COGS 118A: ASSIGNMENT 5

Problem 1. Linear discriminant classification

(a) Show that the least-error function for linear discriminant classification has the following equivalent forms:

\[ E(W) = \sum_{j=1}^{k} \sum_{i=1}^{n} (w_j^T x_i + w_{j0} - \delta_{c,j})^2 = Tr\{(XW - T)^T(XW - T)\} \]

where each row of \( W^T \) is \([w_{j0} \ w_j^T]\) for each of the discriminant functions, each row of \( X \) is \([1 \ x_i^T]\) for each of the data points, the \( ij \)th element of \( T \) is \( \delta_{c,j} \) for data point \( i \) and discriminant function \( j \), and \( \delta_{ab} \) is the Kronecker delta function that evaluates to 1 when \( a = b \) and 0 otherwise.

(b) Use a Lagrange multiplier to enforce the constraint \( w^T w = 1 \), while maximizing class mean separation with respect to \( w \) (Bishop 4.22). Show this leads to the result \( w \propto m_2^{-1} - m_1^{-1} \).

(c) Show that the Fisher criterion has the two following equivalent forms (Bishop Eqs. 4.25-4.26):

\[ J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{w^T S_B w}{w^T S_W w} \]

Hint: use Bishop 4.22 to expand the numerator, use 4.20 & 4.24 to expand the denominator.

Problem 2. (Matlab) Perceptron algorithm

Bishop section 4.1.7 covers the perceptron algorithm. In this problem, you will implement the key steps of the basic algorithm, investigate the effects of the learning rate \( \eta \), and visualize the variability in the final learned decision boundary.

(a) Implementing the key steps in the perceptron algorithm:

You are provided with a code framework that corresponds to the perceptron algorithm as described in Bishop. Your first task is to write code for the functions \texttt{any_points_wrong.m}, \texttt{current_point_wrong.m}, and \texttt{update_w.m}, which are called by the main script \texttt{runPerceptron.m}. Zip up all of the code (provided for you and written by you) when you have written these functions; you will email this in.

(b) The effects of the learning rate \( \eta \):

Once you have implemented the necessary functions, you will investigate what effect the learning rate \( \eta \) has on the rate of convergence. For each of \( \eta = \{0.05, 0.15, 0.5, 10, 100\} \), first get a feel for the algorithm behavior by running the algorithm with the plotting turned on\(^1\). After this, turn plotting off. For each value of \( \eta \), run the algorithm to convergence 50 times, recording the total number of updates to \( w \) each time. When this is complete, determine the min, max, mean, and standard deviation of the number of updates needed for each step size. Do you see any clear winners or losers?

(c) Variability in the learned decision boundary:

There is no single decision boundary to which the perceptron algorithm will converge. To get a sense for the range of possible results, run the algorithm to convergence 50 times with \( \eta = 5 \). Store the final \( w \) vector each time. Next, plot the boundaries for each of the 50 \( w \) vectors using \texttt{plot_boundary} with the marker type set to ‘*’. Finally, determine the mean \( w \) vector and plot it using \texttt{plot_boundary} with the marker type set to ‘-’. You will hand this picture in.

\(^1\)Do this by setting \texttt{plot_flag=1}
Problem 3. (Matlab) Fisher’s Linear Discriminant

Bishop section 4.1.4 covers the Fisher’s linear discriminant. In this problem, you will implement the key steps of the basic algorithm, and visually compare its performance to a simple competitor. Your resulting graphs should resemble Bishop Figure 4.6.
Implementing the key steps in Fisher’s linear discriminant:
You are provided with a code framework (runFisher.m) that corresponds to the Fisher’s linear discriminant as described in Bishop. Your task is to write code for the steps that have been left incomplete. Each section you are to complete has been marked %your code% in the provided m file. Specifically, you are asked to provide code to:

- Sample the points from each class.
- Find the mean of the samples from each class.
- Determine the vector joining the class sample means.
- Project the points using this vector.
- Calculate the total within-class variance.
- Calculate the vector known as Fisher’s linear discriminant.
- Project the points using this Fisher’s vector.