ENGR 421/DASC 521: Introduction to Machine Learning
Fall 2021 Midterm – Solution Key

Question 1:
"Introduction to Machine Learning" (ENGR 421/DASC 521) course is attended by students majoring in engineering and some students that do not major in engineering. 80% of the engineering students and 60% of the non-engineering students passed the course. 10% of the entire class are non-engineering students. What is the percentage of engineering students among those that actually passed the course?

Let S denote that a student passed the exam. Let E denote that a student is an engineering major, and N denote the otherwise.

$$\begin{aligned}\Pr(E) &= 0.9\\ \Pr(N) &= 0.1\\ \Pr(S|E) &= 0.8\\ \Pr(S|N) &= 0.6\end{aligned}$$

$$\begin{aligned}\Pr(E|S) =& \frac{\Pr(S|E)\Pr(E)}{\Pr(S)} = \frac{\Pr(S|E)\Pr(E)}{\Pr(S|E)\Pr(E)+\Pr(S|N)\Pr(N)}\\ =& \frac{0.8\times 0.9}{0.8\times 0.9+0.6\times 0.1}\\ =& \frac{72}{78} = \frac{12}{13} \approx 92.30\%\end{aligned}$$

Question 2:
Consider a data set in which each data point $\boldsymbol{x}_i \in \mathbb{R}^D$ is associated with a read-valued output $y_i \in \mathbb{R}$ and a weighting factor $r_i > 0$, so that the sum-of-squares error function becomes

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N} r_i(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2.$$

Find an expression for the solution $\boldsymbol{w}^\star$ that minimizes this error function.

$$\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = -\sum_{i=1}^{N} r_i(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i = 0$$

$$\begin{aligned}\sum_{i=1}^{N} r_i y_i \boldsymbol{x}_i &= \left(\sum_{i=1}^{N} r_i \boldsymbol{x}_i \boldsymbol{x}_i^\top\right)\boldsymbol{w}\\ \boldsymbol{w}^\star &= \left(\sum_{i=1}^{N} r_i \boldsymbol{x}_i \boldsymbol{x}_i^\top\right)^{-1}\left(\sum_{i=1}^{N} r_i y_i \boldsymbol{x}_i\right)\end{aligned}$$

Question 3:
Let us assume X is a random variable with the following probability distribution:

$$p(x) = 2\alpha x \exp(-\alpha x^2)$$

where x is a positive real number and α is a positive parameter. You are given a training data set consisting of N examples as $\mathcal{X} = \{x_i\}_{i=1}^N$.

(a) Describe a maximum likelihood approach to infer α an write down the log-likelihood objective for this problem.

(b) Find the maximum likelihood solution for α.

(a) By assuming the data points are independent from each other, we can write down the likelihood function as follows:

$$\text{likelihood} = \prod_{i=1}^{N} p(x_i) = \prod_{i=1}^{N} 2\alpha x_i \exp(-\alpha x_i^2)$$

$$\text{log-likelihood} = \sum_{i=1}^{N} \big(\log(2) + \log(\alpha) + \log(x_i) - \alpha x_i^2\big)$$

(b) To maximize log-likelihood, we need to minimize $\sum_{i=1}^{N}(\alpha x_i^2 - \log(\alpha))$ with respect to α.

$$\frac{\partial \sum_{i=1}^{N}(\alpha x_i^2 - \log(\alpha))}{\partial \alpha} = \sum_{i=1}^{N} \frac{\partial(\alpha x_i^2 - \log(\alpha))}{\partial \alpha}$$

$$= \sum_{i=1}^{N} \left(x_i^2 - \frac{1}{\alpha}\right)$$

$$\sum_{i=1}^{N} \left(x_i^2 - \frac{1}{\alpha}\right) = 0 \Rightarrow \frac{N}{\alpha^\star} = \sum_{i=1}^{N} x_i^2$$

$$\alpha^\star = \frac{N}{\sum_{i=1}^{N} x_i^2}$$

Question 4:
Let $p(x|y=1) \sim N(x; \mu_1, \sigma^2)$ and $p(x|y=2) \sim N(x; \mu_2, \sigma^2)$, that is, the variances are equal. For the linear discriminant

$$g(x) = \log \frac{P(y=1|x)}{P(y=2|x)} = wx + w_0,$$

write w and w_0 in terms of μ_1, μ_2, σ, and the prior probabilities, $P(y=1)$, $P(y=2)$.

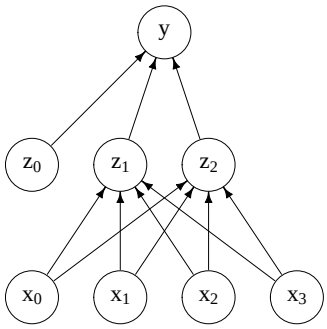
$$g(x) = \log \frac{P(y=1|x)}{P(y=2|x)} = \log \frac{p(x|y=1)P(y=1)}{p(x|y=2)P(y=2)}$$

$$
\begin{aligned}
&= \log \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma^2}\right) P(y=1)}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma^2}\right) P(y=2)} \\
&= -\frac{(x-\mu_1)^2}{2\sigma^2} + \frac{(x-\mu_2)^2}{2\sigma^2} + \log \frac{P(y=1)}{P(y=2)} \\
&= \underbrace{\frac{\mu_1 - \mu_2}{\sigma^2}}_{w} x + \underbrace{\frac{\mu_2^2 - \mu_1^2}{2\sigma^2} + \log \frac{P(y=1)}{P(y=2)}}_{w_0}
\end{aligned}
$$

Question 5:
You are asked to design a multilayer perceptron which receives three binary-valued (i.e., 0 or 1) inputs x_1, x_2, and x_3, and outputs 1 if exactly two of the inputs are 1, and outputs 0 otherwise. All of the units use a hard threshold activation function:

$$
s(a) = \begin{cases} 1 & \text{if } a \geq 0, \\ 0 & \text{if } a < 0. \end{cases}
$$

Specify weights which correctly implement this function. You do not need to explain your solution. (Hint: one of the hidden units should activate if two or more inputs are on, and the other should activate if all of the inputs are on.)



$$
\mathbf{W} = \begin{bmatrix} w_{10} & w_{11} & w_{12} & w_{13} \\ w_{20} & w_{21} & w_{22} & w_{23} \end{bmatrix} = \begin{bmatrix} -1.5 & +1 & +1 & +1 \\ -2.5 & +1 & +1 & +1 \end{bmatrix}
$$
$$
\boldsymbol{v} = \begin{bmatrix} v_0 & v_1 & v_2 \end{bmatrix} = \begin{bmatrix} -0.5 & +1 & -1 \end{bmatrix}
$$

Question 6:
Does it make sense to initialize all weights in a multilayer perceptron to zero? Justify your answer.

No. If all weights are equal, their gradients will be the same as well, and the gradients will possibly vanish. Hence all neurons will learn the same feature.

Question 7:
Assume we have a set of data from patients who have visited a hospital during the year. A set of features (e.g., temperature, height, etc.) have been also extracted for each patient. Our goal is to decide whether a new visiting patient has any of diabetes, heart disease, or Alzheimer (a patient can have one or more of these diseases). We have decided to use a neural network to solve this problem. We have two choices: either to train a separate neural network for each of the diseases or to train a single neural network with one output neuron for each disease, but with a shared hidden layer. Which method do you prefer? Justify your answer.

Neural network with a shared hidden layer can capture dependencies between diseases. It can be shown that in some cases, when there is a dependency between the output nodes, having a shared node in the hidden layer can improve the accuracy.

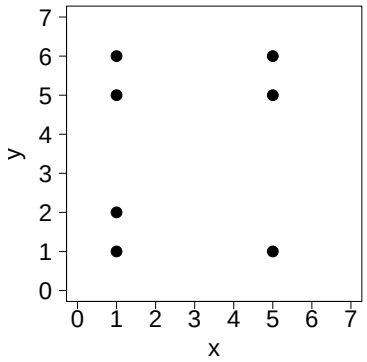
If there is no dependency between diseases (output neurons), then we would prefer to have a separate neural network for each disease.

Question 8:
List two drawbacks of the naive density estimator. What is the advantage of the Parzen windows density estimator compared to the naive density estimator?

Not continuous, curse of dimensionality, zero density if no training points in the bin, etc.

The Parzen windows density estimator gives smoother probability density functions than the naive density estimator.

Question 9:
The following figure shows a data set with one-real valued input x and one real-valued output y. There are seven training samples.



Suppose you are training a kernel smoother using some unspecified kernel function. The only thing you know about the kernel function is that it is a monotonically decreasing function of distance that decays to zero at a distance of 3 units (and is strictly greater than zero at a distance of less than 3 units).

(a) What is the predicted value of y when $x = 1$? Justify your answer.

(b) What is the predicted value of y when $x = 3$? Justify your answer.

(c) What is the predicted value of y when $x = 4$? Justify your answer.

(a) $\widehat{y} = (1+2+5+6)/4$ since the distance to data points at $x = 5$ is 4 (they have no effect) and the distance to data points at $x = 1$ is 0 (they have equal effects).

(b) $\widehat{y} = (1+2+5+6+1+5+6)/7$ since the distance to all data points is 2 (they have equal effects).

(c) $\widehat{y} = (1+5+6)/3$ since the distance to data points at $x = 1$ is 3 (they have no effect) and the distance to data points at $x = 5$ is 1 (they have equal effects).

Question 10:
The naive density estimator $\widehat{p}(x)$ for data point $x \in \mathbb{R}$ is given by

$$\widehat{p}(x) = \frac{1}{Nh} \sum_{i=1}^{N} w\left(\frac{x - x_i}{h}\right)$$

$$w(u) = \begin{cases} 1 & \text{if } |u| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

where the training data set consists of training samples $x_1, x_2, \ldots, x_N$.

(a) Discuss how the value of h affects the naive density estimator. How should the value of h be varied as N changes?

(b) Show that the naive density estimator $\widehat{p}(x)$ is indeed a valid probability density function.

(a) small $h \Rightarrow$ overfitting, large $h \Rightarrow$ underfitting

small $N \Rightarrow$ we can increase h, large $N \Rightarrow$ we can decrease h

(b) $\widehat{p}(x) \geq 0$ is trivially satisfied.

$$\begin{aligned}
\int_{x\in\mathbb{R}} \widehat{p}(x)\mathrm{d}x &= \int_{x\in\mathbb{R}} \frac{1}{Nh}\sum_{i=1}^{N} w\left(\frac{x-x_i}{h}\right)\mathrm{d}x \\
&= \frac{1}{Nh}\sum_{i=1}^{N}\int_{x\in\mathbb{R}} w\left(\frac{x-x_i}{h}\right)\mathrm{d}x \\
&= \frac{1}{Nh}\sum_{i=1}^{N}\left[\int_{x=-\infty}^{x_i-h/2} 0\mathrm{d}x + \int_{x=x_i-h/2}^{x_i+h/2} 1\mathrm{d}x + \int_{x=x_i+h/2}^{+\infty} 0\mathrm{d}x\right] \\
&= \frac{1}{Nh}\sum_{i=1}^{N}[0+h+0] = \frac{1}{Nh}Nh = 1
\end{aligned}$$

Question 11:
Instead of entropy in decision trees, one can use the Gini index or the misclassification error. You have a data set with 400 positive examples (i.e., 400^+) and 400 negative examples (i.e., 400^-). Suppose that you have two possible splits for a decision node m. The first splitting choice results in L_m: $(300^+, 100^-)$ and R_m: $(100^+, 300^-)$. The second splitting choice results in L_m: $(200^+, 400^-)$ and R_m: $(200^+, 0^-)$.

(a) Calculate the impurity after these two choices of splitting using the Gini index and the misclassification error?

(b) Which of these two choices of splitting would be preferred and why?

(a) Gini index $= 2p(1-p)$ Misclassification error $= \min(p, 1-p)$

$$\begin{aligned}
\text{Gini}(1^{st}\text{ split}) &= \frac{400}{800}\left(2\frac{300}{400}\frac{100}{400}\right) + \frac{400}{800}\left(2\frac{100}{400}\frac{300}{400}\right) = \frac{3}{8} \\
\text{Gini}(2^{nd}\text{ split}) &= \frac{600}{800}\left(2\frac{200}{600}\frac{400}{600}\right) + \frac{200}{800}\left(2\frac{200}{200}\frac{0}{200}\right) = \frac{1}{3} \\
\text{Error}(1^{st}\text{ split}) &= \frac{400}{800}\min\left(\frac{300}{400}, \frac{100}{400}\right) + \frac{400}{800}\min\left(\frac{100}{400}, \frac{300}{400}\right) = \frac{1}{4} \\
\text{Error}(2^{nd}\text{ split}) &= \frac{600}{800}\min\left(\frac{200}{600}, \frac{400}{600}\right) + \frac{200}{800}\min\left(\frac{200}{200}, \frac{0}{200}\right) = \frac{1}{4}
\end{aligned}$$

(b) Gini index prefers the second split, whereas there is no difference in terms of misclassification error.

Question 12:
Explain the difference between prepruning and postpruning strategies for decision trees. Why does postpruning work better than prepruning in practice?

Stopping the tree construction early on before it is full is called prepruning the tree. In postpruning, we try to find and prune unnecessary subtrees. We grow the tree full until all leaves are pure and we have no training error. We then find subtrees that cause overfitting and we prune them.

Postpruning generally leads to more accurate trees since, from the initial labeled set, we set aside a pruning set, unused during training, and postpruning is performed by looking at the performance on this set.