Written assignment

1. Let $x$ and $y$ be two independent random variables, show that:
   i. For any constants $a$ and $b$, $E(ax + by) = aE(x) + bE(y)$
      \[
      E(ax + by) = \int \int (ax + by)f(x, y) \, dx \, dy \\
                  = \int \int (ax + by)f(x)f(y) \, dx \, dy \\
                  = \int \int axf(x)f(y) \, dx \, dy + \int \int byf(x)f(y) \, dx \, dy \\
                  = a\int xf(x) \, dx + b\int yf(y) \, dy \\
                  = aE(x) + bE(y)
      \]
   ii. For any constants $a$ and $b$, $Var(ax + by) = a^2Var(x) + b^2Var(y)$
      \[
      Var(aX + bY) = \int \int (ax + by - E(ax + by))^2 f(x, y) \, dx \, dy \\
                   = \int \int a^2(x - E(x))^2 + b^2(y - E(y))^2 + 2ab(x - E(x))(y - E(y)) f(x)f(y) \, dx \, dy \\
                   = a^2Var(x) + b^2Var(y) + 2\int a(x - E(x))f(x)b\int (y - E(y))f(y) \, dx \, dy \\
                   = a^2Var(x) + b^2Var(y)
      \]
   iii. $Cov(x, y) = 0$
      \[
      Cov(x, y) = \int \int (x - E(x))(y - E(y))f(x, y) \, dx \, dy \\
                   = \int (x - E(x))f(x)\int (y - E(y))f(y) \, dy \, dx \\
                   = \int (x - E(x))f(x)\int yf(y) \, dy - \int E(y)f(y) \, dy \, dx \\
                   = \int (x - E(x))f(x)E(y) - E(y) \, dx \\
                   = 0
      \]

2. Let $X$ and $Y$ be two independent uniformly distributed random variables. $X, Y \sim U(0, 1)$. Let $Z = \max(X, Y)$. What is the PDF function of $Z$?

   First we consider the CDF function of $Z$:
   \[
   P(Z \leq z) = p(X \leq z \text{ and } Y \leq z) = p(X \leq z)p(Y \leq z) = z^2
   \]
   Taking the first order derivative of $P$, we have the pdf function:
   \[
   f(z) = \frac{dP}{dz} = 2z
   \]
3. We have three boxes colored Red, Blue and Green respectively. There are 3 apples and 6
oranges in the red box, 3 apples and 3 oranges in the green box, and 5 apples and 3 oranges
in the blue box. Now we randomly select one of the boxes (with equal probability) and grab
a fruit from the box. What is the probability that it is an apple? If the fruit that we got is
an orange, what is the probability that we selected the green box?

\[ p(a) = p(a|B)p(B) + p(a|G)p(G) + p(a|R)p(R) = \frac{5}{8} \cdot \frac{1}{3} + \frac{1}{2} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{3} = \frac{35}{72} \]

\[ p(G|o) = \frac{p(o|G)p(G)}{p(o)} = \frac{\frac{1}{2} \cdot \frac{1}{3}}{1 - \frac{35}{72}} = \frac{12}{37} \]

4. (Probability Decision Boundary). Consider a case where we have learned a conditional prob-
ability 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>

It can be shown that the decision $\hat{y}$ that minimizes the expected loss is equivalent to setting
a specific probability threshold $\theta$ and predicting $\hat{y} = 0$ if $p_1 < \theta$ and $\hat{y} = 1$ if $p_1 \geq \theta$. Please compute the $\theta$ for the above given loss matrix. Show a loss matrix where the threshold is 0.1.

We want to predict $\hat{y} = 1$ if the expected loss of guessing 1 is less than the expected loss of
guessing zero. Following the derivation in class, we should predict 1 iff

\[ P(y = 0|x) 5 < P(y = 1|x) 10 \] (1)

Define $p_1 = P(y = 1|x)$, then this becomes

\[ (1 - p_1) 5 < p_1 10 \] (2)

\[ 5 - 5p_1 < 10p_1 \] (3)

\[ -15p_1 < -5 \] (4)

\[ p_1 > \frac{5}{15} \] (5)

\[ p_1 > \frac{1}{3} \] (6)

So we should set the threshold to 1/3.

To get a threshold of 0.1, we could use the 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>1</td>
</tr>
</tbody>
</table>
5. (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>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?

Show that in cases such as this there will be two specific thresholds \( \theta_0 \) and \( \theta_1 \) 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 \( p_1 > \theta_1 \).

For the above loss matrix, the expected losses of the three actions are:

- predict 0: \( 0.2 \times 10 = 2 \)
- predict 1: \( 0.8 \times 10 = 8 \)
- reject: 3.

So the best action is to predict 0.

Consider the general loss matrix where \( L(0, 1) = c_1 \), \( L(1, 0) = c_0 \), and the reject action costs a fixed amount \( c_r \). Plugging into the decision-making formula, we obtain the following:

We should predict 0 if \( p_1c_1 < \min((1 - p_1)c_0, c_r) \)
We should predict 1 if \( (1 - p_1)c_0 < \min(p_1c_1, c_r) \)
We should reject if \( c_r < \min(p_1c_1, (1 - p_1)c_0) \)

Note that the constraints on \( c_r \) imply that \( c_r < \min(c_0, c_1) \) in order for the reject option to make sense. Otherwise, there will be no setting of \( p_1 \) that satisfies this constraint.

By doing some algebra, we can transform these constraints into the following:

We should predict 0 if \( p_1 < \min(\frac{c_0}{c_0 + c_1}, \frac{c_r}{c_1}) \)
We should predict 1 if \( p_1 > \max(\frac{c_0}{c_0 + c_1}, 1 - \frac{c_r}{c_0}) \)
else reject.

This gives us the two thresholds \( \theta_0 \) and \( \theta_1 \).

6. (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.

The partial derivative for training instance \( i \) wrt weight \( w_j \) is

\[
\frac{\partial \tilde{J}(w)}{\partial w_j} = \left\{ \begin{array}{ll}
0 & \text{if } y_i w \cdot x_i > 0 \\
-z_i y_i x_{ij} & \text{otherwise}
\end{array} \right.
\]
So the batch gradient descent algorithm is the following:

**Given:** training examples \((x_i, y_i), i = 1 \ldots N\)

**Let** \(w = (0, 0, 0, \ldots, 0)\) be the initial weight vector.

**Repeat** until convergence

Let \(\delta = (0, 0, \ldots, 0)\) be the gradient vector.

For \(i = 1\) to \(N\) do

if \((y_i = -1 \text{ and } w \cdot x_i > 0)\) \(\text{error}_i = c_0\)

else if \((y_i = +1 \text{ and } w \cdot x_i < 0)\) \(\text{error}_i = c_1\)

else \(\text{error}_i = 0\)

\(\delta = \delta - \text{error}_i \cdot y_i \cdot x_i\)

\(\delta := \delta / N\)

\(w := w - \eta \delta\)

We can interpret this as using different learning rates for the two kinds of errors.

7. In class when discussing linear regression, we assume that the Gaussian noise is independently identically distributed. Now we assume the noises \(\epsilon_1, \epsilon_2, \cdots, \epsilon_N\) are independent but each \(\epsilon_i \sim N(0, \sigma_i^2)\). Please 1) write down the log likelihood function of \(w\); 2) show that minimizing the log likelihood is equivalent to minimizing a weighted least square loss function \(J(W) = \sum_1 a_i (y_i - x_i^T w)^2\), and express each \(a_i\) in terms of \(\epsilon_i\); and 3) solve for \(w_{\text{MLE}}\):

\[
\log p(D|M) = \sum_{i=1}^{N} \log N(y_i | x_i^T w, \sigma_i^2)
\]

\[
= \sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma_i}} e^{-\frac{1}{2\sigma_i^2} (y_i - x_i^T w)^2}
\]

\[
= \sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma_i}} - \sum_{i=1}^{N} \frac{1}{2\sigma_i^2} (y_i - x_i^T w)^2
\]

Maximizing the log likelihood is equivalent to minimizing the second term in the above equation, which can be represented as:

\[
J(w) = \sum_i a_i (y_i - x_i^T w)^2
\]

where \(a_i = \frac{1}{2\sigma_i^2}\).

Let \(y = y_1, y_2, \cdots, y_N^T\), \(X\) be the data matrix whose rows correspond to training examples, and \(A\) be a diagonal matrix with \(A(i, i) = a_i\). Rewriting the objective in matrix form:

\[
J(w) = (y - Xw)^T A (y - Xw)
\]

\[
= y^T A y - 2 w^T X^T A y + w^T X^T A X w
\]

Take the gradient of \(J\) and set it to zero, we obtain \(w^*\) as follows:

\[
\nabla_w J = 2 X^T A X w - 2 X^T A y = 0 \Rightarrow
\]

\[
2 X^T A X w = 2 X^T A y \Rightarrow
\]

\[
w^* = (X^T A X)^{-1} X^T A y
\]