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

Question 1:
Explain why it is not a good idea to use standard regression algorithms on classification problems. What makes the logistic regression algorithm more suitable for classification problems?

Regression algorithms may produce arbitrary outputs (i.e., smaller than 0 or greater than 1). However, logistic regression algorithm is guaranteed to produce probability values between 0 and 1.

Question 2:
Suppose that you are given the source code of a binary classification algorithm. Describe how you can use the same source code to obtain a multiclass classification algorithm.

We can use the source code of a binary classification algorithm to obtain a multiclass classification algorithm either by one-versus-all or one-versus-other approach.

Question 3:
The softmax function is defined as

$$\widehat{y}_c = \frac{\exp(a_c)}{\sum_{b=1}^{K} \exp(a_b)}.$$

In softmax implementations, we usually use the following formula:

$$\widehat{y}_c = \frac{\exp(a_c - m)}{\sum_{b=1}^{K} \exp(a_b - m)},$$

where $m = \max(a_1, \ldots, a_K)$. What is the purpose of this modification?

We use this modification to avoid overflow problems when calculating the softmax function.

Question 4:
It is always a good idea to initialize the weights in linear discrimination with random values close to 0. They are generally drawn uniformly from the interval $[-0.01, 0.01]$. What is the reason for such an initialization strategy?

If the initial weights are large in magnitude, the weighted sum may also be large and may saturate the sigmoid. If the initial weights are close to 0, the weighted sum will stay close to 0 where the derivative is nonzero and an update can take place. If the weighted sum is large in magnitude, the derivative of the sigmoid will be almost 0 and weights will not be updated.

Question 5:
In gradient descent, we use a constant learning rate η that is the same for all weights and for all epochs. How can we make it adaptive?

In gradient descent, the learning factor η determines the magnitude of change to be made in the parameter. A very early heuristic was to make it adaptive for faster convergence, where it is kept large when learning takes place and is decreased when learning slows down. Thus we increase η by a constant amount (or keep it unchanged) if the error on the training set decreases and decrease it geometrically if it increases.
In a deep network with many hidden layers and units, the contributions of weights to the error are not the same, and one can adapt the learning factor separately for each weight, depending on the convergence along that direction. The idea in such methods is to accumulate the past error gradient magnitudes for each weight and then make the learning factor inversely proportional to that. In directions where the gradient is small, the learning factor is larger to compensate, and it can be smaller along directions where the gradient is already large.

Question 6:
Consider a multilayer perceptron architecture with one hidden layer where there are also direct weights from the inputs directly to the output units. Explain when such a structure would be helpful and how it can be trained.

It can be trained as usual by backpropagation. The direct weights from the input to the outputs is a perceptron and the rest is a multilayer perceptron, each being trained together as usual. The perceptron that contains the direct weights from the inputs to the outputs is a linear model and can be thought of as implementing the linear component of whatever we are trying to learn; the multilayer perceptron then learns to correct this default linear model.

Question 7:
What are the advantages of using the Mahalanobis distance over the Euclidean distance?

- Mahalanobis distance: $\sqrt{(\boldsymbol{x}-\widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}})}$
- Euclidean distance: $\sqrt{(\boldsymbol{x}-\widehat{\boldsymbol{\mu}})^\top (\boldsymbol{x}-\widehat{\boldsymbol{\mu}})}$

The Mahalanobis distance normalizes features based on variance.

- This makes all independent attributes equally important.
- It alleviates problem caused by using different scales.
- It downplays the contribution of correlated attributes in distance computations.

Question 8:
In generating a univariate tree, a discrete attribute with K possible values can be represented by K 0/1 dummy variable and then treated as K separate numeric attributes. What are the advantages and disadvantages of this approach?

A discrete variable:

- Split decisions are complex.
- Tree size is small.

K binary variables:

- Split decisions become easier.
- Tree may grow large.

Question 9:
Consider a data set in which each data point $\boldsymbol{x}_i \in \mathbb{R}^D$ is associated with a real-valued output $y_i \in \mathbb{R}$. Instead of using the original sum-of-squares error function, we modify the error function by adding a regularization term:

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

where $\lambda > 0$ is the regularization coefficient.

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

(b) Describe a method to choose the best λ parameter from a set of candidate values.

(a) We should take the derivative of the error function and set it equal to zero.

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

$$\sum_{i=1}^{N} y_i \boldsymbol{x}_i = \left(\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^\top + \lambda \mathbf{I}\right)\boldsymbol{w}$$

$$\boldsymbol{w}^\star = \left(\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^\top + \lambda \mathbf{I}\right)^{-1}\left(\sum_{i=1}^{N} y_i \boldsymbol{x}_i\right)$$

(b) We should check the performance on the validation set(s) to pick the best λ parameter.

Question 10:
You are given a univariate binary classification data set. The first class has the distribution $p(x|y=1) \sim \mathcal{N}(x; \mu_1, \sigma^2)$, and the second class has the distribution $p(x|y=2) \sim \mathcal{N}(x; \mu_2, \sigma^2)$. The prior probabilities of each class are $P(y=1) = P(y=2) = 1/2$. Show that the posterior probability $P(y=1|x)$ is of the form

$$P(y=1|x) = \frac{1}{1+\exp(-wx - w_0)}$$

and determine w and w_0 in terms of μ_1, μ_2, and σ^2.

$$\log\left[\frac{P(y=1|x)}{1-P(y=1|x)}\right] = \log\left[\frac{p(x|y=1)P(y=1)/p(x)}{p(x|y=2)P(y=2)/p(x)}\right]$$

$$= \log\left[\frac{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu_1)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma^2}\right)}\right]$$

$$
\begin{aligned}
&= \frac{(\mu_1 - \mu_2)}{\sigma^2} x + \frac{(\mu_2^2 - \mu_1^2)}{2\sigma^2} \\
P(y = 1|x) &= \frac{1}{1 + \exp\left[-\frac{(\mu_1 - \mu_2)}{\sigma^2} x - \frac{(\mu_2^2 - \mu_1^2)}{2\sigma^2}\right]} \\
w &= \frac{(\mu_1 - \mu_2)}{\sigma^2} \\
w_0 &= \frac{(\mu_2^2 - \mu_1^2)}{2\sigma^2}
\end{aligned}
$$

Question 11:
You are asked to design a text classifier to classify documents as either "sports" or "politics". You decide to represent each document as a vector of attributes describing the presence or absence of words.

$$\boldsymbol{x} = (\text{"goal"}, \text{"defence"}, \text{"office"}, \text{"strategy"})$$

The training data set constructed from eight documents is given below.

$$
\begin{aligned}
\boldsymbol{x}_1 &= (1,0,0,0) & y_1 &= \text{"sports"} \\
\boldsymbol{x}_2 &= (1,1,1,0) & y_2 &= \text{"sports"} \\
\boldsymbol{x}_3 &= (1,1,0,1) & y_3 &= \text{"sports"} \\
\boldsymbol{x}_4 &= (1,1,1,1) & y_4 &= \text{"politics"} \\
\boldsymbol{x}_5 &= (0,1,1,1) & y_5 &= \text{"politics"} \\
\boldsymbol{x}_6 &= (0,1,1,0) & y_6 &= \text{"politics"} \\
\boldsymbol{x}_7 &= (0,0,0,1) & y_7 &= \text{"politics"} \\
\boldsymbol{x}_8 &= (0,1,0,1) & y_8 &= \text{"politics"}
\end{aligned}
$$

Using a naive Bayes classifier trained with a maximum likelihood approach, what is the probability that a new document $\boldsymbol{x} = (1,0,1,0)$ is about "politics"?

$$
\begin{aligned}
\widehat{P}(\text{sports}) &= 3/8 & \widehat{P}(\text{politics}) &= 5/8 \\
\widehat{P}(\texttt{1...}|\text{sports}) &= 3/3 & \widehat{P}(\texttt{1...}|\text{politics}) &= 1/5 \\
\widehat{P}(\texttt{.1..}|\text{sports}) &= 2/3 & \widehat{P}(\texttt{.1..}|\text{politics}) &= 4/5 \\
\widehat{P}(\texttt{..1.}|\text{sports}) &= 1/3 & \widehat{P}(\texttt{..1.}|\text{politics}) &= 3/5 \\
\widehat{P}(\texttt{...1}|\text{sports}) &= 1/3 & \widehat{P}(\texttt{...1}|\text{politics}) &= 4/5 \\
\widehat{P}(\texttt{0...}|\text{sports}) &= 0/3 & \widehat{P}(\texttt{0...}|\text{politics}) &= 4/5 \\
\widehat{P}(\texttt{.0..}|\text{sports}) &= 1/3 & \widehat{P}(\texttt{.0..}|\text{politics}) &= 1/5 \\
\widehat{P}(\texttt{..0.}|\text{sports}) &= 2/3 & \widehat{P}(\texttt{..0.}|\text{politics}) &= 2/5 \\
\widehat{P}(\texttt{...0}|\text{sports}) &= 2/3 & \widehat{P}(\texttt{...0}|\text{politics}) &= 1/5
\end{aligned}
$$

$$
\begin{aligned}
\widehat{P}(\text{sports}|\texttt{1010}) &\propto (3/3)(1/3)(1/3)(2/3)(3/8) \\
\widehat{P}(\text{politics}|\texttt{1010}) &\propto (1/5)(1/5)(3/5)(1/5)(5/8) \\
\widehat{P}(\text{politics}|\texttt{1010}) &\approx 0.0975
\end{aligned}
$$

Question 12:
Show that the derivative of the softmax

$$\widehat{y}_c = \frac{\exp(a_c)}{\sum_{b=1}^{K} \exp(a_b)}$$

is $\partial \widehat{y}_c / \partial a_d = \widehat{y}_c(\delta_{cd} - \widehat{y}_d)$, where δ_{cd} is 1 if $c = d$ and 0 otherwise.

$$\frac{\partial \widehat{y}_c}{\partial a_d} = \frac{\frac{\partial \exp(a_c)}{\partial a_d}\left(\sum_{b=1}^{K} \exp(a_b)\right) - \exp(a_c)\frac{\partial\left(\sum_{b=1}^{K} \exp(a_b)\right)}{\partial a_d}}{\left(\sum_{b=1}^{K} \exp(a_b)\right)^2}$$

$$= \frac{\exp(a_c)\delta_{cd}}{\sum_{b=1}^{K} \exp(a_b)} - \frac{\exp(a_c)}{\sum_{b=1}^{K} \exp(a_b)} \frac{\exp(a_d)}{\sum_{b=1}^{K} \exp(a_b)} = \widehat{y}_c\delta_{cd} - \widehat{y}_c\widehat{y}_d = \widehat{y}_c(\delta_{cd} - \widehat{y}_d)$$

Question 13:
The kernel density estimator $\widehat{p}(\boldsymbol{x})$ for a multivariate data point $\boldsymbol{x} \in \mathbb{R}^D$ is written as follows:

$$\widehat{p}(\boldsymbol{x}) = \frac{1}{Nh^D} \sum_{i=1}^{N} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right)$$

$$K(\boldsymbol{u}) = \frac{1}{\sqrt{(2\pi)^D}} \exp\left(-\frac{\boldsymbol{u}^\top \boldsymbol{u}}{2}\right)$$

where the training data set consists of training samples $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$.

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

(b) Show that the kernel density estimator $\widehat{p}(\boldsymbol{x})$ 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) $\widehat{p}(\boldsymbol{x}) \geq 0$ is trivially satisfied.

$$\int_{x_1 \in \mathbb{R}} \cdots \int_{x_D \in \mathbb{R}} \widehat{p}(\boldsymbol{x}) \mathrm{d}x_1 \ldots \mathrm{d}x_D = \int_{x_1 \in \mathbb{R}} \cdots \int_{x_D \in \mathbb{R}} \frac{1}{Nh^D} \sum_{i=1}^{N} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right) \mathrm{d}x_1 \ldots \mathrm{d}x_D$$

$$= \frac{1}{Nh^D} \sum_{i=1}^{N} \int_{x_1 \in \mathbb{R}} \cdots \int_{x_D \in \mathbb{R}} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right) \mathrm{d}x_1 \ldots \mathrm{d}x_D$$

$$= \frac{1}{Nh^D} \sum_{i=1}^{N} \int_{u_1 = -\infty}^{+\infty} \cdots \int_{u_D = -\infty}^{+\infty} K(\boldsymbol{u})(h\mathrm{d}u_1) \ldots (h\mathrm{d}u_D)$$

$$= \frac{1}{Nh^D} \sum_{i=1}^{N} h^D = \frac{1}{Nh^D} Nh^D = 1$$

Question 14:
Instead of entropy in decision trees, one can use the Gini index or the misclassification error. You have a data set with 400 examples from o class (i.e., 400°), 400 examples from $\square$ class (i.e., $400^{\square}$), and 400 examples from $\triangle$ class (i.e., $400^{\triangle}$). Suppose that you have two possible splits for a decision node m. The first splitting choice results in L_m: $(300^{\circ}, 200^{\square}, 100^{\triangle})$ and R_m: $(100^{\circ}, 200^{\square}, 300^{\triangle})$. The second splitting choice results in L_m: $(400^{\circ}, 200^{\square}, 0^{\triangle})$ and R_m: $(0^{\circ}, 200^{\square}, 400^{\triangle})$.

- Gini index with $K > 2$ classes: $\sum_{c=1}^{K} \sum_{d>c} 2p_c p_d$
- Misclassification error: $1 - \max(p_1, p_2, \ldots, p_K)$

(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)

$$\text{Gini}(1^{st}\text{ split}) = \frac{6}{12}\left(2\frac{3}{6}\frac{2}{6} + 2\frac{3}{6}\frac{1}{6} + 2\frac{2}{6}\frac{1}{6}\right) + \frac{6}{12}\left(2\frac{1}{6}\frac{2}{6} + 2\frac{1}{6}\frac{3}{6} + 2\frac{2}{6}\frac{3}{6}\right) = \frac{11}{18}$$

$$\text{Gini}(2^{nd}\text{ split}) = \frac{6}{12}\left(2\frac{4}{6}\frac{2}{6} + 2\frac{4}{6}\frac{0}{6} + 2\frac{2}{6}\frac{0}{6}\right) + \frac{6}{12}\left(2\frac{0}{6}\frac{2}{6} + 2\frac{0}{6}\frac{4}{6} + 2\frac{2}{6}\frac{4}{6}\right) = \frac{8}{18}$$

$$\text{Error}(1^{st}\text{ split}) = \frac{6}{12}\left[1 - \max\left(\frac{3}{6}, \frac{2}{6}, \frac{1}{6}\right)\right] + \frac{6}{12}\left[1 - \max\left(\frac{1}{6}, \frac{2}{6}, \frac{3}{6}\right)\right] = \frac{1}{2}$$

$$\text{Error}(2^{nd}\text{ split}) = \frac{6}{12}\left[1 - \max\left(\frac{4}{6}, \frac{2}{6}, \frac{0}{6}\right)\right] + \frac{6}{12}\left[1 - \max\left(\frac{0}{6}, \frac{2}{6}, \frac{4}{6}\right)\right] = \frac{1}{3}$$

(b) Both the Gini index and misclassification error prefer the second split.