1. For the pth training pattern, let the discriminants from problem 1 and the correct class numbers be denoted as \( d_i(p) \) and \( i_c(p) \) respectively, where \( 1 \leq i \leq N_c \) and \( 1 \leq p \leq N_v \). In order to map \( d_i(p) \) to the type \( B_3 \) discriminant \( d_i'(p) = a_i + b_i \cdot d_i(p) \), we can minimize the error functions \( E(i) \), defined as

\[
E(i) = \frac{1}{N_v} \sum_{p=1}^{N_v} [t_{pi} - (a_i + b_i \cdot d_i(p))]^2
\]

(a) Define \( t_{pi} \).
(b) In terms of \( i \), give the two equations that must be solved for \( a_i \) and \( b_i \). Define the coefficients \( r(m,n) \) and \( c(m) \) in the equations.

2. Assume we have \( N=4 \) features available in a pattern recognition application. The available features are \( x = \{x_1, x_2, x_3, x_4\} \) with \( x_1 = u+n \), \( x_2 = v+2n \), \( x_3 = u+n \), \( x_4 = n \). Here, \( u \) and \( v \) are statistically independent and useful for classification, but \( n \) represents noise independent of the class.
(a) Give the optimal subset of size \( N_1 = 3 \).
(b) If a transformation approach \( (A \cdot x = z) \) is used to find the same subset as in part (a), give the matrix \( A \).
(c) Give the optimal transformation matrix \( A \).
(d) Which approach (subsetting or transformation) is best and why?

3. Before applying discriminant fusion we want to put all \( K \) discriminants into the type \( B_3 \) form. The type \( B_3 \) form for the Bayes-Gaussian discriminant is

\[ b_i = (x - m_i)^T A_i (x - m_i) - 2 \ln(P_i) + \ln(|C_i|), \]

where we know \( N \) and \( P_i \).
(a) In terms of the symbols \( N \), \( P_i \), and \( b_i \), how do we calculate the type \( B_1 \) Bayes-Gaussian discriminant \( u_i \)?
(b) Continuing part (a), give \( f(x) \) in terms of \( u_i \). Also give the type \( B_3 \) discriminant \( d_i \) in terms of \( u_i \) and \( f(x) \).
(c) For a single pattern, assume that we have \( N_c \) nearest neighbor classifier distances \( E_i \). If \( E_{\text{max}} \) and \( E_{\text{min}} \) denote the maximum and minimum values of \( E_i \), give two general ways to produce discriminants \( e_i \) which are between 0 and 1, using the symbols \( E_i \), \( E_{\text{max}} \) and \( E_{\text{min}} \). Here a larger value of \( e_i \) denotes a more probable class.
(d) Unfortunately, the \( e_i \) values from part (c) may not sum up to equal 1. How can we process the \( e_i \) values to produce a new discriminant \( f_i \), which sums to 1?
4. Here we continue problem 3. Assume that we have \( N_c \) discriminants \( f_i \), from problem 3(d) which sum to 1.
(a) Let \( f_{\text{max}} \) and \( f_{\text{min}} \) denote the largest and smallest values of \( f_i \) and assume that \( f_{\text{max}} < 1 \) and \( f_{\text{min}} > 0 \). Unfortunately, it is possible that \( f_{\text{max}} \) and \( f_{\text{min}} \) have very similar values, so that the user cannot have confidence in the classifier’s decision. We want to process the \( f_i \) values into the final discriminants as \( d_i = a + b \cdot f_i \). If the \( d_i \) values sum up to 1, give the equation that must be satisfied, in terms of \( a \), \( b \), and \( N_c \).
(b) Find an expression for \( d_{\text{min}} \) in terms of \( a \), \( b \), and \( f_{\text{min}} \). Give final values for \( a \) and \( b \) so that \( d_{\text{min}} = 0 \).
(c) Find an expression for \( d_{\text{max}} \) in terms of \( a \), \( b \), and \( f_{\text{max}} \). Using the equation from part (a) and the \( d_{\text{max}} \) expression, give the inequality, in terms of “\( a \)”, that must be satisfied so that \( d_{\text{max}} < 1 \).

5. Center vectors for a nearest neighbor classifier can be stored as \( \{ m_{ik} \} \) for \( 1 \leq i \leq N_c \), \( 1 \leq k \leq K(i) \) and as \( \{ m_{ik}, i_c(k) \} \) for \( 1 \leq k \leq K \). Here \( K = \sum K(i) \) and \( i_c(k) \) is the correct class number for the \( k \)th center vector. When partial distances are used in the \( \{ m_{ik}, i_c(k) \} \) form, the resulting minimum distance \( d_i \) for the \( i \)th class will be incorrect and very similar for each class, making it difficult to convert them to the \( B_3 \) form, useful for discriminant fusion. Using the \( \{ m_{ik} \} \) form for the cluster center vectors, describe, in words, a partial distance approach which allows us to calculate the discriminants \( d_i \) with no error.

6. Given training data of the form \( \{ x_p, i_c(p) \} \), we want to find a good feature subset of size \( N_1 \) for \( N_1 \) between 1 and \( N-1 \). We know that Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) require the same numbers of evaluations of the J function, for this task. However, that does not prove that both algorithms are equally efficient. Let \( J(N_1) \) denote a J function evaluation for one subset of size \( N_1 \).
(a) How many \( J(1) \) and \( J(2) \) evaluations are used in the first and second steps respectively of SFS ?
(b) How many \( J(N-1) \) and \( J(N-2) \) evaluations are used in the first and second steps respectively of SBS ?
(c) Find \( N_{\text{SFS}}(N_1) \), the number of \( J(N_1) \) evaluations used by SFS.
(d) Find \( N_{\text{SBS}}(N_1) \), the number of \( J(N_1) \) evaluations used by SBS.
(e) Which algorithm is most efficient ? (Hint: compare \( N_{\text{SFS}}(N-1) \) to \( N_{\text{SBS}}(N-1) \) )
Let $o(n)$ denote the index of the $n$th chosen feature from Sequential Forward Selection. $oc(n)$, for $1 \leq n \leq N_{oc}$, is the index of the $n$th feature that has not yet been chosen, where $N_{oc}$ is the number of yet to be chosen features. The pseudocode below performs forward selection. The Create function generates a data file using features $o(1)$ through $o(N_{i})$. The Class function designs a classifier using the datafile, and returns the $P_e$.

(a) In the first loop, give values for $U$ and $V$ so that the $oc(n)$ array is correctly initialized.
(b) In the second loop, give a value for $W$
(c) In the third loop, give a value for $X$
(d) In the last loop, give a value for $Y$.

$N_{oc} = U$
For $1 \leq n \leq N_{oc}$
$oc(n) = V$
End

For $1 \leq N_{i} \leq N-1$
o($N_{i}$) = $W$
n$_{min}$=1
Call Create(o, $N_{i}$, datafile)
Call Class(datafile, $N_{i}$, $P_{e}$)
P$_{emin}$ = $P_{e}$
o$_{min}$ = o($N_{i}$)

For $X \leq n \leq N_{i}$
o($N_{i}$) = oc(n)
Call Create(o, $N_{i}$, datafile)
Call Class(datafile, $N_{i}$, $P_{e}$)
If($P_{e} < P_{emin}$) Then
$P_{emin}$ = $P_{e}$
n$_{min}$ = n
o$_{min}$ = o($N_{i}$)
Endif
End

o($N_{i}$) = o$_{min}$
P$_{e}(N_{i})$ = $P_{emin}$
For $n_{min} \leq n \leq N_{oc} - 1$
\[ oc(n) = Y \]
End
\[ N_{oc} = N_{oc} - 1 \]
End

Call Create(o, N, datafile)
Call Class(datafile, N, P_e)
\[ P_{ce}(N) = P_e \]