Machine Learning and Big Data

Mock Exam

Question 1 — Classification

Consider the following sample of the two random variables X and Y:

Obs.	X	Y
1	1	3
2	2	3
3	3	-1
4	4	3
5	6	-1
6	8	-1

1. If we are interested in predicting Y given X, is this a regression or classification problem? Justify your answer. (10 marks)

2. Using 2-nearest neighbor methodology, predict Y given $X = -1$ and $X = 7$. (15 marks)

3. Using some other learner, an estimation yields

$$\hat{Y} = 3I\left(X \leqslant 3\right) - I(X > 3)$$

   Compute the training error rate. (15 marks)

Question 2 — Classification using discriminant analysis

Consider the problem where the output Y is binary,

$$\Pr\left(Y = 1\right) = \pi = 1 - \Pr\left(Y = -1\right)$$

and where the input X conditional on the output follows the exponential distribution. In particular, conditional on $Y = 1$, X has density $\lambda_1 \exp\left(-\lambda_1 x\right)$ and conditional on $Y = -1$, X has density $\lambda_2 \exp\left(-\lambda_2 x\right)$.

1. Is the odds ratio linear or quadratic in x? Or does it have some other functional form?

   (20 marks)

2. Describe a detailed algorithm for classifying data arising from that model. (10 marks)

3. Compare this approach to the traditional Linear discriminant analysis approach.

   (10 marks)

Remark: Note that when a (positive) random variable has density $\lambda \exp\left(-\lambda x\right)$, then a common estimator for λ is the maximum likelihood estimator which is equal to the inverse of the sample average.

Question 3 **Interpreting simple R commands**

Describe what each of the following R functions do and where it would be useful. Justify your answer mathematically if necessary.

1. (5 marks)

```
d1 = function(p)
{
return( log(p/(1-p)) )
}
```

2. (5 marks)

```
d2 = function(x,xi)
{
return( (x-xi)(x>xi) )
}
```

3. (5 marks)

```
d3 = function(w)
{
return( w/sum( w ) )
}
```

4. (5 marks)

```
d4 = function(a,b)
{
return( ( 1 + sum(a*b) )^2 )
}
```

Solution. Question 1

1. It is clearly a classification problem since Y is a discrete variables. It takes only two values: -1 and 3.

2. Denoting by x_1 to x_6 the values taken by X, then

$$\begin{aligned} d(-1,x_1) &= \sqrt{|-1-1|^2} = |-2| = 2 \\ d(-1,x_2) &= |-1-2| = 3 \\ d(-1,x_3) &= |-1-3| = 4 \\ d(-1,x_4) &= |-1-4| = 5 \\ d(-1,x_5) &= |-1-6| = 7 \\ d(-1,x_6) &= |-1-8| = 9 \end{aligned}$$

   Thus the 2 nearest neighbors to -1 are observations $\{1,2\}$.

$$\Pr\{Y=3|X=-1\} = \frac{1}{2}\sum_{i\in\{1,2\}} I(y_i=3) = 1$$

   and we predict $Y=3$ as $\Pr\{Y=3|X=-1\} > \frac{1}{2}$.

   Similarly,

$$\begin{aligned} d(7,x_1) &= |7-1| = 6 \\ d(7,x_2) &= |7-2| = 5 \\ d(7,x_3) &= |7-3| = 4 \\ d(7,x_4) &= |7-4| = 3 \\ d(7,x_5) &= |7-6| = 1 \\ d(7,x_6) &= |7-8| = 1 \end{aligned}$$

   Thus the 2 nearest neighbors to 7 are observations $\{5,6\}$.

$$\Pr\{Y=-1|X=7\} = \frac{1}{2}\sum_{i\in\{5,6\}} I(y_i=-1) = 1$$

   ane we predict $Y=-1$ as $\Pr\{Y=-1|X=7\} > \frac{1}{2}$.

3. Computing the prediction for $x_1=1$ yields $\hat{y}_1 = 3I(x_1 \leqslant 3) - I(x_1 > 3) = 3\times 1 - 0 = 3$. Computing all the other predictions yields the following table

Obs.	X	Y	$\hat{Y}$
1	1	3	3
2	2	3	3
3	3	-1	3
4	4	3	-1
5	6	-1	-1
6	8	-1	-1

   and thus the training error rate is given by

$$\frac{1}{6}\sum_{i=1}^{6} I(y_i \neq \hat{y}_i) = \frac{2}{6} = 0.33333.$$

Solution. Question 2

1. The probability $\Pr(Y=k|X=x)$ can be computed by Bayes rule

$$\Pr(Y=k|X=x) = \frac{\Pr(Y=k)p_k(x)}{\sum_{k'}\Pr(Y=k')p_{k'}(x)}.$$

When computing the odds ratio, the denominator (which is common to both probabilities) will simplify out to yield

$$\begin{aligned}\log\left(\frac{\Pr(Y=1|X=x)}{\Pr(Y=-1|X=x)}\right) &= \log\left(\frac{\pi\lambda_1\exp(-\lambda_1 x)/\sum_{k'}\Pr(Y=k')p_{k'}(x)}{(1-\pi)\lambda_2\exp(-\lambda_2 x)/\sum_{k'}\Pr(Y=k')p_{k'}(x)}\right)\\ &= \log\left(\frac{\pi\lambda_1\exp(-\lambda_1 x)}{(1-\pi)\lambda_2\exp(-\lambda_2 x)}\right)\\ &= \log\frac{\pi\lambda_1}{(1-\pi)\lambda_2}+x(\lambda_2-\lambda_1)\end{aligned}$$

Thus the log-odds ratio is linear in x. It takes the form c_0+c_1x with $c_0=\log\frac{\pi\lambda_1}{(1-\pi)\lambda_2}$ and $c_1=\lambda_2-\lambda_1$.

2. Compute $\hat{\lambda}_1$ and $\hat{\lambda}_2$ by dividing the sample in two parts (One subsample when $y_i=1$ and the other when $y_i=-1$.) Denote by n_1 the size of the first subsample and by n_2 the size of the second.

   Thus compute $\hat{\lambda}_1=\frac{n_1}{\sum_{i:\,y_i=1}x_i}$ and $\hat{\lambda}_2=\frac{n_2}{\sum_{i:\,y_i=-1}x_i}$.

   Obviously, the probability that $Y=1$ can be estimated through $\hat{\pi}=\frac{n_1}{n}$ (the proportion of 1s among the whole sample).

   Finally, at point x, predict $\hat{Y}=1$ if $\Pr(Y=1|X=x)>\Pr(Y=-1|X=x)$ which is equivalent to $\log\frac{\hat{\pi}\hat{\lambda}_1}{(1-\hat{\pi})\hat{\lambda}_2}+x\left(\hat{\lambda}_2-\hat{\lambda}_1\right)>0$ and $\hat{Y}=-1$ otherwise. The decision boundary is given by those values of x such that $\log\frac{\hat{\pi}\hat{\lambda}_1}{(1-\hat{\pi})\hat{\lambda}_2}+x\left(\hat{\lambda}_2-\hat{\lambda}_1\right)=0$ which is just the equation of a line (hyperplane).

3. Both discriminating functions are linear. The main difference is that the classical discriminant analysis assumes normality of the inputs conditionally on the outputs whereas in this particular model, we are working with exponential inputs.

Solution. Question 3

1. This computes the function $f(p)=\log\left(\frac{p}{1-p}\right)$ which is just the logit transformation.

   This is useful for logistic regression calculation.

2. The expression `(x>xi)` is a conditional statement that should return `TRUE` or `FALSE`. In a numeric calculation, it should return either 1 or 0.

   This means that for two inputs x and ξ, the function computes $(x-\xi)I(x>\xi)$ which is the formula for a basis function of a linear spline.

   This is useful for nonparametric nonlinear regression.

   Notice also that the function could accomodate a vector input `x` with no particular problem.

3. The expression `sum(w)` computes the sum of elements in a (numeric) list.

   This means that for a vector input w (say of length n), the function computes a vector whose ith element is $\frac{w_i}{\sum_{i'=1}^{n}w_{i'}}$.

   This is useful when we want a vector of positive numbers to have a sum of 1 (this is called normalizing the sum). This also would explicitly make the vector a vector of weights. An example where this could be used is in the adaboost algorithm or for kernel regression.

4. The function takes two vector inputs a and b (say of length p) and returns $(1+\sum_{i=1}^{p}a_ib_i)^2$. This is the formula for the quadratic kernel used in support vector machines.

   This is useful when doing classification through nonlinear decision boundaries through support vector machines.