Q1 Multiclass Perceptron
65 Points

In this problem, we will train a Multi-class perceptron on data of the form $(f(X) \in \mathbb{R}^2, Y \in \{A, B, C\})$. In particular, we will use training data to update three weight vectors, $W_y \in \mathbb{R}^2, y = A, B, C$.

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

y	$W_{y,1}$	$W_{y,2}$
A	-0.82	-0.02
B	-1.63	-0.88
C	0.39	0.65

Q1.1
13 Points

We will now incorporate the training data point $f(X) = (-1.06, 0.95); Y = C$.
First fill in the resulting weight-feature dot products.

$W_A \cdot f(X)$

.8502

$W_B \cdot f(X)$

.8918

$W_C \cdot f(X)$

.2041

Q1.2
13 Points

Now update the weight values as necessary for the training point from part 1.

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

new $W_{A,1}$

-.82

new $W_{A,2}$

-.02

new $W_{B,1}$

-.57

new $W_{B,2}$

-1.83

new $W_{C,1}$

-.67

new $W_{C,2}$

1.6

Q1.3
13 Points

We will now incorporate the training data point $f(X) = (0.09, 1.48); Y = A.$ Fill in the resulting weight-feature dot product, and update the weight values as necessary.

$W_A \cdot f(X)$

-.1034

$W_B \cdot f(X)$

-2.7597

$W_C \cdot f(X)$

2.3077

Q1.4
13 Points

new $W_{A,1}$

-.73

new $W_{A,2}$

1.46

new $W_{B,1}$

-.57

new $W_{B,2}$

-1.83

new $W_{C,1}$

-.76

new $W_{C,2}$

.12

Q1.5
13 Points

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 722 steps. Of these steps, 103 changed the weight vector.

At convergence, we have the following weight vectors:

y	$W_{y,1}$	$W_{y,2}$
A	3.12	0.96
B	3.11	-0.97
C	-8.29	-0.24

Use the converged perceptron to classify the new data point $f(X) = (-1.35, 0.42)$. Fill in the weight-feature dot product for each value of y.

$W_A \cdot f(X)$

-3.8088

$W_B \cdot f(X)$

-4.6059

$W_C \cdot f(X)$

11.0907

What is the predicted label?

A

B

C

Q2 A Variant on the Perceptron Algorithm

35 Points

You were recently promoted to the Vice President of Recruiting Science at Pacapalooza Technologies. Pacapalooza is expanding rapidly, and you decide to use machine learning to hire the best and brightest. To do so, you have the following available to you for each candidate i in the pool of candidates I: (i) their GPA, (ii) whether they took CS 164 with Hilfinger and received an A, (iii) whether they took CS 188 and received an A, (iv) whether they have a job offer from Pactronic LLC, (v) whether they have a job offer from Pacmania Corp., and (vi) the number of misspelled words on their resume. You decide to represent each candidate $i \in I$ by a corresponding 6-dimensional feature vector $f(x^{(i)})$. You believe that if you just knew the right weight vector $w \in \mathbb{R}^6$ you could reliably predict the quality of a candidate i by computing $w^T f(x^{(i)})$. To determine w, you sample pairs of candidates from the pool. For a pair of candidates (k, l) you can have them face off in a "Pacapalooza-fight." The result is $\text{score}(k > l)$, which tells you that a candidate k is at least $\text{score}(k > l)$ better than candidate l. Note that the score will be negative when l is a better candidate than k. Assume you collected scores for a set of pairs of candidates P, that $\text{score}(k > l) = -\text{score}(k < l)$, and that $\text{score}(k > l) \neq 0$ for any pair $(k, l) \in P$.

Q2.1

22 Points

You decide to employ a perceptron-like algorithm to determine w, where your dataset is P.

Suppose that we encounter a pair $(k, l) \in P$ for which $\text{score}(k > l) > 0$. How do we update w?

- If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w + f(x^{(k)}) - f(x^{(l)})$.
- Update $w \leftarrow w + f(x^{(k)}) - f(x^{(l)})$.
- If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w - f(x^{(k)}) + f(x^{(l)})$.
- If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w + w^T(f(x^{(k)}) - f(x^{(l)}))$.
- Update $w \leftarrow w - f(x^{(k)}) + f(x^{(l)})$.

Suppose that we encounter a pair $(k, l) \in P$ for which $\text{score}(k > l) < 0$. How do we update w?

- If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w - f(x^{(l)}) - f(x^{(k)})$.
- Update $w \leftarrow w - f(x^{(k)}) + f(x^{(l)})$.
- If $w^T f(x^{(l)}) \geq w^T f(x^{(k)}) - \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w + f(x^{(l)}) - f(x^{(k)})$.
- Update $w \leftarrow w + f(x^{(k)}) - f(x^{(l)})$.
- If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w - w^T(f(x^{(k)}) - f(x^{(l)}))$.

Q2.2
13 Points

Your perceptron-like algorithm is unable to reach zero errors on your training data. Which of the following techniques would help improve performance on the training data?

- [ ] Running the perceptron algorithm for a longer period of time; since the perceptron algorithm is guaranteed to keep equal or reduce the number of errors at each time step, we are guaranteed to eventually reach zero errors.
- [x] Add higher-order features to our list of six features, e.g., pairwise products, and run our perceptron algorithm with this newly constructed dataset. New features increase the dimensionality of the space, and improve the chances that the data is separable.
- [ ] Removing some of the features from the training data, and training the perceptron on this subset of data. Too many features increases the chance of overfitting on the training data, which would decrease performance on the training data.
- [ ] Collect a larger set of data, so the perceptron algorithm does a better job of fitting to the data distribution; with a small training set, the perceptron cannot fully learn w, causing it to produce errors on the training data.

HW 10 (Electronic Component)

Graded

Student

ریحانه شاهرخیان

Total Points

100 / 100 pts

Question 1

Multiclass Perceptron		**65** / 65 pts
1.1	(no title)	**13** / 13 pts
1.2	(no title)	**13** / 13 pts
1.3	(no title)	**13** / 13 pts
1.4	(no title)	**13** / 13 pts
1.5	(no title)	**13** / 13 pts

Question 2

A Variant on the Perceptron Algorithm		**35** / 35 pts
2.1	(no title)	**22** / 22 pts
2.2	(no title)	**13** / 13 pts