Classification and the Perceptron Algorithm

1. Perceptron Mistakes

Consider applying the perceptron algorithm (through the origin) based on a small training set containing three points: \( x^{(1)} = [-1, 1] \), \( x^{(2)} = [0, -1] \), and \( x^{(3)} = [1.5, 1] \) with labels \( y^{(1)} = 1 \), \( y^{(2)} = -1 \), and \( y^{(3)} = 1 \). Given that the algorithm starts with \( \theta^{(0)} = 0 \), the first point that the algorithm sees is always considered a mistake. The algorithm starts with some data point and then cycles through the data until it makes no further mistakes.

(a) How many mistakes does the algorithm make until convergence if the algorithm starts with data point \( x^{(1)} \)? How many mistakes does the algorithm make if it starts with data point \( x^{(2)} \)? Draw a diagram showing the progression of the plane as the algorithm cycles.

(b) Now assume that \( x^{(3)} = [10, 1] \). How many mistakes does the algorithm make until convergence if cycling starts with data point \( x^{(1)} \)? How many if it starts with data point \( x^{(2)} \)? Draw a diagram showing the progression of the plane as the algorithm cycles.

(c) Explain the difference between the number of mistakes made by the algorithm in part a) and part b).

(d) Broadly describe an adversarial procedure for selecting the order and value of labeled data points so as to maximize the number of mistakes the perceptron algorithm makes before converging. Assume that the data is indeed linearly separable, by a hyperplane that you (but not the algorithm) know.

2. Perceptron Performance

Given a set of linearly separable training examples, we train a classifier using the perceptron algorithm twice, initializing \( \theta \) differently for each run. The two training procedures traverse the data points in the same order and are run until convergence.

(a) Do they converge to the same \( \theta \)? If so, provide a proof. If not, provide a counterexample.

(b) Both classifiers converge, so their performance on the training set is the same. What about their performance on the test set?

(c) The following table shows a data set and the number of times each point is misclassified during a run of the perceptron algorithm (with offset). \( \theta \) is initialized to zero.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( x^{(i)} )</th>
<th>( y )</th>
<th>times misclassified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[-3, 2]</td>
<td>+1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>[-1, 1]</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>[-1, -1]</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>[2, 2]</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>[1, -1]</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>
Write down the post-training $\theta$.

3. Decision Boundaries

(a) Consider the OR function defined over three binary variables: $f(x_1, x_2, x_3) = (x_1 \lor x_2 \lor x_3)$. We aim to find a $\theta$ such that, for any $x = [x_1, x_2, x_3]$, where $x_i \in \{0, 1\}$:

$$\theta \cdot x + \theta_0 > 0 \text{ when } f(x_1, x_2, x_3) = 1,$$
$$\theta \cdot x + \theta_0 < 0 \text{ when } f(x_1, x_2, x_3) = 0.$$

i. If $\theta_0 = 0$ (no offset), would it be possible to learn such a $\theta$.

ii. Would it be possible to learn the pair $\theta$ and $\theta_0$?

(b) You are given the following labeled data points:

- Positive examples: $[-1, 1]$ and $[1, -1]$,
- Negative examples: $[1, 1]$ and $[2, 2]$.

For each of the following parameterized families of classifiers, find the parameters of a family member that can correctly classify the above data, or explain why no such family member exists.

i. Inside or outside of an origin-centered circle with radius $r$,
ii. Inside or outside of an $[x, y]$-centered circle with radius $r$,
iii. Above or below a line through the origin with normal $\theta$,
iv. Above or below a line with normal $\theta$ and offset $\theta_0$.

(c) Which of the above are families of linear classifiers.

4. Feature Vectors

Consider a sequence of $n$-dimensional data points, $x^{(1)}, x^{(2)}, \ldots$, and a sequence of $m$-dimensional feature vectors, $z^{(1)}, z^{(2)}, \ldots$, extracted from the $x$’s by a linear transformation, $z^{(i)} = Ax^{(i)}$. If $m$ is much smaller than $n$, you might expect that it would be easier to learn in the lower dimensional feature space than in the original data space.

(a) Suppose $n = 6$, $m = 2$, $z_1$ is the average of the elements of $x$, and $z_2$ is the average of the first three elements of $x$ minus the average of fourth through sixth elements of $x$. Determine $A$.

(b) Suppose $h(z) = sign(\theta_z \cdot z)$ is a classifier for the feature vectors, and $h(x) = sign(\theta_x \cdot x)$ is a classifier for the original data vectors. Given a $\theta_z$ that produces good classifications of the feature vectors, is there a $\theta_x$ that will identically classify the associated $x$’s?

(c) Given the same classifiers as in (b), if there is a $\theta_z$ that produces good classifications of the data vectors, will there always be a $\theta_z$ that will identically classify the associated $z$’s? Under what conditions on $A$ will such a $\theta_z$ always exist.

(d) If $m < n$, can the perceptron algorithm converge more quickly when training in $z$-space? If so, provide an example.

(e) If $m < n$, can we find a more accurate classifier by training in $z$-space, as measured on the training data? How about on unseen data?