Name: Solution

Problem	Points	Score
1(a)	15	
1(b)	15	
1(c)	10	
2(a)	10	N/A
2(b)	10	N/A
2(c)	10	N/A
3(a)	10	
3(b)	10	
3(c)	10	
Total	70	

Notes:

(1) The exam is closed books and notes except for one double-sided sheet of notes.

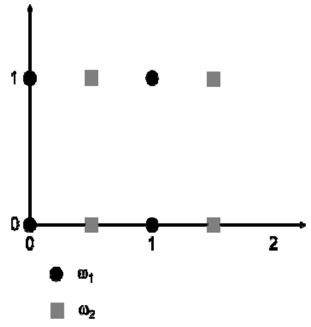
(2) Please indicate clearly your answer to the problem.

(3) If I can’t read or follow your solution, it is wrong and no partial credit will be awarded.

Problem No. 1: Consider a two-category two-dimensional classification problem in which you are given the following training data: [ω_1: {(1,1), (0, 1), (1,0), (0,0}] and [ω_2: {(0.5, 0), (0.5, 1), (1.5, 1), (1.5, 0)].

(a) Using maximum likelihood principles, design a classifier and compute the associated probability of error, P(E).

Let’s plot the data to get some insight:

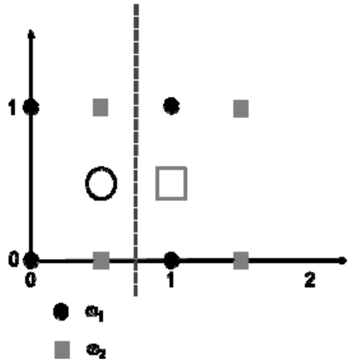


The easy way out on this problem is to postulate a model, call it “Joe’s model” that can form decision surfaces based on training data points. One can see that you could easily get 0% error on this simple data set using such a model. Soon we will learn that this is called a Support Vector Machine.

However, using what we learned in Chapters 2 and 3, let’s assume a Gaussian model for the data in each class (note that we need not assume each class has the same underlying model). Our justification is to cite Occam’s Razor: without additional knowledge, we will use simplest model possible.

Our strategy for maximum likelihood training will be to estimate the mean and covariance of each set using the well-known ML equations (which we derived in class). Our strategy for maximum likelihood classification will be to use a decision surface that is a hyperplane halfway between the means. Why? The covariances for each class are equal. For ML classification, we do not worry about the priors. So the decision surface is essentially determined by the means, and is essentially a threshold on the likelihood function, which in this 2D case will be a line (for the Gaussian assumption).

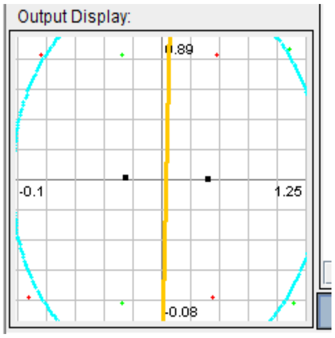
The means are easy to compute: $\mathbf{m}_1 = \frac{1}{4}\sum_{i=1}^{4}\mathbf{x}_i$ =(0.5,0.5) (note that this is easy to determine from inspection since the mean lies at the center of the square defined by the four points. Similarly, $\mathbf{m}_2 = \frac{1}{4}\sum_{i=1}^{4}\mathbf{x}_i$ =(1,0.5) .



The decision surface will consist of a vertical line halfway between the means. This will result in an error rate of 50%: P(E) = 0.5.

Note that the support regions for the two-dimensional Gaussian distributions that we estimated as models for this data will be circles because the variance is equal in all dimensions.

As a check of our calculations, let’s use the Java Pattern Recognition applet in class-independent PCA mode (why?). To the right is the result.



What will happen if you use class-dependent PCA?

(b) Using Bayesian principles, design a classifier and compute the associated probability of error, P(E).

The challenge here is to estimate the prior probabilities from the data. Fortunately, in the absence of other information, all we can assume is that the classes are equally likely since each has the same number of points.

If the priors are equal, the decision surface will not change.

(c) Assume you are told class 2 is 10 times more likely than class 1, explain how your results for (a) and (b) would change.

ML classification ignores the priors, so there is no change for (a). For (b), the decision surface will shift away from the more likely class and towards the less likely class. We can confirm this by noting that the discriminant function in the multivariate Gaussian case (see lecture 3, slide 18) is governed by the equation:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathbf{t}} \Sigma_i{}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i)$$

The equation of the surface (lecture 3, slide 20), is proportional to $2\sigma^2 \ln\frac{P(\omega_2)}{P(\omega_1)}$.

Problem No. 2: Consider a two-state model of a coin toss: $\boldsymbol{A} = \begin{bmatrix} 0.25 & 0.75 \\ 0.5 & 0.5 \end{bmatrix}$, $\boldsymbol{B} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$, and $\boldsymbol{\pi} = [0.5 \quad 0.5]$.

(a) Compute the probability that a sequence of two heads (e.g., HH) can be observed, or generated from this model.

You can work this in preparation for the final exam.

(b) What is the most likely state sequence that produced this sequence of "HH"?

You can work this in preparation for the final exam.

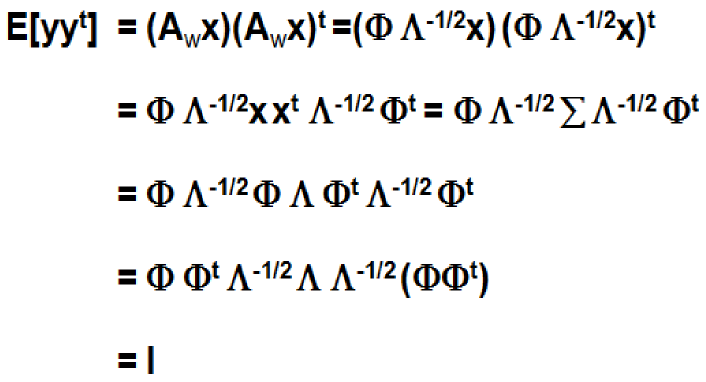
(c) Given a training sequence of "HTHT", reestimate the transition probabilities. Does this result make sense? Explain.

You can work this in preparation for the final exam.

Problem No. 3: Consider the problem of classifying two-dimensional data that is known not to be Gaussian and known not to have an identity covariance matrix.

(a) Consider a linear transformation of the data using PCA: $\boldsymbol{y} = \boldsymbol{Ax}$**.** Demonstrate that the covariance of $\mathbf{y}$ is an identity matrix if $\mathbf{A}$ is estimated using PCA.

From lecture 3, slide 16:



$$
\begin{aligned}
\mathbf{E[yy^t]} &= (\mathbf{A_w x})(\mathbf{A_w x})^t = (\Phi\,\Lambda^{-1/2}\mathbf{x})\,(\Phi\,\Lambda^{-1/2}\mathbf{x})^t \\
&= \Phi\,\Lambda^{-1/2}\mathbf{x}\,\mathbf{x}^t\,\Lambda^{-1/2}\,\Phi^t = \Phi\,\Lambda^{-1/2}\,\Sigma\,\Lambda^{-1/2}\,\Phi^t \\
&= \Phi\,\Lambda^{-1/2}\,\Phi\,\Lambda\,\Phi^t\,\Lambda^{-1/2}\,\Phi^t \\
&= \Phi\,\Phi^t\,\Lambda^{-1/2}\,\Lambda\,\Lambda^{-1/2}\,(\Phi\Phi^t) \\
&= \mathbf{I}
\end{aligned}
$$

(b) Explain the difference between PCA and Linear Discriminant Analysis. How would you determine which transformation was more appropriate for your data set?

PCA rotates the data into a space in which the transformed data has an identity covariance matrix (as proved in part (a)). Since this rotation uses all the data to compute the covariance, PCA models the directions in which the variance is greatest.

LDA maximizes the ratio of inter-class scatter to intra-class scatter. The former tends to represent the direction of discrimination, while the latter tends to represent the directions of variance. By maximizing this ratio, the directions in which the classes differ should be accentuated.

By labeling some representative data, you can first analyze the data using PCA and then determine if the resulting transformation makes sense in disambiguating the classes. You might also look at how a line connecting the means compares to the direction of variance. You can also look at the difference between the class-independent covariance and the class-dependent covariances (and the means). Of course, the best way is to run an experiment!

(c) Describe the difference between class-dependent and class-independent PCA for a two-class classification problem. Explain the difference in decision surfaces that can be achieved by the two schemes.

Class-dependent PCA computes a covariance matrix for each class, and uses a likelihood computation to make the class assignment. A likelihood is calculated for each class based on the class-dependent mean and covariance. It is still a maximum likelihood approach, but the decision surface can be slightly more complex than for a class-independent approach. In the 2D case, the decision surface changes from a line to a parabola (for the two-class case), or a series of parabolas (for the multi-class case). We have seen that the performance of class-dependent PCA approaches LDA.