PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

DECEMBER 2021 EXAMINATIONS

CSC 2515HF

Duration: Please submit on MarkUs by Monday December 6 at 16:59.

Aids Allowed: You may consult the course slides and your notes.

Student Number: 1 0 0 3 0 6 3 2 2 1

Last (Family) Name(s): GARG

First (Given) Name(s): KOPAL

Please read *carefully* every reminder on this page.

- Fill out your name and student number above—do it now, don't wait!
- This take-home test consists of 12 questions on 21 pages (including this one).
- You may either (1) print these pages, answer each question directly on the examination paper, and then scan it and upload it as a PDF file, or (2) type your answers using your favourite word processor using the same order of questions as here. In the latter case, make sure you use the right question numbers and part (e.g., 4(a), 7(c), etc.). If you don't, you may not get the mark. Points will be deducted if we have a hard time reading your solutions.
- You should help us having a fair take-home test. You may consult the slides and your notes. We in fact encourage you to do so. But do not discuss the questions with anyone else (including on Piazza). And do not search for answer to these questions on the Internet.
- Do not share this take-home test with anyone else, even after the semester ends, as we may reuse some of these questions in the future.
- There will not be an auto-fail in this take-home test. But try hard to do a good job. This will be a good opportunity to practice your ML skills once more, and get feedback.
- **Late Submission:** There is no grace period or a gradual late penalty, unless there is an emergency.
- **Collaboration:** The take-home test must be done individually, and you **should not** collaborate with others (this is different from your homework assignments).

MARKING GUIDE

Nº 1:	_____/	12
Nº 2:	_____/	16
Nº 3:	_____/	5
Nº 4:	_____/	6
Nº 5:	_____/	9
Nº 6:	_____/	5
Nº 7:	_____/	8
Nº 8:	_____/	10
Nº 9:	_____/	8
Nº 10:	_____/	4
Nº 11:	_____/	15
Nº 12:	_____/	2
TOTAL:	_____/	100

Question 1. True or False Questions [12 MARKS]

For each of these questions, a correct answer is +1 point, an empty answer is 0 point, and a wrong answer is −1 point.

	Statement	True	False
(1)	Deciding whether a mushroom is edible or not based on its image is an example of a regression problem		F
(2)	A K-Nearest Neighbourhood method with larger K may generalize worse than a one with a smaller K	T	
(3)	Decision trees can achieve zero classification error on any training data (assuming each training data point is unique)	T	
(4)	A lower entropy implies lower uncertainty	T	
(5)	Linear regression has to be linear in both parameters and features	T	
(6)	Covariance matrix can have negative values	T	
(7)	Squared error loss in regression is only suitable if the target values are from a Gaussian distribution	T	
(8)	The ℓ_1 regularization cannot shrink parameters to zero, hence it can be used for the purpose of feature selection		F
(9)	PCA can be used as a feature selection method		F
(10)	Bagging decreases bias		F
(11)	AdaBoost cannot overfit		F
(12)	Projection to the highest variance direction is the same as projection to the direction that minimizes the total squared norm of each point to its projection	T	

Question 2. Short Answer Questions [16 MARKS]

Answer these questions concisely. In most cases, one or two sentences are enough.

Part (a) Fill in the blanks. [3 MARKS]

- Information Gain $IG(Y, X) = H(\ldots Y \ldots) - H(\ldots Y|X \ldots)$, where $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$ is the entropy.
- If X and Y are independent, then $H(X, Y) =$ $H(Y) + H(X)$
- If X and Y are independent, then $H(X|Y) =$ $H(X)$

Part (b) Fill in the blanks, using the following terms Posterior, Likelihood, Prior, Evidence. [3 MARKS]

$$\text{Posterior} = \frac{(\text{likelihood}) \times (\text{Prior})}{(\text{Evidence})}$$

Part (c) Explain the relationship between posterior probability and prior probability given the likelihood. [1 MARK] We can convert prior probability into posterior probability by incorporating the evidence provided by observed data. The likelihood function expresses how probable the observed data is for different settings. This can be written as:

$$\text{Posterior} \propto \text{likelihood} \times \text{prior}$$

Part (d) Why do we use validation set? Explain briefly. [1 MARK]
Validation set is used to get an unbiased estimate of model performance and to fine-tune model parameters while training. E.g. K-fold cross validation partitions initial trainin set into multiple gtrain-valid setsp to tune the model parameters to prevent overfitting & a hold out set to get a realistic estimate of model's generalization abilities

Part (e) What happens if we use the test set for tuning hyper-parameters? Explain briefly. [1 MARK]
This can lead to the model overfitting to the intricacies of the test set, and not being able to generalize over truly unseen data. The validity of a model performance typically depends on the independence of data used in training vs. testing. This would lead to a biased estimate of model performance, and if deployed, may cause unintended harms.

Part (f) Suppose that your classifier achieves poor accuracy on both the training and test sets. Which would be a better choice to try to improve the performance: Bagging or Boosting? Briefly justify your answer. [2 MARKS] If the model achieves poor accuracy on both training and testing set, it is underfit (high bias). Boosting would likely improve model performance. It is a sequential ensemble method that penalizes misclassifications by assigning them higher weights, thereby decreasing bias & error and building a strong predictive model. Bagging reduces variance by taking an average of ensemble models trained in parallel on random samples (with replacement) of original dataset. Therefore, boosting would be a better choice

Part (g) What is the effect of Boosting and Bagging on the bias/variance tradeoff? [4 MARKS]
The goal is to find a function that doesn't overfit to the training data while being sufficiently complex to capture general characteristics of the data. Keeping functions simple to avoid overfitting introduces bias, while increasing complexity causes overfitting and results in high variance in predictions. Boosting works well in underfit models with high bias, low variance. It reduces bias. Bagging reduces variance, while having no significant impact on bias.

Part (h) What is the main modelling assumption in the Naïve Bayes classifier? [1 MARK]

Naïve Bayes assumes features are conditionally independent given class label.

If x_i and x_j are independent conditioned on class label, t:

$$P(x_i, x_j \mid t) = P(x_i \mid t) P(x_j \mid t), \quad i \neq j$$

Question 3. Hyper-Parameter Identification [5 MARKS]

Consider the following ML methods. Write down one or more hyper-parameters used in each method.

Part (a) KNN (1 answer) [1 MARK]

K = Number of neighbors to inspect

Part (b) Deep Neural Networks (2 answers) [2 MARKS]

Learning rate, activation function,
Number of hidden layers, loss function, etc.

Part (c) Support Vector Machines (1 answer) [1 MARK]

choice of Kernel (linear: C, cost parameter
Polynomial, Radial: C, γ, radius of influence of SVs)

Part (d) PCA (1 answer) [1 MARK]

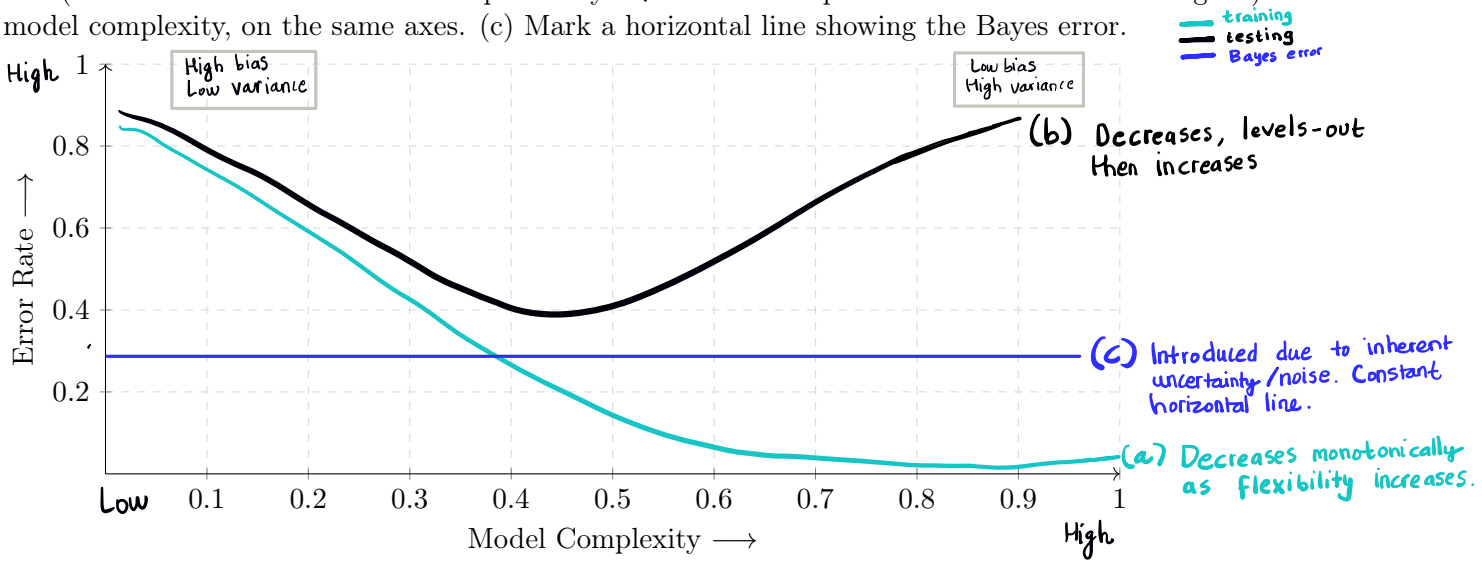
Number of principal components (% of variance explained by PCs)

Question 4. Training/Testing Error Curves [6 MARKS]

This question asks you to show your general understanding of underfitting and overfitting as they relate to model complexity and training set size. Given the provided axes, where both vertical and horizontal are scaled between 0 and 1, draw your graphs with increasing error upwards and increasing complexity/training set size rightwards. Make sure that you clearly mark each curve.

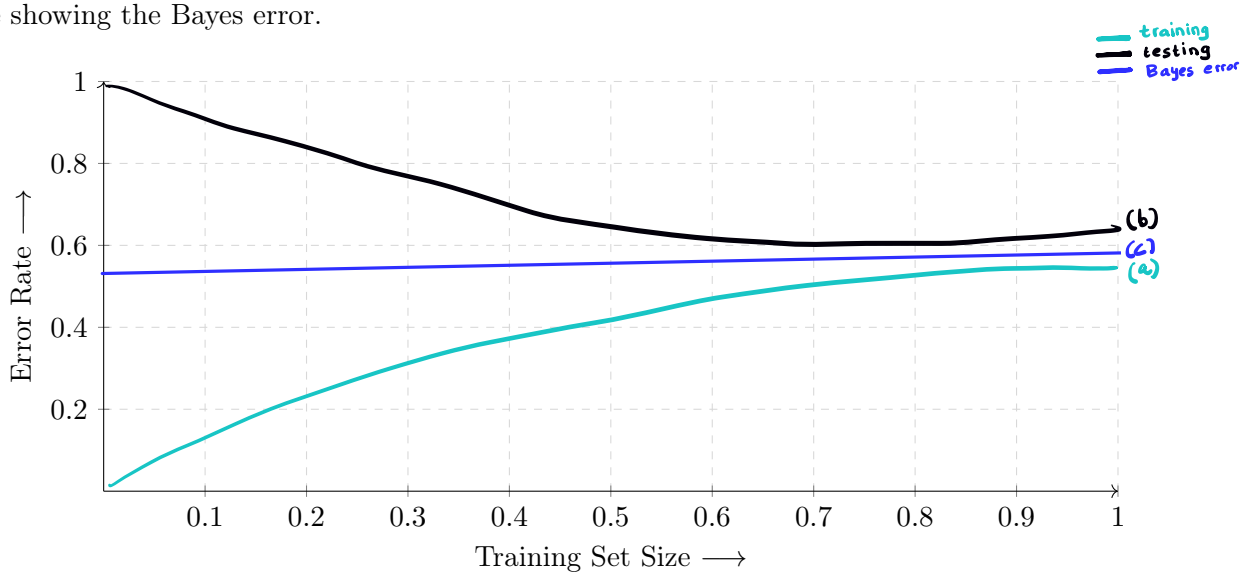
Part (a) [3 MARKS]

For a fixed training set size, (a) sketch a graph of the typical behaviour of training error rate versus model complexity in a learning system. (b) Add to this graph a curve showing the typical behaviour of test error rate (for an infinite test set drawn independently from the same input distribution as the training set) versus model complexity, on the same axes. (c) Mark a horizontal line showing the Bayes error.



Part (b) [3 MARKS]

For a fixed model complexity, (a) sketch a graph of the typical behaviour of training error rate versus training set size in a learning system. (b) Add to this graph a curve showing the typical behaviour of test error rate (again on an iid finite test set) versus training set size, on the same axes. (c) Mark a horizontal line showing the Bayes error.



Question 5. Decision Trees [9 MARKS]

The two dimensional input space shown in Figure 1 is partitioned into five (5) regions R1, R2,..., R5. We also show training data points consisting of circles and crosses.

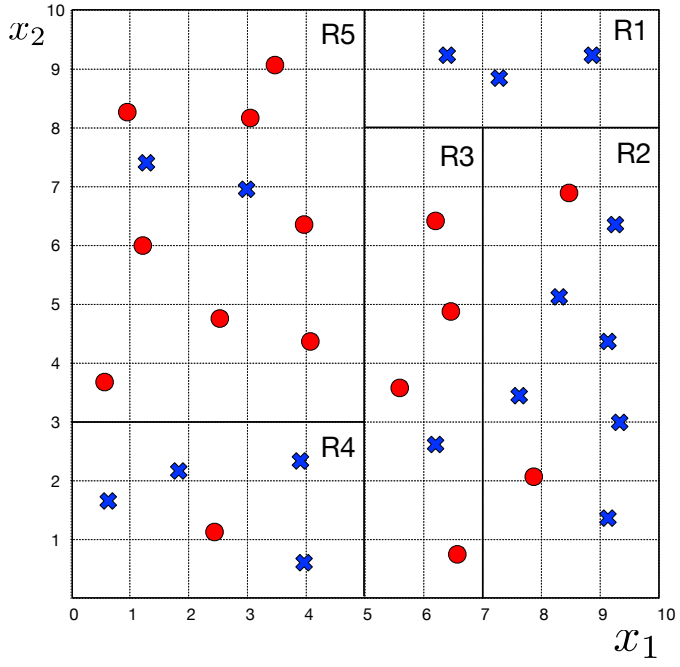
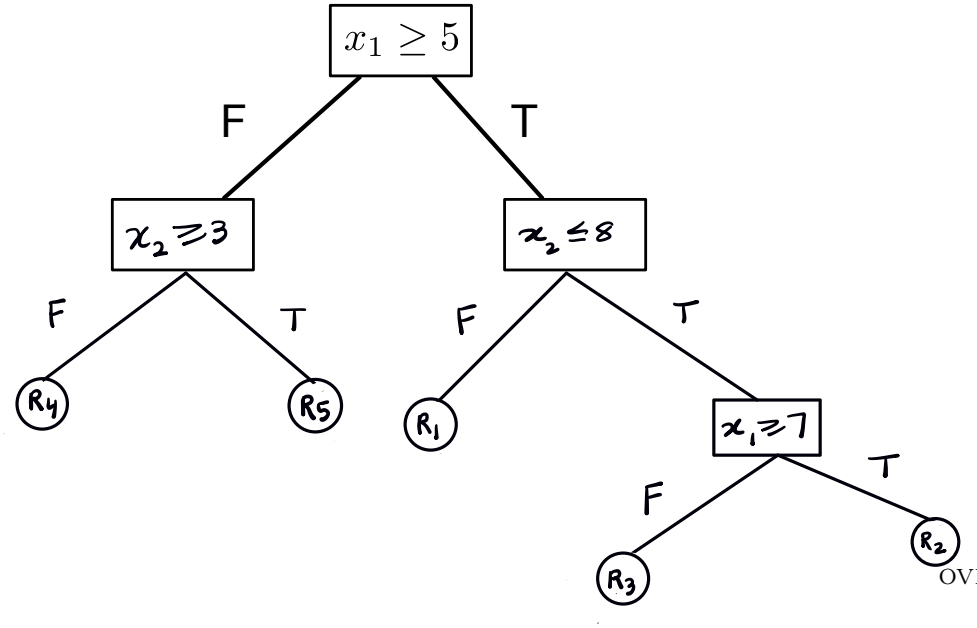


Figure 1: Training points consists of two classes of crosses (X) and circles (O), and five regions partitioning the input space.

Part (a) Design a decision tree that partitions the space into these regions. You should complete the following figure, which only has the root of the tree. The leaves of the tree should be one of the regions R1, R2, R3, R4, or R5. [4 MARKS]



R1	X
R2	X
R3	O
R4	X
R5	O

Table 1: Fill in with cross (X) or circle (O).

Part (b) What the predictions of the leaves (corresponding to regions R1, R2, ..., R5) should be in order to minimize the classification error over the training set? Fill Table 1 with crosses (X) and circles (O). [3 MARKS]

Please refer to Table 1.

Part (c) Suppose that we constructed a tree with the partitions as in Figure 2. Given the training data points in each region, do you expect it to generalize better or worse compared to the tree in the previous part? Briefly justify your answer. [2 MARKS]

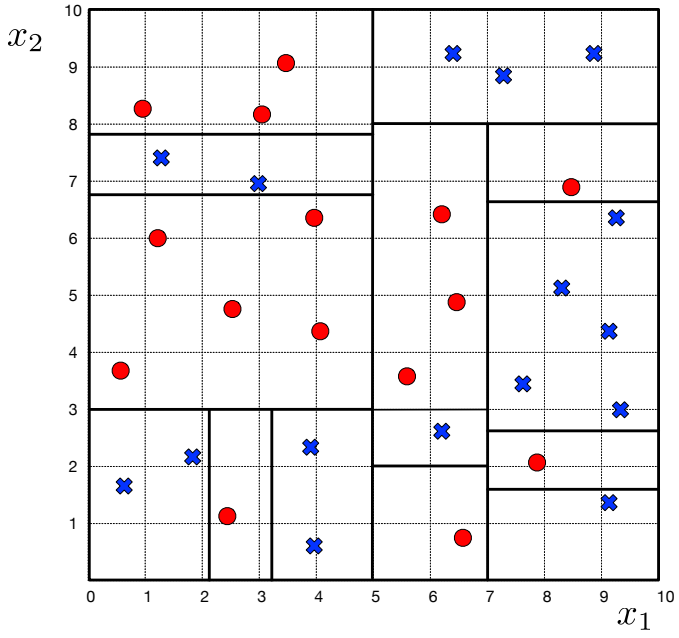


Figure 2: Training points consists of two classes of crosses (X) and circles (O), and fourteen (14) regions partitioning the input space.

The tree resulting from this partitioning might be overly complex (some regions contain just data point) and may have too many regions. This will result in the model overfitting to the intricacies (noise, outliers) of the training set. This can cause poor generalization performance on an unseen test set. A balance must be achieved in depth/complexity of the tree to optimize generalization performance. Pruning & early stopping are used for this purpose. Pruning penalizes the objective function for the # terminal nodes. It aims to find the smallest pruned tree with the lowest penalized error.

Question 6. Multi-Class Classification for Regression Problems [5 MARKS]

Suppose that you have a regression problem with data $(\mathbf{x}, t)$ with the target values t being between $[0, 1]$. We usually solve this type of problems using one of a regression method by minimizing the squared error loss. But assume that you only need to predict t with the resolution of 0.1, e.g., it does not matter whether you predict 0.72 or 0.79; predicting 0.7 would be enough. Briefly describe how this problem can be formulated as a multi-class classification problem.

The problem can be formulated as a classification problem by mapping the target values to discrete bins. Since we are looking for a 0.1 resolution, the bin size should be: $\frac{1-0}{0.1} = 10$

E.g.

$0 \leq y \leq 0.1$	$y' = 0.0$
$0.1 < y \leq 0.2$	$y' = 0.1$
$0.2 < y \leq 0.3$	$y' = 0.2$
⋮	⋮
$0.9 < y \leq 1$	$y' = 0.9$

(Taking the lower limit as the new target value)

This would result in 10 unique (discrete) class labels. This can also be done by creating a histogram of the original continuous target values and setting the new target value to the corresponding bin ID (setting the bin size to 10).

The new target values can optionally be one-hot encoded, depending on which classifier is used, and required input format.

Additionally, since not all classifiers support multi-class classification, we might need to use a One-vs-rest (OVR) strategy that would convert multi-class classification dataset into multiple binary classification dataset and fit a binary classification model on each dataset. Each model would then predict the class membership probability which could then be passed through an argmax function to return the lable with the highest probability.

Metrics like OVR confusion matrices, F1-scores (micro or macro averaging), OVR precision-recall curves, multiclass accuracy can be used to gauge model performance.

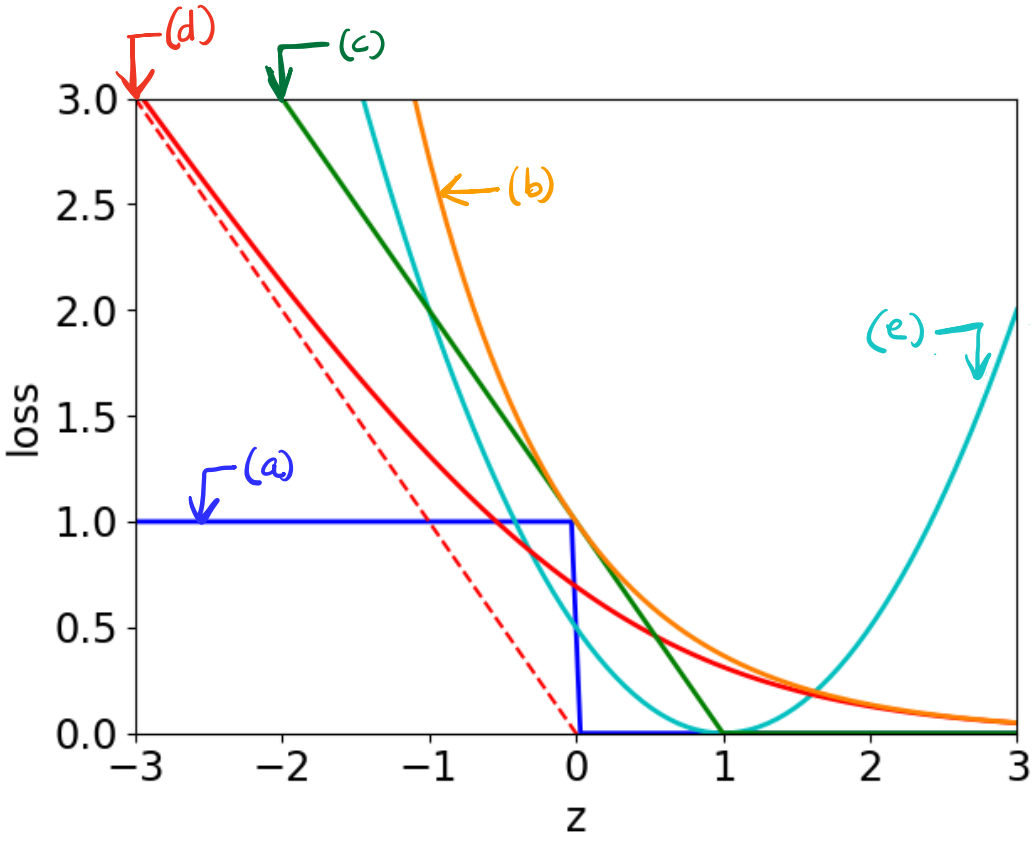
Question 7. Loss Functions for Classification [8 MARKS]

Consider a model whose output is z, e.g., the output of a linear classifier is $z = w^\top x + b$. We have been introduced to a variety of classification loss functions $\mathcal{L}(z, t)$, where t is the target (either $\{0, 1\}$ or $\{-1, +1\}$ for binary classification). This question asks you about these loss functions.

Part (a) Identifying Loss Functions [5 MARKS]

Identify the following loss functions on the figure (or write down their formulate, if you cannot write down on the figure). In this figure, you should assume that the target is $t = 1$. Make sure you identify them clearly without any ambiguity.

- $0 - 1$ Loss (a)
- Exponential Loss (b)
- Hinge Loss (c)
- Cross-Entropy with a Logistic Model (d)
- Squared Error Loss (e)



Part (b) Why don't we minimize the $0-1$ loss function with a linear model? [1 MARK]

The optimization problem can become computationally intractable (NP-hard). This is because 0-1 loss is a non-convex, non-differentiable function. Additionally, $\frac{\partial L_{0-1}}{\partial z} = 0$ everywhere it is defined. (so can't be minimized)

Part (c) What ML method uses the Hinge loss? [1 MARK]

(Soft margin) support vector classifier

Part (d) What ML method can be interpreted as using the exponential loss? [1 MARK]

Ada Boost

Question 8. Optimization of Deep Linear Neural Networks [10 MARKS]

Consider the simplest deep linear neural network that is described by the following equations:

$$z = w_1 x,$$
$$y = w_2 z,$$

with $x, w_1, w_2, y \in \mathbb{R}$. This is indeed a very simple DNN that receives a one dimensional input, has one hidden layer with one unit, and one output. This simplicity is to ensure that the calculations are all easy. Consider the squared error loss function $l(y,t) = \frac{1}{2}(y-t)^2$.

Part (a) Show that one can replace this 2-layer NN with a 1-layer NN (show the relation of the input x to the output y). [2 MARKS]

$y = w_2 z = w_2 (w_1 x)$

$\boxed{y = w_2 w_1 x}$

Part (b) Compute the gradient of the loss of the 2-layer NN with respect to w_1 and w_2. [4 MARKS]

$$\frac{\partial l}{\partial w_1} = \frac{\partial l}{\partial z}\frac{\partial z}{\partial w_1} = \frac{\partial l}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w_1}$$
$$= \frac{\partial}{\partial y}\left(\frac{(y-t)^2}{2}\right)\frac{\partial w_2 z}{\partial z}\frac{\partial w_1 x}{\partial w_1}$$
$$= (y-t)\cdot w_2 \cdot x$$

$$\boxed{\frac{\partial l}{\partial w_1} = w_2 x (y-t)}$$

$$\frac{\partial l}{\partial w_2} = \frac{\partial l}{\partial y}\frac{\partial y}{\partial w_2}$$
$$= \frac{\partial}{\partial y}\left(\frac{(y-t)^2}{2}\right)\frac{\partial w_2 z}{\partial w_2}$$
$$= (y-t)\cdot z$$
$$= (y-t) w_1 x$$

$$\boxed{\frac{\partial l}{\partial w_2} = (y-t) w_1 x}$$

Part (c) Is the loss function of this 2-layer NN convex with respect to w_1 and w_2 or not? Prove your claim. [4 MARKS]

Taking the second derivative of the loss function w.r.t. w_1 and w_2:

From part (b): $\frac{\partial l}{\partial w_1} = w_2 x (y-t)$

$$\frac{\partial^2 l}{\partial w_1^2} = \frac{\partial}{\partial w_1}\left(w_2 x (y-t)\right)$$
$$= \frac{\partial}{\partial w_1}\left(w_2 x (w_2 w_1 x - t)\right)$$
$$= \frac{\partial}{\partial w_1}\left((w_2 x)^2 w_1 - w_2 x t\right)$$
$$\frac{\partial^2 l}{\partial w_1^2} = (w_2 x)^2 \geq 0$$

From part (b): $\frac{\partial l}{\partial w_2} = z\cdot(y-t)$

$$\frac{\partial^2 l}{\partial w_2^2} = \frac{\partial}{\partial w_2}\left(z(y-t)\right)$$
$$= \frac{\partial}{\partial y}\left(z(y-t)\right)\cdot\frac{\partial y}{\partial w_2}$$
$$= z\cdot w_1 x$$
$$= w_1 x \cdot w_1 x = (w_1 x)^2$$
$$\frac{\partial^2 l}{\partial w_2^2} = (w_1 x)^2 \geq 0$$

This shows that the loss is twice differentiable w.r.t. w_1 and w_2 and both $\frac{\partial^2 l}{\partial w_1^2}$ and $\frac{\partial^2 l}{\partial w_2^2}$ are non-negative, and therefore the loss function is $\boxed{\text{convex}}$ w.r.t w_1, w_2

Question 9. Bayesian Classifier [8 MARKS]

Consider a binary classification problem with input x being a scalar. The data generation process works as follows:

- First, a target t is sampled from $\{0, 1\}$ with equal probability.
- If $t = 0$, x is sampled from a uniform distribution over the interval $[0, 1]$.
- If $t = 1$, x is sampled from a uniform distribution over the interval $[0, 2]$.

Part (a) Write down the formula for $P(x|t = 0)$, $P(x|t = 1)$, $P(t = 0)$, and $P(t = 1)$. [4 MARKS]

$P(t=1) = 1 - P(t=0) = 0.5$ [Given t is sampled from $\{0,1\}$ with equal probability, so $P(t=0) = 1 - P(t=1) = 0.5$]

$P(t=0) = 1 - P(t=1) = 0.5$

$$P(x|t=0) = \begin{cases} \frac{1}{1-0} = 1, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

$$P(x|t=1) = \begin{cases} \frac{1}{2-0} = 0.5, & 0 \le x \le 2 \\ 0, & \text{otherwise} \end{cases}$$

Part (b) Compute the posterior probability $P(t = 0|x)$ as a function of x. [4 MARKS]

x is a r.v. in the interval $[0,1]$ i.e. $x \sim [0,1]$

The probability density function over interval $[0,1]$:

$$P(x|t=0) = \begin{cases} \frac{1}{1-0}, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

using Bayes Theorem: $P(t=0|x) = \frac{P(t=0,x)}{P(x)} = \frac{P(x|t=0)P(t=0)}{P(x)}$

$P(x) = P(x|t=0)P(t=0) + P(x|t=1)P(t=1)$
$= (1)(0.5) + (0.5)(0.5) = 0.75$

But since the final answer has to be a function of x,

$P(x \in [0,1]) = \int_0^1 p(x)\,dx$ where $p(x)$ is the cumulative probability density over x.

$$P(t=0|x) = \begin{cases} \frac{0.5}{\int_0^1 p(x)\,dx}, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

or can be written as:

$$P(t=0|x) = \begin{cases} \frac{0.5}{0.75}, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

Also, since $P(x)$ is used in both cases, it might be unnecessary to compute it, as we can compare the ratio instead.

Question 10. Gaussian Discriminant Analysis [4 MARKS]

Consider a GDA model with two classes with covariance matrices Σ_1 and Σ_2.

Part (a) Write down the formula describing the decision boundary (you should be able to find this from the slides). [2 MARKS]

Let the two classes be $K=1$ and $K=2$. The decision boundary between class $K=1$ and class $K=2$:

$\log p(t_{K=1}|x) = \log p(t_{K=2}|x)$. Expanding this, we get:

$$\Rightarrow -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_{K=1}^{-1}| - \frac{1}{2}(x-\mu_{K=1})^T\Sigma_{K=1}^{-1}(x-\mu_{K=1}) + \log p(t_{K=1}) - \log p(x) =$$
$$-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_{K=2}^{-1}| - \frac{1}{2}(x-\mu_{K=2})^T\Sigma_{K=2}^{-1}(x-\mu_{K=2}) + \log p(t_{K=2}) - \log p(x)$$

Part (b) If the covariance matrices are shared between two classes (i.e., $\Sigma_1 = \Sigma_2 = \Sigma$), mathematically show that the decision boundary is linear. [2 MARKS]

If we assume $\Sigma_1 = \Sigma_2 = \Sigma$, we get:

$$\Rightarrow -\cancel{\frac{d}{2}\log(2\pi)} - \cancel{\frac{1}{2}\log|\Sigma_{K=1}^{-1}|} - \frac{1}{2}(x-\mu_{K=1})^T\Sigma_{K=1}^{-1}(x-\mu_{K=1}) + \log p(t_{K=1}) - \cancel{\log p(x)} =$$
$$\cancel{-\frac{d}{2}\log(2\pi)} - \cancel{\frac{1}{2}\log|\Sigma_{K=2}^{-1}|} - \frac{1}{2}(x-\mu_{K=2})^T\Sigma_{K=2}^{-1}(x-\mu_{K=2}) + \log p(t_{K=2}) - \cancel{\log p(x)}$$

Rewriting this:

$$\Rightarrow -\frac{1}{2}(x-\mu_{K=1})^T\Sigma_{K=1}^{-1}(x-\mu_{K=1}) + \log p(t_{K=1}) =$$
$$-\frac{1}{2}(x-\mu_{K=2})^T\Sigma_{K=2}^{-1}(x-\mu_{K=2}) + \log p(t_{K=2})$$

Treating terms without x as constants:

$$\Rightarrow -\frac{1}{2}(x-\mu_{K=1})^T\Sigma^{-1}(x-\mu_{K=1}) + C_{K=1,K=2} = -\frac{1}{2}(x-\mu_{K=2})^T\Sigma^{-1}(x-\mu_{K=2})$$
$$\Rightarrow -\frac{1}{2}\left[x^T\cancel{\Sigma^{-1}}x - 2x\Sigma^{-1}\mu_{K=1}^T + \mu_{K=1}^T\Sigma^{-1}\mu_{K=1}\right] + C_{K=1,K=2} =$$
$$-\frac{1}{2}\left[x^T\cancel{\Sigma^{-1}}x - 2x\Sigma^{-1}\mu_{K=2}^T + \mu_{K=2}^T\Sigma^{-1}\mu_{K=2}\right]$$
$$\Rightarrow x\Sigma^{-1}\mu_{K=1}^T + \frac{1}{2}\mu_{K=1}^T\Sigma^{-1}\mu_{K=1} + C_{K=1,K=2} = x\Sigma^{-1}\mu_{K=2}^T + \frac{1}{2}\mu_{K=2}^T\Sigma^{-1}\mu_{K=2}$$
$$\Rightarrow x\Sigma^{-1}(\mu_{K=1}^T - \mu_{K=2}^T) + C_{K=1,K=2} + \frac{1}{2}\left[\mu_{K=1}^T\Sigma^{-1}\mu_{K=1} - \mu_{K=2}^T\Sigma^{-1}\mu_{K=2}\right] = 0$$

↳ this term is also a constant

$$\Rightarrow \boxed{x\Sigma^{-1}(\mu_{K=1}^T - \mu_{K=2}^T) + C_{K=1,K=2} = 0}$$

Question 11. Estimation Problems in Casino [15 MARKS]

You are at a casino and you decide to play a slot machine. The machine works as follows: On each round of the game, you pull an arm. The machine tells you whether you have won or lost. If you lose, it costs you \$1. If you win, you get a known value \$r, e.g., $r = 5$. You play with the machine until you win, and then it restarts, and then you play another round. In other words, each round consists of playing until a win.

The number of times that you have to pull an arm before winning (and finishing a round) is a random variable K that can take values of $0, 1, 2, \ldots$. For each round, you record this value. For example, if you play three rounds and get LLLLW (round 1), LLW (round 2), and W (round 3), you have $K_1 = 4$, $K_2 = 2$, $K_3 = 0$.

Let us model this process. The probability of winning at each arm pull is θ and the probability of losing is $1 - \theta$, for some unknown $\theta \in [0, 1]$ that depends on the machine. You can assume that the probability of winning at each arm pull is independent from each other and does not change as you play the game. With this assumption, the probability of k losses before winning has the following distribution:

$$P(K = k|\theta) = (1 - \theta)^k \theta, \qquad k = 0, 1, 2, \ldots .$$

As a perfectly rational person, deciding whether or not to play this game should depend on the expected money that you gain.[1] You can calculate the expected gain as follows (you do not need to understand this derivation completely to answer this question): If at one round you suffer k losses before winning, you have lost $k \times \$1$ and you gained $\$r$. This happens with probability $P(K = k|\theta)$. Therefore, your expected gain is

$$\begin{aligned}\text{expected gained money} &= \sum_{k\geq 0}(-k + r)P(K = k|\theta) = r\sum_{k\geq 0} P(K = k|\theta) - \sum_{k\geq 0} kP(K = k|\theta) \\ &= r - \frac{1 - \theta}{\theta}.\end{aligned}$$

So, if $r > \frac{1-\theta}{\theta}$, playing the game is worth it as the expected gain is positive; otherwise, it is not, and you better get out of the casino as soon as possible! Likewise, you can say that if $\theta > \frac{1}{1+r}$, the game is worth playing.

Since you do not know θ, it is crucial to estimate it in order to make an informed decision. This question asks you to develop some estimators for θ.

You play for N rounds and collect a dataset of $\mathcal{D}_N = \{K_1, K_2, \ldots, K_N\}$ describing the result of each rounds. For example, $\{4, 2, 0\}$ are the values for the example above. We assume that this data has already been collected, so you do not have to worry about your gain or loss so far. You can also assume that each round is independent from the previous ones (this is not an extra assumption, as it is implied by the independence of each arm pull).

[1] In reality, we are not rational agents, but that is another story.

Part (a) Likelihood [2 MARKS]

Write the likelihood function $P(\mathcal{D}_N|\theta)$ given a dataset $\mathcal{D}_N = \{K_1, K_2, \ldots, K_N\}$. It should be in the following form:

$$\theta^{\cdots\cdots} \times (1-\theta)^{\cdots\cdots},$$

where $\cdots$ should be completed by you.

$$P(D_N|\theta) = P(K_1, K_2, \ldots, K_N|\theta) = P(K_1|\theta)\,P(K_2|\theta)\ldots P(K_N|\theta)$$
$$= \theta\,(1-\theta)^{K_1} + \theta\,(1-\theta)^{K_2} \ldots + \theta\,(1-\theta)^{K_N}$$
$$= \prod_{i=1}^{N} \theta\,(1-\theta)^{K_i}$$
$$= \theta^{\sum_{i=1}^{N} 1}\,(1-\theta)^{\sum_{i=1}^{N} K_i} = \boxed{\theta^{N}\,(1-\theta)^{\sum_{i=1}^{N} K_i}}$$

Part (b) Log-likelikelihood [1 MARK]

Write the log-likelihood function $\ell(\theta) = \log P(\mathcal{D}_N|\theta)$ given a dataset $\mathcal{D}_N = \{K_1, K_2, \ldots, K_N\}$.

$$L(\theta) = \log P(D_N|\theta) = \boxed{N \log\theta + \sum_{i=1}^{N} K_i \log(1-\theta)}$$

Part (c) MLE [2 MARKS]

Find the Maximum Likelihood Estimator. You need to show your derivations in order to get any mark.

$$\hat{\theta}_{MLE} = \operatorname*{argmax}_{\theta} P(D_N|\theta)$$

$$\frac{d\ell(\theta)}{d\theta} = \frac{d}{d\theta}\left(\theta^{N}\log\theta + \sum_{i=1}^{N} K_i \log(1-\theta)\right)$$

$$= \frac{N}{\theta} - \frac{\sum_{i=1}^{N} K_i}{1-\theta}$$

Setting this to 0 to get MLE:

$$\Rightarrow \quad N - \theta N - \theta \sum_{i=1}^{N} K_i = 0$$
$$\Rightarrow \quad \theta\left(N + \sum_{i=1}^{N} K_i\right) = N$$

$$\boxed{\hat{\theta}_{MLE} = \frac{N}{N + \sum_{i=1}^{N} K_i}}$$

Part (d) Encoding Prior Belief [2 MARKS]

You are skeptical that a casino would setup a game such that you win money. You believe that they set their slot machine such that its θ is small enough to make you lose money in average. You can formulate this belief as a prior distribution on θ. Your skepticism can be expressed by stating that the prior probability that $\theta < \frac{1}{1+r}$ should be high.

As you are already familiar with the Beta distribution, you decide to use it as your prior. Recall that

$$\text{Beta}(\theta; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\,\theta^{a-1}(1-\theta)^{b-1},$$

where Γ is the gamma function. How do you reasonably encode your prior belief with a Beta distribution? You need to specify conditions on a and b. Note that there is no single correct answer. Specify a relation between a and b and briefly justify your answer.

Using the beta distribution as the uninformative prior, where a and b are the underlying parameters of the distribution, ignoring the normalization constant, we get: $\text{Beta}(\theta; a, b) \propto \theta^{a-1}(1-\theta)^{b-1}$. If $r < \frac{1-\theta}{\theta}$ or $\theta < \frac{1}{1+r}$, we know the expected gain is low, and if the game is rigged, then the $P(\theta < \frac{1}{1+r})$ should be high. $\theta < \frac{1}{1+r}$; $E[\theta] < \frac{1}{1+r}$. From the lecture we know, $E_{\text{Beta}}[\theta; a, b] = a/a+b$. So, $\frac{a}{a+b} < \frac{1}{1+r}$, we get $\frac{a}{a+b} < \frac{1}{1+r} \Rightarrow a < \frac{a+b}{1+r} \Rightarrow a + ar - a < b \Rightarrow \boxed{ar < b}$

a and b are set according to prior beliefs about θ, and by varying these, we can encode a wide range of possible beliefs.

Part (e) MAP [2 MARKS]

Assume that you have selected a Beta distribution with a particular choice of a and b to encode your prior belief. Find the Maximum A-Posteriori (MAP) estimate. Your answer should be a function of $\mathcal{D} = \{K_1, K_2, \ldots, K_N\}$ and a and b. You should not use the particular a and b that you found in the previous question, but write it for any a and b.

$$P(\theta|D) \propto P(\theta)P(D_N|\theta)$$
$$\propto \theta^{a-1}(1-\theta)^{b-1}\,\theta^{\sum_{i=1}^N K_i}(1-\theta)^N$$
$$\propto \left[\theta^{a-1+N}\right]\left[(1-\theta)^{b-1+\sum_{i=1}^N K_i}\right]$$

where $N + a = a'$

$\sum_{i=1}^N + b = b'$

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmax}}\; P(\theta|D)$$
$$= \underset{\theta}{\text{argmax}}\; P(\theta, D)$$
$$= \underset{\theta}{\text{argmax}}\; P(\theta)\,P(D|\theta)$$
$$= \underset{\theta}{\text{argmax}}\; \log P(\theta) + \log P(D|\theta)$$

$$\log P(\theta|D) = \log P(\theta) + \log P(D_N|\theta)$$
$$= \text{constant} + (a-1)\log\theta + (b-1)\log(1-\theta) + N\log\theta + \textstyle\sum_{i=1}^N K_i \log(1-\theta)$$
$$= \text{Constant} + (a-1+N)\log\theta + (b-1+\textstyle\sum_{i=1}^N K_i)\log(1-\theta)$$

Finding the critical point:

$$\frac{d}{d\theta}\left(\log P(\theta|D)\right) = \frac{N+a-1}{\theta} - \frac{\sum_{i=1}^N K_i + b - 1}{1-\theta} = 0$$

$$(N+a-1)(1-\theta) - \theta\left(\textstyle\sum_{i=1}^N K_i + b - 1\right) = 0$$
$$N + a - 1 - N\theta - a\theta + \theta - \theta\textstyle\sum_{i=1}^N K_i - b\theta + \theta = 0$$
$$\theta\left(-N - a + 1 - \textstyle\sum_{i=1}^N K_i - b + 1\right) = -(N + a - 1)$$

$$\boxed{\hat{\theta}_{MAP} = \frac{N+a-1}{N+a-2+\sum_{i=1}^N K_i + b}}$$

Part (f) Bayesian Posterior [2 MARKS]

Calculate the Posterior probability of θ given the prior Beta$(\theta; a, b)$ and the data $\mathcal{D}_N = \{K_1, K_2, \ldots, K_N\}$. (Hint: Notice that the Beta distribution is a conjugate prior for this likelihood, so your posterior is in the form of a Beta distribution too.)

The posterior distribution would be: (since we ignore normalization constant, we use proportionality)

$$P(\theta|D) \propto P(\theta)P(D_N|\theta)$$
$$\propto \theta^{a-1}(1-\theta)^{b-1}\,\theta^{N}(1-\theta)^{\sum_{i=1}^{N} K_i}$$
$$\propto \theta^{a-1+N}(1-\theta)^{b-1+\sum_{i=1}^{N} K_i}$$

, where $\underbrace{N+a}_{a'}$ and $\underbrace{\sum_{i=1}^{N} K_i + b}_{b'}$ are the parameters of the new beta distribution

$$\boxed{P(\theta|D) = \theta^{a-1+N}(1-\theta)^{b-1+\sum_{i=1}^{N} K_i}}$$

Part (g) Bayesian Estimation of ~~[1]~~ [1 MARK] $E[\theta|D]$ (Piazza @ 222)

What is the expected value of the parameter θ according to the posterior distribution?

$$P(\theta|D) \propto \theta^{a-1+N}(1-\theta)^{b-1+\sum_{i=1}^{N} K_i}, \text{ where } \begin{matrix} N+a = a' \\ \sum_{i=1}^{N} K_i + b = b' \end{matrix} \Big\} \text{ Parameters of the new beta distribution}$$

$$\boxed{E[\theta|D] = \frac{a'}{a'+b'} = \frac{N+a}{N+a+\sum_{i=1}^{N} K_i + b}}$$

Part (h) Comparison of MLE, MAP, and Bayesian Estimation [3 MARKS]

Briefly explain what the advantages and disadvantages of each of MLE, MAP, and Bayesian Estimation are (you should be able to answer this even if you have not calculated the estimators correctly.)

	ADVANTAGES	DISADVANTAGES
MLE	• Gives the θ that maximizes the likelihood. • MLE is a special case of MAP where prior is uniform. • can be used to derive ML algorithms, like regression with squared loss (MLE with Gaussian noise)	• Assumes data observations are independently and identically distributed (this assumption may not hold true in many real-world settings) • Returns a single point estimate which might be sub-optimal in some cases (discards important info.) • Performs poorly on small datasets.
MAP	• Gives the θ that maximizes the posterior probability. This is an approx. of the full Bayesian estimation and inference. • Incorporates information from a prior distribution for model of X.	• Returns a single point estimate which again might be sub-optimal in some cases as important information of the probability distribution is discarded. • without prior $P(\theta)$, MAP = MLE equation. • would assign zero probabilities if $a, b \leq 1$ • Since its dependent on prior, may give biased prediction.
Bayesian	• returns the probability density function instead of a single point estimate. • works well with a small dataset • Incorporates prior information into calculations. With new info, previous posterior can be used as prior.	• Need to mathematically represent subjective prior beliefs. Additionally, it produces posterior distributions heavily influenced by priors, so depending on the situation this might lead to biased results.

CONT'D...

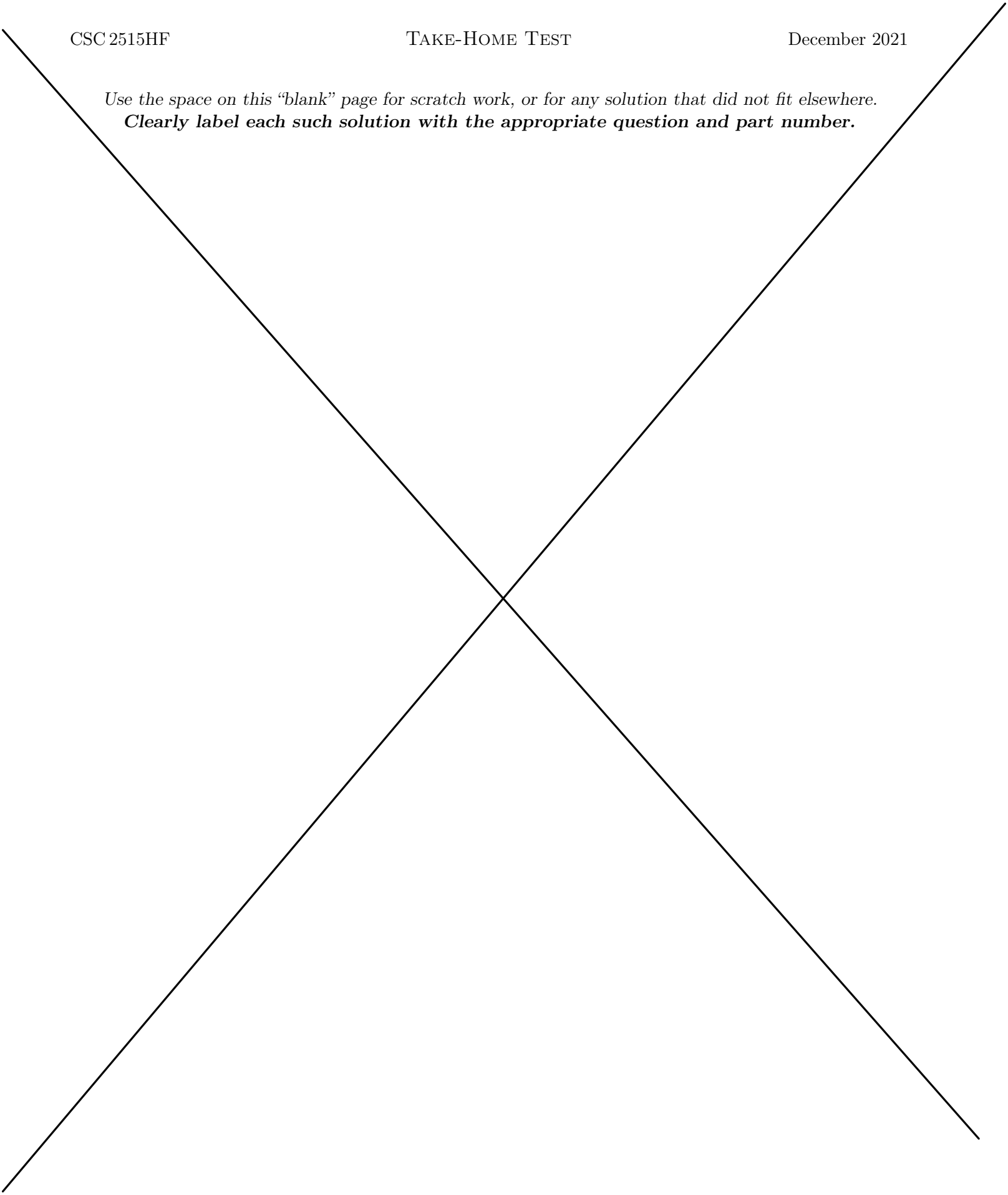
Question 12. Course Evaluation [2 MARKS]

Please fill the course evaluation! I will read all your comments. It helps me improve the course in the future. Also feel free to send me an email if you want to provide more detailed comments.

If you did, please specify it here. You get the point if you have filled it.

I've submitted the course evaluation. Thank you for the bonus! ☺

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.



Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.