ..., (\alpha\lambda_n^i + \beta\lambda_n^p)]I && (18)
\end{aligned}$$

Note that $\lambda^i$ are not exactly the eigenvalues of $\Sigma_i$, though in practice we expect them to be very close to the eigenvalues. With small sample size problems, most of the $\lambda^i$ values will be zero. This is because the class covariance matrix will be computed from a small number of samples, and the number of non-zero eigenvalues will be less than or equal to that number of samples. For all cases where the values of $\lambda^i$ are low or zero we are simply reducing the variance contribution from the pooled matrix. Conversely in cases were a $\lambda^i$ value is large then we are changing its value from the class specific value to the pooled matrix value, thus we are loosing information about the specific class. The problem is that the blending parameters $\alpha$ and $\beta$ are the same for all variables and therefore cause loss of information. We are averaging class specific information with pooled information hence we are loosing some of the defining characteristics of the class

## A Maximum Entropy Covariance Estimate

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

$$h(X_i) = \frac{n}{2}ln(2\pi) + \frac{1}{2}ln|\Sigma_i| + \frac{n}{2} \tag{19}$$

which is simply a function of the determinant of $\Sigma_i$ and is invariant under any orthonormal tranformation. So we can choose to maximise the equation with the diagonal covariance matrix (that would be determined by the PCA transformation). This means maximising:

$$ln|\Phi^T\Sigma_i\Phi| = ln|\Lambda_i| = \sum_{k=1}^{n} ln\lambda_k \tag{20}$$

In other words we must select the covariance estimation of $\Sigma_i$ that gives the largest eigenvalues. If we now think again about the mixture covariance matrix, and take logs:

$$\begin{aligned} ln|\Sigma| &= ln|\Phi^T \Sigma_i^{mix} \Phi| && (21)\\ &= ln|\Phi^T(\alpha\Sigma_i + \beta\Sigma_p)\Phi| && (22)\\ &\simeq ln|diag[(\alpha\lambda_1^i + \beta\lambda_1^p), (\alpha\lambda_2^i + \beta\lambda_2^p), ..., (\alpha\lambda_n^i + \beta\lambda_n^p)]| && (23)\\ &= ln \prod_{k=1}^{n} (\alpha\lambda_k^i + \beta\lambda_k^p) && (24)\\ &= \sum_{k=1}^{n} ln(\alpha\lambda_k^i + \beta\lambda_k^p) && (25)\\ & && (26) \end{aligned}$$

Now, we observe that for any convex combination $(\alpha + \beta = 1)$:

$$(\alpha\lambda_k^i + \beta\lambda_k^p) \leq \max(\lambda_k^i, \lambda_k^p) \tag{27}$$

Therefore, to achieve the maximum entropy we do not try to determine the best parameters $\alpha$ and $\beta$ but simply select the maximum variances of the corresponding matrices.

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

1. Find the eigenvectors of $\Sigma_i + \Sigma_p$
2. Calculate the variance contribution of the individual matrices

$$diag(\Phi^T \Sigma_i \Phi) = [\lambda_1^i, \lambda_2^i, ..., \lambda_n^i]I \tag{28}$$

$$diag(\Phi^T \Sigma_p \Phi) = [\lambda_1^p, \lambda_2^p, ..., \lambda_n^p]I \tag{29}$$

3. Form new variance matrix based on the largest values

$$Z_i^{max} = [max(\lambda_1^i, \lambda_1^p), max(\lambda_2^i, \lambda_2^p), ..., max(\lambda_n^i, \lambda_n^p)]I \tag{30}$$

4. Form the MECS estimate of the covariance by the inverse projection

$$\Sigma_i^{mecs} = \Phi Z_i^{max} \Phi^T \tag{31}$$

The MECS estimator is very much faster to compute that any other, since it involves only selection. There is no optimisation process. It has been found to outperform, or in the limit equal the performance of other methods.