## Homework 10: Machine Learning

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 1: Multiclass Perceptron**

0.0/44.0 points

In this problem, we will train a Multi-class perceptron on data of the form (f(X) ∈ R2, Y ∈ {A, B, C}). In particular, we will use training data to update three weight vectors, Wy ∈ R2, y = A, B, C.

We begin with the following set of randomly-initialized weight vectors:

|  |  |  |
| --- | --- | --- |
| y | Wy,1 | Wy,2 |
| A | 1.08 | 0.19 |
| B | -1.75 | -1.59 |
| C | 0.58 | 1.17 |

We will now incorporate the training data point f(X) = (-0.43, -1.19); Y + C. Fill in the resulting weight-feature dot product, and update the weight values as necessary.

**Note:** For all of these questions, if a weight vector doesn't get updated, make sure to still write its value in the table provided.

|  |  |  |  |
| --- | --- | --- | --- |
| y | Wy **• f(x)** | new Wy,1 | new Wy,2 |
| A |  |  |  |
| B |  |  |  |
| C |  |  |  |

We will now incorporate the training data point f(X) = (-0.44, 0.49; Y = C. Fill in the resulting weight-feature dot product, and update the weight values as necessary.

|  |  |  |  |
| --- | --- | --- | --- |
| y | Wy **• f(x)** | new Wy,1 | new Wy,2 |
| A |  |  |  |
| B |  |  |  |
| C |  |  |  |

We took over from here and ran the perceptron algorithm till convergence. In case you're curious, this data set consisted of 50 data points, and the perceptron algorithm converged after 314 steps. Of these steps, 37 changed the weight vector.

At convergence, we have the following weight vectors:

|  |  |  |
| --- | --- | --- |
| y | Wy,1 | Wy,2 |
| A | 2.5 | 0.57 |
| B | 2.54 | -0.35 |
| C | -5.13 | -0.45 |

Use the converged perceptron to classify the new data point f(X) = (-1.38, -0.19). Fill in the weight-feature dot product for each value of y.

|  |  |
| --- | --- |
| y | **Wy • f(x)** |
| A |  |
| B |  |
| C |  |

What is the predicted label?



**Question 2: Perceptrons and Mira**

0.0/18.0 points

Which of the following statements about perceptron and MIRA are true?

Immediately after updating on a missed example where label y was chosen instead of the correct label y\*, perceptron is guaranteed to give higher weight to y\* than to y.

Immediately after updating on a missed example where label y was chosen instead of the correct label y\*, MIRA with C = +∞ is guaranteed to give higher weight to y\* than to y.

Neither of the above.

On a missed example, from the same starting weight vector, the perceptron might make an update with a larger step size than MIRA.

On a missed example, from the same starting weight vector, MIRA might make an update with a larger step size than the perceptron.

Neither of the above.

Immediately after updating on a missed example, perceptron will always have a lower training error rate.

Immediately after updating on a missed example, MIRA will always have a lower training error rate.

Neither of the above.

**Question 3: Datasets**

0.0/6.0 points

When training a classifier, it is common to split the available data into a training set, a hold-out set, and a test set, each of which has a different role.

Which data set is used to learn the conditional probabilities?

Training Data

Hold-Out Data

Test Data

Which data set is used to tune the Laplace Smoothing hyperparameters?

Training Data

Hold-Out Data

Test Data

Which data set is used for quantifying performance results?

Training Data

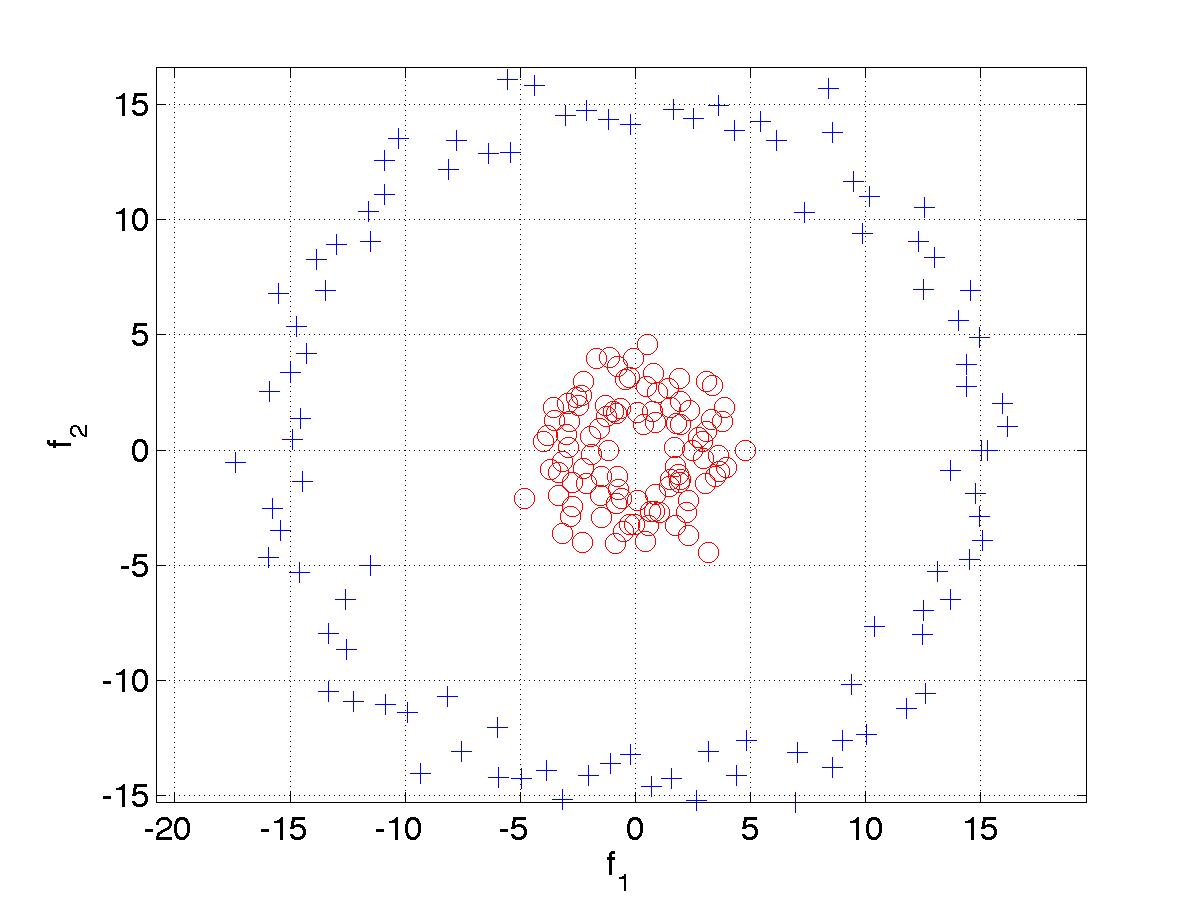
Hold-Out Data

Test Data

**Question 4: Linear Separability**

0.0/16.0 points

Consider the data in the figure below.



The data is plotted as a function of two features, f1 and f2. As plotted, the data is not linearly separable. Which of the following candidate features f3, when added, would cause the data to be linearly separable?

f3 = |f1| + |f2|

 f3 = sin(f1)

 f3 = f12+ f22

 f3 = f12

 f3 = f1

 f3 = 1

 f3 = f1f2

 f3 = 1 if f1 ∈ [-7,7] and f2 ∈ [-7,7], 0 otherwise

**Question 5: Kernels**

0.0/16.0 points

Which of the following statements about kernels are True?

A kernel function is defined as a function that corresponds to a dot product of two feature vectors in some expanded feature space.

Every training set is perfectly separable under some feature space transformation if each datapoint is distinct.

Calculating kernel functions takes work that is linearly proportional to the number of weight vectors.

We have three points, x, z1, and z2. Suppose that z1 is geometrically far from x, while z2 is geometrically close to x. If we use a RBF Kernel, K(xi, xj) = exp(-||xi – xj||2 / (2σ)), then k(z1, x) will be close to 0 and k(z2, x) will be close to 1.

Using a quadratic kernel is equivalent to mapping 2-dimensional feature (x1, x2) to a 1-dimensional space (x1 \* x2).