ECE523: Engineering Applications of Machine Learning and Data Analytics

Name: ______________________________

Signature: ______________________________

Date: ______________________________

Instructions: There are seven problems. Partial credit is given for answers that are partially correct. No credit is given for answers that are wrong or illegible. All work must be supported and code must be submitted for credit.

Theory: ___________

Practice: ___________

Total: ___________

Part A: Theory (20pts)

(3pts) Maximum Posterior vs Probability of Chance

Show/explain that $\mathbb{P}(\omega_{\max}|\mathbf{x}) \geq \frac{1}{c}$ when we are using the Bayes decision rule. Derive an expression for $\mathbb{P}(\text{error})$. Let $\omega_{\max}$ be the state of nature for which $P(\omega_{\max}|\mathbf{x}) \geq P(\omega_i|\mathbf{x})$ for $i =, 1\ldots, c$. Show that $\mathbb{P}(\text{error}) \leq (c-1)/c$ when we use the Bayes rule to make a decision. Hint, use the results from the previous questions.

We know from the axioms of probability that $\sum_{i=1}^{c} \mathbb{P}(\omega_i|\mathbf{x}) = 1$. Clearly, $\mathbb{P}(\omega_i|\mathbf{x}) = \frac{1}{c}$ will hold for the normalization axiom and the stated inequality as well. For the remainder of the inequality I use a proof by contradiction. Assume not that $\mathbb{P}(\omega_{\max}|\mathbf{x}) < \frac{1}{c}$, then $\exists j$ such that $\mathbb{P}(\omega_j|\mathbf{x}) > \frac{1}{c}$, which is a result of the normalization axiom. This is a contradiction to the definition of $\omega_{\max}$, thus proving

$$\mathbb{P}(\omega_{\max}|\mathbf{x}) \geq \frac{1}{c}$$

Now to find $\mathbb{P}(\text{error})$ we use the fact from our notes that $\mathbb{P}(\text{error}|\mathbf{x}) = 1 - \mathbb{P}(\omega_{\max}|\mathbf{x})$, which gives us:

$$\begin{aligned}
\mathbb{P}(\text{error}) &= \int_{\mathbf{x}} \mathbb{P}(\text{error}, \mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \int_{\mathbf{x}} \mathbb{P}(\text{error}|\mathbf{x}) \mathbb{P}(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \int_{\mathbf{x}} (1 - \mathbb{P}(\omega_{\max}|\mathbf{x})) \mathbb{P}(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \int_{\mathbf{x}} \mathbb{P}(\mathbf{x}) \mathrm{d}\mathbf{x} - \int_{\mathbf{x}} \mathbb{P}(\omega_{\max}|\mathbf{x}) \mathbb{P}(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= 1 - \int_{\mathbf{x}} \mathbb{P}(\omega_{\max}|\mathbf{x}) \mathbb{P}(\mathbf{x}) \mathrm{d}\mathbf{x}
\end{aligned}$$

Thus proving the form of $\mathbb{P}(\text{error})$. The error can be bounded using the previous results.

$$\begin{aligned}
\mathbb{P}(\text{error}) &= 1 - \int_{\mathbf{x}} \mathbb{P}(\omega_{\max}|\mathbf{x}) \mathbb{P}(\mathbf{x}) d\mathbf{x} && (1) \\
&\leq 1 - \frac{1}{c} \int_{\mathbf{x}} \mathbb{P}(\mathbf{x}) d\mathbf{x} && (2) \\
&= 1 - \frac{1}{c} = \frac{c-1}{c} && (3)
\end{aligned}$$

(3pts) Bayes Decision Rule Classifier

Let the elements of a vector $\mathbf{x} = [x_1, \ldots, x_d]^\mathsf{T}$ be binary valued. Let $\mathbb{P}(\omega_j)$ be the prior probability of the class ω_j ($j \in [c]$), and let

$$p_{ij} = \mathbb{P}(x_i = 1|\omega_j)$$

with all elements in $\mathbf{x}$ being independent. If $\mathbb{P}(\omega_1) = \mathbb{P}(\omega_2) = \frac{1}{2}$, and $p_{i1} = p > \frac{1}{2}$ and $p_{i2} = 1 - p$, show that the minimum error decision rule is

$$\text{Choose } \omega_1 \text{ if } \sum_{i=1}^{d} x_i > \frac{d}{2}$$

Solution: By independence of x_i, we know that $\mathbb{P}(\mathbf{x}|\omega_j) = \prod_{i=1}^{d} \mathbb{P}(x_i|\omega_j)$, and let $k = \sum_{i=1}^{d} x_i$. Then there are k values where $\mathbf{x}$ is non-zero and $d-k$ where $\mathbf{x}$ is zero. Since the prior probabilities for each class are the same, we select the class based off of the calculation of the likelihood probabilities. Furthermore, we can view $\mathbf{x}$ as a collection of success and failures, which from ECE503 should remind you of a Bernoulli random variable. Using Bayes rule, we select ω_1 if

$$\begin{aligned}
\mathbb{P}(\mathbf{x}|\omega_1) &> \mathbb{P}(\mathbf{x}|\omega_2) \\
\prod_{i=1}^{d} \mathbb{P}(x_i|\omega_1) &> \prod_{i=1}^{d} \mathbb{P}(x_i|\omega_2) \\
p^k (1-p)^{d-k} &> (1-p)^k p^{d-k} \\
\left(\frac{1-p}{p}\right)^{d-k} &> \left(\frac{1-p}{p}\right)^{k} \\
\left(\frac{1-p}{p}\right)^{d} \left(\frac{1-p}{p}\right)^{-k} &> \left(\frac{1-p}{p}\right)^{k} \\
\left(\frac{1-p}{p}\right)^{d} &> \left(\frac{1-p}{p}\right)^{2k} \\
\left(\frac{1-p}{p}\right)^{d} &> \left(\frac{1-p}{p}\right)^{2\sum_{i=1}^{d} x_i} \\
d &< 2\sum_{i=1}^{d} x_i \\
\frac{d}{2} &< \sum_{i=1}^{d} x_i
\end{aligned}$$

Therefore, if $\sum_{i=1}^{d} x_i > \frac{d}{2}$ we choose ω_1. Note that the second to last step is a result of $\frac{1-p}{p} < 1$ and $\frac{1}{d}\mathbb{E}[k]$.

(3pts) The Ditzler Household Growing Up

My parents have two kids now grown into adults. Obviously there is me, Greg. I was born on Wednesday 13 November 1985. What is the probability that I have a brother? You can assume that $\mathbb{P}(\text{boy}) = \mathbb{P}(\text{girl}) = \frac{1}{2}$.

Solution: At first glance, it appears that the "born on a Wednesday" fact is unimportant, but this is not so. In fact, it makes all the difference! First, let's see why by listing all allowable outcomes. Each outcome is a quadruple (gender of child 1, day of birth of child 1, gender of child 2, day of birth of child 2). This problem is similar to the Monte Hall problem in the sense that Wednesday may not seem to be relevant to the problem. More formally this is known as the "Two Child Problem".

Our best guess to determine if I have a brother is to consider making a decision that will lead to the minimum probability of error. This should have us immediately thinking of making our decision using Bayes rule. First off, we'll define some notation for the interesting events, which are better shown in the table below:

Event	Symbol	Conditional Probability
At least one child is a boy born on a Wednesday	B	–
Both children are boys	BB	$\mathbb{P}(B\|BB) = \frac{13}{49}$
Elder child is a boy, younger is a girl	BG	$\mathbb{P}(B\|BG) = \frac{1}{7}$
Elder child is a girl, younger is a boy	GB	$\mathbb{P}(B\|GB) = \frac{1}{7}$
Both children are girls	GG	$\mathbb{P}(B\|GG) = 0$

This question is asking you to find the probability that I have a brother given that I was born on a Wednesday, or more formally $\mathbb{P}(BB|B)$. The table below breaks how the conditional probability calculations, where a ✓indicates an outcome that is possible given the description of the problem.

		Boy								**Girl**						
		M	T	W	Th	F	S	Su		M	T	W	Th	F	S	Su
Boy	M	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	T	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	W	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	✓	✓
	Th	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	F	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	S	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	Su	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
Girl	M	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	T	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	W	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	Th	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	F	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	S	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
	Su	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–

Our goal then is to compute:

$$\begin{aligned}\mathbb{P}(BB|B) = \frac{\mathbb{P}(B|BB)\,\mathbb{P}(BB)}{\mathbb{P}(B)} &= \frac{\mathbb{P}(B|BB)\,\mathbb{P}(BB)}{\mathbb{P}(B|BB)\,\mathbb{P}(BB)+\mathbb{P}(B|GB)\,\mathbb{P}(GB)+\mathbb{P}(B|BG)\,\mathbb{P}(BG)+\mathbb{P}(B|GG)\,\mathbb{P}(GG)} \\ &= \frac{\mathbb{P}(B|BB)\,\mathbb{P}(BB)}{\frac{1}{4}\times(\mathbb{P}(B|BB)+\mathbb{P}(B|GB)+\mathbb{P}(B|BG)+\mathbb{P}(B|GG))}\end{aligned}$$

Now, to calculate $\mathbb{P}(B)$ we need to first calculate the conditional probability that at least one child is a son born on a Wednesday, given that the children are BB, BG, GB, or GG. This is just the evidence term, which can be computed using the total probability law. Thus,

$$\mathbb{P}(BB|B) = \frac{\mathbb{P}(B|BB)\,\mathbb{P}(BB)}{\mathbb{P}(B)} = \frac{\frac{13}{49}\times\frac{1}{4}}{\left(\frac{13}{49}+\frac{7}{49}+\frac{7}{49}+\frac{0}{49}\right)\times\frac{1}{4}} = \frac{\frac{13}{49}\times\frac{1}{4}}{\frac{27}{49}\times\frac{1}{4}} = \frac{13}{27}.$$

(10pts) Linear Classifier with a Margin

Show that, regardless of the dimensionality of the feature vectors, a data set that has just two data points, one from each class, is sufficient to determine the location of the maximum-margin hyperplane. Hint #1: Consider a data set of two data points, $\mathbf{x}_1 \in \mathscr{C}_1$ $(y_1 = +1)$ and $\mathbf{x}_2 \in \mathscr{C}_2$ $(y_2 = -1)$ and set up the minimization problem (for computing the hyperplane) with appropriate constraints on $\mathbf{w}^\mathsf{T}\mathbf{x}_1 + b$ and $\mathbf{w}^\mathsf{T}\mathbf{x}_2 + b$ and solve it. Hint #2: This can be formed as a constrained optimization problem.

$$\arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{w}\|_2^2$$

Subject to: (some constraint)

What is $\mathbf{w}$? b? Hint: What are the constraints? How did we solve the constrained optimization problem in Fisher's linear discriminant?

Solution: Irrespective of the dimensionality of the data space, a data set consisting of just two data points – one from each class – is sufficient to determine the location of the maximum-margin hyperplane.

Consider two points, $\mathbf{x}_1 \in \mathscr{C}_1$ $(y_1 = +1)$ and $\mathbf{x}_2 \in \mathscr{C}_2$ $(y_2 = -1)$. The maximum margin hyperplane is therefore determined by solving

$$\arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{w}\|_2^2$$

subject to the constraints

$$\begin{aligned} y_1\mathbf{w}^\mathsf{T}\mathbf{x}_1 + b = 1 \text{ or } \mathbf{w}^\mathsf{T}\mathbf{x}_1 + b = 1 \\ y_2(\mathbf{w}^\mathsf{T}\mathbf{x}_2 + b) = 1 \text{ or } -(\mathbf{w}^\mathsf{T}\mathbf{x}_2 + b) = 1 \end{aligned}$$

To solve this constrained optimization problem, we introduce Lagrange multipliers λ_1 and λ_2.

$$\arg\min_{\mathbf{w},b} \left\{\|\mathbf{w}\|_2^2 + \lambda_1\left(\mathbf{w}^\mathsf{T}\mathbf{x}_1 + b - 1\right) - \lambda_2\left(\mathbf{w}^\mathsf{T}\mathbf{x}_2 + b + 1\right)\right\}$$

To find $\mathbf{w}$ we can take the derivative with respect to $\mathbf{w}$ and set it equal to zero

$$\begin{aligned} &\frac{\partial}{\partial\mathbf{w}}\left\{\|\mathbf{w}\|_2^2 + \lambda_1\left(\mathbf{w}^\mathsf{T}\mathbf{x}_1 + b - 1\right) - \lambda_2\left(\mathbf{w}^\mathsf{T}\mathbf{x}_2 + b + 1\right)\right\} = 0 \\ &0 = \mathbf{w} + \lambda_1\mathbf{x}_1 - \lambda_2\mathbf{x}_2 \end{aligned}$$

Therefore, $\mathbf{w} = -(\lambda_1\mathbf{x}_1 - \lambda_2\mathbf{x}_2)$. If we take the derivative with respect to b, and set it equal to zero we see further that

$$\begin{aligned} &\frac{\partial}{\partial b}\left\{\|\mathbf{w}\|_2^2 + \lambda_1\left(\mathbf{w}^\mathsf{T}\mathbf{x}_1 + b - 1\right) - \lambda_2\left(\mathbf{w}^\mathsf{T}\mathbf{x}_2 + b + 1\right)\right\} = 0 \\ &0 = \lambda_1 - \lambda_2 \qquad (4) \end{aligned}$$

Hence, $\lambda = \lambda_1 = \lambda_2$, so $\mathbf{w} = \lambda(\mathbf{x}_1 - \mathbf{x}_2)$. Note that we are going to let the negative sign be absorbed into the constant λ since the sign is rather arbitrary here. We may now solve for b, starting with our constraints,

$$\begin{aligned}
0 = 1 - 1 = (\mathbf{w}^{\mathsf{T}}\mathbf{x}_1 + b) + (\mathbf{w}^{\mathsf{T}}\mathbf{x}_2 + b) \\
\rightarrow b = -\frac{1}{2}\mathbf{w}^{\mathsf{T}}(\mathbf{x}_1 + \mathbf{x}_2) \\
= -\frac{1}{2}\lambda(\mathbf{x}_1 - \mathbf{x}_2)^{\mathsf{T}}(\mathbf{x}_1 + \mathbf{x}_2) \\
= -\frac{\lambda}{2}(\|\mathbf{x}_1\|_2^2 - \|\mathbf{x}_2\|_2^2)
\end{aligned} \tag{5}$$

We can also eliminate λ by combining our constraints differently.

$$\begin{aligned}
2 &= (\mathbf{w}^{\mathsf{T}}\mathbf{x}_1 + b) - (\mathbf{w}^{\mathsf{T}}\mathbf{x}_2 + b) \\
&= \mathbf{w}^{\mathsf{T}}(\mathbf{x}_1 - \mathbf{x}_2) \\
&= \lambda(\mathbf{x}_1 - \mathbf{x}_2)^{\mathsf{T}}(\mathbf{x}_1 - \mathbf{x}_2) \\
\lambda &= \frac{2}{(\mathbf{x}_1 - \mathbf{x}_2)^{\mathsf{T}}(\mathbf{x}_1 - \mathbf{x}_2)}
\end{aligned}$$

(1pt) Decision Making with Bayes

The Bayes decision rule describes the approach we take to choosing a class ω for a data point $\mathbf{x}$. This can be achieved modeling $\mathbb{P}(\omega|\mathbf{x})$ or $\mathbb{P}(\mathbf{x}|\omega)\mathbb{P}(\omega)/\mathbb{P}(\mathbf{x})$. Compare and contrast these two approaches to modeling and discuss the advantages and disadvantages. For the latter model, why might knowing $\mathbb{P}(\mathbf{x})$ be useful?

Solution: This question is asking use to consider the tradeoffs between a generative and discriminative classifier. A discriminative classifier attempts to model $\mathbb{P}(\omega|\mathbf{x})$, which is typically easier to estimate since we done not make assumptions or directly try to model the distribution of the data. A generative classifier, while generally being a more complex model, allows us to directly model the distribution of the data. On top of obtaining the Bayes classifier, we can directly compute $\mathbb{P}(\mathbf{x})$. This term is useful for looking for outliers in the data.

Part B: Practice (20pts)

You are free to use functions already implemented in Matlab, Python or R. I recommend using Python's Scikit-learn (`http://scikit-learn.org/stable/`) as is implements most of the methods we will be discussing in this course…as well as problems in this homework!

(15pts) Half Moon Data Generator and Linear Classifier

Write a script to generate the "half moon" data set shown in Figure 1. Implement a linear classifier (e.g., logistic regression or sign$(\mathbf{w}^{\mathsf{T}}\mathbf{x})$) to discriminate between the two classes. Show the decision boundary between the two classes. For example, one approach could be to plot the posterior over a 2D grid where the data lie. Note that you must use a linear classifier. I have posted example code on Github. Matlab also has many built in functions to implement linear classifiers and naïve Bayes.

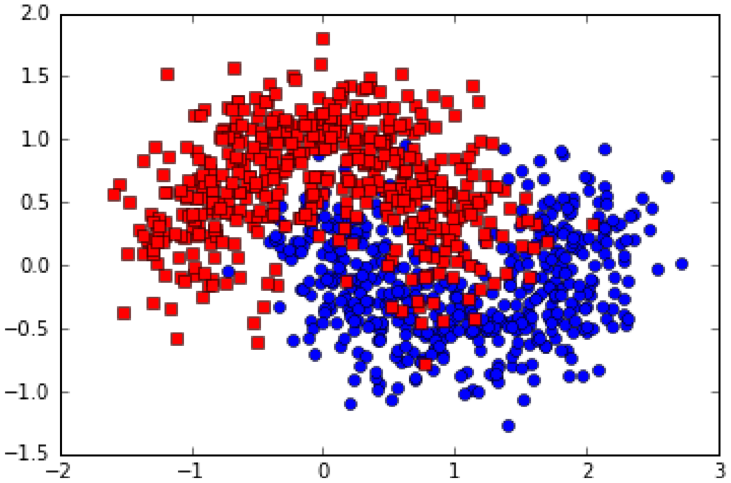

Figure 1: Example of the half moon data set.

Solution: See Github.

```
# SOLUTION FROM: Heriberto Encinas
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures

#define poly
poly = PolynomialFeatures(degree=3)
#build classifier
lr = LogisticRegression()
#define number of samples
nsamp = 2000

#generate dataset
halfmoon = datasets.make_moons(nsamp, noise = 0.3)

#separate features and labels
X,y = halfmoon

#plot original data
plt.figure()
plt.title('Original Dataset')
plt.scatter(X[np.where(y==0)[0], 0], X[np.where(y==0)[0], 1], s= 30, c='r', marker='s')
plt.scatter(X[np.where(y==1)[0], 0], X[np.where(y==1)[0], 1], s= 30, c='b', marker='o')
plt.savefig('figures/fig1B1')

#apply transformation
X_new = poly.fit_transform(X)

#fit classifier
lr.fit(X_new,y)
#get labels for transformed dataset
```

```
y_hat=lr.predict(X_new)

#plot transformed dataset
plt.figure()
plt.title('Transformed Dataset')
plt.scatter(X[np.where(y_hat==0)[0], 0], X[np.where(y_hat==0)[0], 1], s= 30, c='r',
    marker='s')
plt.scatter(X[np.where(y_hat==1)[0], 0], X[np.where(y_hat==1)[0], 1], s= 30, c='b',
    marker='o')
plt.savefig('figures/fig2B1')
plt.show()
```

(5pts) Naïve Bayes Spam Filter

A Spam data set has been uploaded to the ECE523 Github page (use `data/spambase_train.csv`). Using whatever library you wish, implement a naïve Bayes classifier and report the 5-fold cross validation error.

Solution:

```
# SOLUTION FROM: Heriberto Encinas
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import ShuffleSplit
from sklearn.preprocessing import MinMaxScaler

#load dataset
dataset = np.loadtxt('spambase.csv', delimiter=',')

#separate features and labels
X = dataset[:,0:57]
y = dataset[:,57]

#preprocess data
x = MinMaxScaler().fit_transform(X)

#build classifier
clf = MultinomialNB()

#build a cross validation shuffler, this ensures
#a meaningful cv result
cv = ShuffleSplit(n_splits=5)

#compute cross validation scores
scores = cross_val_score(clf, x, y, cv=cv, verbose=5)

#print information
print '------------------------------------------'
print 'Average (Accuracy) Score:................%.6f' %np.mean(scores)
print 'Average Error:...........................%.6f' %(1-np.mean(scores))
```

(3pts) Bonus: Comparing Classifiers

A text file, `hw1-scores.txt`, containing classifier errors measurements has been uploaded to D2L. Each of the columns represents a classifier and each row a data set that was evaluated. Are all of the classifiers performing equally? Is there one or more classifiers that is performing better than the others? Your response should be backed by statistics. Suggested reading:

- Janez Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," Journal of Machine Learning Research, vol. 7, 1–20.

Read the abstract to get an idea about the theme of comparisons. Sections 3.1.3 and 3.2.2 can be used to answer the question posed here.