Solutions

Problem 1: Recall (2pts) (Answer in one sentence only.)
Given an unbiased linear discriminant defined by the augmented weight vector $a$, with modified input vector $y$, write the equation for the discriminant’s decision boundary.

$$a^T y = 0$$

Problem 2: Work (8 pts) (Show all derivations/work and explain.)
You are given a 4-sample data set of points in the 2D Cartesian plane. The samples, given in the form $\begin{pmatrix} x_i & \omega_i \end{pmatrix}$, are

$$\begin{cases} \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{cases}.$$.

Using the unbiased Batch-Perceptron algorithm, compute the discriminant weight vector $a$. Assume a fixed increment $\eta = 1$ and an initial vector of $\begin{pmatrix} 1 & 0 \end{pmatrix}$. Do not normalize the discriminant vector (this will result in messy calculation). Plot $a$.

Begin by normalizing the training data, replacing the labeled sample pairs with unlabeled vectors $y_i = x_i \omega_i$. This gives us 4 modified data samples:

$$\begin{cases} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} \end{cases}.$$.

We then apply our initial weight vector to these samples, and collect all the samples that are not currently satisfied (i.e. those samples $y$ such that $a^T y \leq 0$). Given our initial $a$ of $\begin{pmatrix} 1 & 0 \end{pmatrix}$, this means the first, third and fourth samples are unsatisfied. We then collect these samples in a batch by summing their vectors and multiplying the result by $\eta$. This gives us a $\delta_a$ of $\begin{pmatrix} -1 & 3 \end{pmatrix}$. We then compute the new $a$ via $a = a + \delta_a$, thus obtaining a new weight vector $\begin{pmatrix} 0 & 3 \end{pmatrix}$.

From here, we can continue on to the next iteration. Once again, we begin by collecting all unsatisfied samples, but this time $a^T y > 0$ for all 4 samples. We have thus successfully computed a weight vector describing a linear discriminant that correctly classifies all training points, and can stop here.

Note: some students converted the problem to a homogeneous coordinates system by adding an additional 1 to the beginning of every sample vector, likely because the algorithm was described as unbiased. This conversion is not necessary simply because the classifier is unbiased. It is a technique for converting a classifier that does have a bias parameter into an unbiased form. That was not called for in this problem (and, indeed, a bias parameter is not needed to find a solution that fully separates the two classes in this data set).