Written assignment

1. Text book exercise 1.11

2. (Probability Decision Boundary). Consider a case where we have learned a conditional probability distribution $P(y|x)$. Suppose there are only two classes, and let $p_0 = P(y = 0|x)$ and $p_1 = P(y = 1|x)$. Consider the following loss matrix:

<table>
<thead>
<tr>
<th>predicted label $\hat{y}$</th>
<th>true label $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Show that the decision $\hat{y}$ that minimizes the expected loss is equivalent to setting a specific probability threshold $\theta$ (please compute this threshold value) and predicting $\hat{y} = 0$ if $p_1 < \theta$ and $\hat{y} = 1$ if $p_1 \geq \theta$. Show a loss matrix where the threshold is 0.1.

3. (Reject Option). In many applications, the classifier is allowed to “reject” a test example rather than classifying it into one of the classes. Consider, for example, a case in which the cost of a misclassification is $10 but the cost of having a human manually make the decision is only $3. We can formulate this as the following loss matrix:

<table>
<thead>
<tr>
<th>decision</th>
<th>true label $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>predict 0</td>
<td>0</td>
</tr>
<tr>
<td>predict 1</td>
<td>10</td>
</tr>
<tr>
<td>reject</td>
<td>3</td>
</tr>
</tbody>
</table>

Suppose $P(y = 1|x)$ is predicted to be 0.2. Which decision minimizes the expected loss? Now suppose $P(y = 1|x) = 0.4$. Now which decision minimizes the expected loss? Show that in cases such as this there will be two specific thresholds $\theta_0$ and $\theta_1$ (compute their values for this loss matrix) such that the optimal decision is to predict 0 if $p_1 < \theta_0$, reject if $\theta_0 \leq p_1 \leq \theta_1$, and predict 1 if $p_1 > \theta_1$.

<table>
<thead>
<tr>
<th>decision</th>
<th>true label $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>predict 0</td>
<td>0</td>
</tr>
<tr>
<td>predict 1</td>
<td>5</td>
</tr>
<tr>
<td>reject</td>
<td>3</td>
</tr>
</tbody>
</table>
4. (Weighted hinge loss). In our derivation of the Perceptron algorithm, we used the hinge loss to approximate the 0/1 loss. Suppose that we have a general loss matrix with the cost of a false positive being \( L(1, -1) = c_0 \) and the cost of a false negative \( L(-1, 1) = c_1 \). Suppose we used

\[
\tilde{J}(w) = \frac{1}{N} \sum_{i=1}^{N} z_i \max(0, -y_i w \cdot x_i)
\]

for our approximate objective function, where \( z_i = c_0 \) if \( y = -1 \) and \( z_i = c_1 \) if \( y = 1 \). Compute the gradient using this approximation, and show how the batch Perceptron algorithm is modified to incorporate this change.

5. In our definition of logistic regression, we defined

\[
p_1(x; w) = \frac{1}{1 + \exp[-w \cdot x]},
\]

\[
p_0(x; w) = 1 - p_1(x; w).
\]

Show that this is equivalent to

\[
\log \frac{p_1(x; w)}{p_0(x; w)} = w \cdot x.
\]

Show also that

\[
p_1(x; w) = \frac{\exp w \cdot x}{1 + \exp w \cdot x}.
\]

This is in exponential form.
Implementation assignment

In this assignment, you will code the batch perceptron and voted perceptron algorithms. Matlab is preferred but your implementation can be in any language of your choice. You need to submit your source code electronically together with a write-up including the contents specified below.

You will be provided with two data sets (will be posted soon). You will test your batch perceptron algorithm on the first data set (twogaussian), which contains two linearly separable gaussian classes. Your write up needs to include:

1. A plot of the classification error on the training set as a function of the number of training epoches. Note that one epoch of training goes through the full training set exactly once.

2. A scatter plot of the training data using different colors for different classes. On this scatter plot, please plot the final linear decision boundary learned by your perceptron algorithm (please also provide the weights output by your perceptron algorithm).

You will test your voted perceptron implementation on the second data set iris-twoclass. This is based on a commonly used bench-mark data set iris. In particular, we extracted two classes, and two input features from the original problem. You need to include the following in your write-up:

1. A plot of the classification error on the training set as a function of the number of training epoches (up to 100 epoches). Note that one epoch of training goes through the full training set exactly once.

2. Please visualize the final decision boundary produced by your voted perceptron algorithm (trained for 100 epoches). Note that to produce this decision boundary, one strategy is to sample a large number of $x$ points on a fine grid in the input space, and compute their predicted output. You can then visualize the decision boundary by plotting these points using different colors based on their predicted classes.

3. Note that for voted perceptron, there is a simple modification people often use in order to avoid storing all intermediate weight vectors. That is to compute

$$w_{avg} = \sum_{n=0}^{N} c_n w_n$$

and use the following decision rule:

$$h(x) = sgn\{w_{avg} \cdot x\}$$

Please compute $w_{avg}$ based on the weights that are learned by your voted perceptron algorithm (100 epoches), and plot the linear decision boundary it produces on a scatter plot of the training data. Are these two decision boundaries equivalent? Why?

Please provide good comments to your code — be sure to 1) provide the pseudo-code of your algorithm, and to 2) clearly indicate which part of your code corresponds to which step of your pseudo-code. This will help the instructor to pinpoint issues that you may having in your understanding/coding if the output of your code is not correct.