Name: Victoria Yong	Student ID: 1004455

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

50.007 Machine Learning, Fall 2021
Midterm Exam

Date: 5 November 2021
Time: 17:00 - 20:00

Instructions:

1. This is an open-book exam.
2. Write your name and student ID at the top of this page.
3. The problems are not necessarily in order of difficulty. We recommend that you scan through all the questions first, and then decide on the order to answer them.
4. Problem 1 and Problem 2 were already provided through the link.
5. Write your answers in the space provided.
6. You may access the Internet.
7. **You may NOT communicate via any means with anyone.**

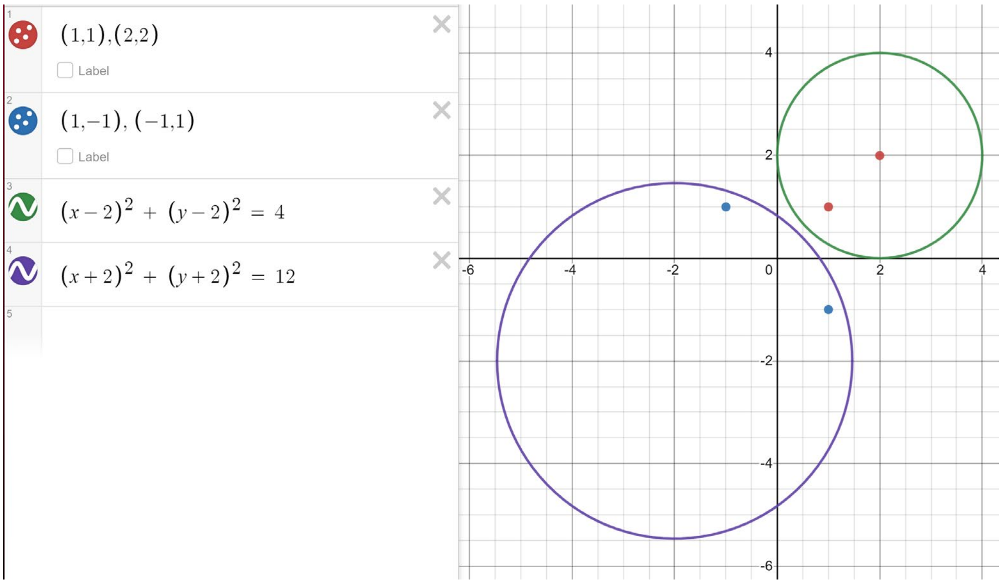
For staff's use:

Problem 1	/7
Problem 2	/12
Problem 3	/4
Problem 4	/6
Problem 5	/5
Problem 6	/6
Problem 7	/5
Problem 8	/5
Total	**/50**

Problem 3: Classification (4 Points)

(a) Consider data points from a 2-d space where each point is of the form $x = (x_1, x_2)$. You are given the same dataset as in homework 1, problem 1, with two positive examples: (1, 1) and (2, 2), and two negative examples (-1, 1) and (1, -1). For the hypothesis space, inside or outside of a (a, b)-centered circle with radius r, find the parameters (where, a, b, r are the parameters) of the classifier (a member of the hypothesis space) that can correctly classify all the examples in the dataset, or explain why no such classifier exists. If a classifier is possible then graphically represent all the examples of the dataset along with the classifier parameters. *(2 points)*

A classifier exists.
The decision boundary is given as $(x-a)^2 + (y-b)^2 = r^2$, where
a > 1 or a < -1
b > 1 or b < -1
r2 = max distance from (a, b) to either all positive or all negative points



(b) We know that *"The perceptron update rule converges after a finite number of mistakes when the training examples are linearly separable through origin."* Consider a training example $x^{(t)}$, from a dataset that is linearly separable through the origin, which has been initially misclassified. Prove that the perceptron update rule ($\theta^{(k+1)} = \theta^{(k)} + y^{(t)}x^{(t)}$) does indeed attempt to correctly classify the training example by increasing the value of $y^{(t)}(\theta \cdot x^{(t)})$. *(2 points)*

Under the perceptron update rule, to check if $x^{(t)}$ is misclassified, the algorithm checks if $y^{(t)}(\vartheta \cdot x^{(t)}) <= 0$. This checks the sign of the condition, where the point is considered misclassified if the condition returns a negative value (i.e. the sign of the prediction and label are not the same). Since $x^{(t)}$ is initially misclassified, $y^{(t)}(\vartheta \cdot x^{(t)})$ returns a negative value.

The algorithm will then increase or decrease the weights according to $\vartheta^{(k+1)} = \vartheta^{(k)} + y^{(t)}x^{(t)}$. Upon making a second prediction, $\vartheta \cdot x^{(t)}$, on the data, the updated weight will then pull the prediction in the opposite direction towards the alternative class, thus attempting to re-classify it correctly.

Problem 4: Regression (6 Points)

A candy factory is preparing for Christmas and they want to optimize their production pricing. They currently produce 3 types of candy: Apple, Banana and Chocolate. They have data from 10 previous years regarding their production price, the price at which they sold the candies, and the quantities that were bought. They also have data about the total number of candies of these 3 types that were on the market in the past 2 years, and they know the retail price charged for competitor candies in the last 10 years (but not the quantities that were sold). They want you to help them predict the amount of candy of each category that they will be able to sell, as a function of all the data available (production price, selling price, retail price of competitors and total number of candies in the market). For each of the following algorithm, explain how it works, how would the algorithm perform and any drawbacks. Also, if the algorithm can be improved by any pre-processing of the data, please state so.

1. Linear Regression *(2 points)*

   The Linear Regression algorithm attempts to model the data with a straight line. It iteratively calculates the error (most commonly mean squared error) of points from its predicted line and tries to minimize the error with each prediction of a new line to separate the data. The target of the problem is to predict the prospective quantity of sales for each category of candy. This is a multiple linear regression problem, since there are multiple features in the data, and multiple dependents. I would expect this algorithm to perform relatively poorly with high bias, since retail and economic data in general do not follow linear trends.

   The algorithm may be improved by normalizing the data.

2. Polynomial Regression *(2 points)*

   Polynomial regression is a regression algorithm that attempts to model the relationship between the x and y variables as an nth degree polynomial. Polynomial regression also commonly uses mean squared error as its loss function, and similarly attempts to minimize the mean squared error of points from the polynomial curve prediction. I would expect this model to take the longest time to compute, but able to generalize the data better than linear regression. However, using this model may be more prone to overfitting the data compared to the linear regression.

3. Ridge Regression *(2 points)*

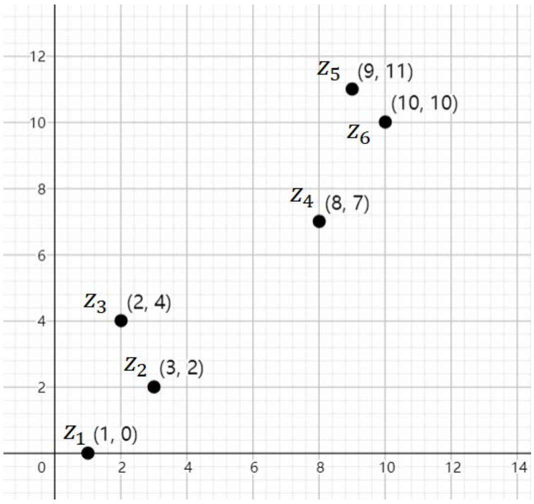
   Ridge regression is a method of estimating the coefficients for multiple regression models. Essentially, it is a form of regularized linear regression and is useful for data where all the variables are highly correlated. I would expect this method to generalize the best out of all 3 methods. Similarly, the algorithm may be improved by normalizing the features.

Problem 5: K-Means (5 Points)

Suppose we want to use k-means algorithm to perform clustering on the dataset

$$S = \{z_1, z_2, z_3, z_4, z_5, z_6\}$$

which is also given below:



In the initialization step, z_1 is used as the first cluster center C_1 and z_2 as the second cluster center C_2. k is set to 2.

a) Suppose we simulate k-means algorithm for **ONE iteration**. We first calculate the Euclidean distance between cluster centers and each point in the dataset S, we then perform cluster assignment. Please fill your results in the table given below for one iteration. *(2 points)*

Data		Distance to		Cluster Assignment
i	z_i	$C_1 = (1,0)$	$C_2 = (3,2)$	
1	(1,0)	0	2.828	C1
2	(3,2)	2.828	0	C2
3	(2,4)	4.123	2.236	C2
4	(8,7)	9.899	7.071	C2
5	(9,11)	13.601	10.817	C2
6	(10,10)	13.454	10.630	C2

b) How many iterations we need for **convergence**? Please list the **final** cluster assignments. *(2 points) Please note that you don't need to list all the iterations.*

2 Iterations.

The mean of C1 will be (1, 0) and C2 will be (6.4, 6.8)

The final assignments are

Data		Distance to		Cluster Assignment
i	z_i	C_1 = (1,0)	C_2 = (6.4, 6.8)	
1	(1,0)	0	8.683	C1
2	(3,2)	2.828	5.882	C1
3	(2,4)	4.123	5.215	C1
4	(8,7)	9.899	1.612	C2
5	(9,11)	13.601	4.940	C2
6	(10,10)	13.454	4.817	C2

c) Suppose we want to cluster a new data point $z_7 = (11, 8)$ after the k-means algorithm is converged. Which cluster is z_7 assigned to? *(1 point)*

C2

Problem 6: SVM (6 points)

1. In the Dual formulation of with SVM soft margin, why is Lagrange multiplier a upper bounded by C (in lecture slides)? (1 points)

   In formulating the dual formula without soft margin, the langrange multiplier is constrained only by a lower bound, where $0 \leq \alpha$. The addition of a soft margin means that the langrange multiplier must allow the model to 'ignore' terms which satisfy the constraints given by the slack term. Hence there is an upper bound to allow the points which satisfy the slack constraint to pass.

2. What's the "kernel trick" in SVM and how is it useful? Please provide an example (a particular task) in which we need to use "kernel trick". (1 points)

   The kernel trick is employed on data that is not linearly separable and involves transforming the data into a higher feature space via a kernel function, making it linearly separable in that dimension.

Indicate whether the following functions are valid SVM kernels, and explain your answers (i.e. provide a formal proof to justify your answer). *Hint: Please remember the properties of kernel functions that we discussed in the lectures.*

3. $K(x, x') = 16$ *(1 points)*

Is this a kernel? Yes Explanations: According to the properties of kernel functions, K(x, x') = 1 is a kernel function, and any sum of kernel functions is also a kernel function. The given kernel is the sum of [K(x, x') = 1] 16 times.

4. $K(x, x') = (x \cdot x' + 9)$ *(1 points)*

Is this a kernel? No Explanations: The given kernel does not meet any of the 4 properties of kernel functions.

5. $K(x, x') = (x \cdot x')^2 - 8$ *(1 points)*

Is this a kernel?No Explanations: According to the properties, -8 cannot be a kernel.

6. $K(x, x') = (x \cdot x)^4 + (x' \cdot x')^2$ *(1 points)*

Is this a kernel? Yes Explanations: According to the properties, ~K(x; x0) = f(x)K(x; x0)f(x0) is a kernel.

Problem 7: Logistic Regression (5 points)

In our lectures, we have studied how to formulate the decision boundary of a classifier based on logistic regression. Please answer the following questions qualitatively.

(a) Please explain a scenario you need to use logistic regression to solve a particular problem. *(2 points)*

Clearly state the problem, the data that you have (labeled/unlabeled), input-output to your logistic regression, and why you would prefer using logistic regression over SVM. Please do not use more than 6-7 sentences.

Problem: Classifying whether an image of a person's face is conventionally attractive or not attractive

Training Data: Labeled portrait images, possibly from social media, including metrics to model conventional attractiveness like number of likes on the photo

Input: Image of a face Output: Target class and the array of confidence levels

Reason: With Logistic regression, we can predict the image into binary classes, but also be able to obtain valuable information such as degree of attractiveness within the class. (e.g. Person who has 100% of the features of conventional beauty standards 100% vs a person who has 70%, but both will classified as attractive)

(b) Suppose you trained a logistic classifier with a hypothesis function as given below:

$$h_\theta(x) = g(\theta_0 + x_1\theta_1 + x_2\theta_2 + x_1^2\theta_3 + x_2^2\theta_4)$$

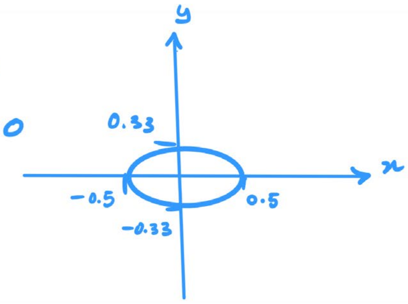
Then, you obtained $\theta = [-1\ 0\ 0\ \ 4\ \ 9]^T$ using gradient descent algorithm. Please **formulate and draw** the decision boundary of your classifier. Note that this is a binary classification problem, which means class label y can be 0 or 1. *(2 points)*

given $\theta = [-1\ 0\ 0\ 4\ 9]^T$,

The decision boundary is: $4x_1^2 + 9x_2^2 - 1 = 0$

$y = 1$ if $4x_1^2 + 9x_2^2 - 1 \geq 0$

$y = 0$ if $4x_1^2 + 9x_2^2 - 1 < 0$

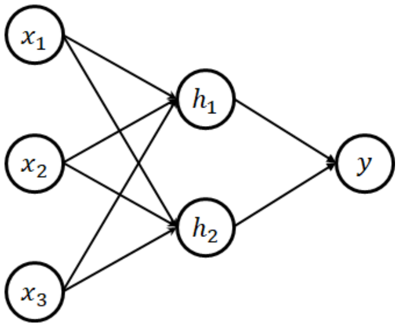


(c) After training completed, if you test your logistic regression with a new data point, would your decision boundary change? Please explain your answer. *(1 points)*

The decision boundary would not change since the weights are not updated when the model is tested.

Problem 8: Neural Networks (5 points)

As can be seen below, we have a simple neural network, which includes an input layer, a single hidden layer and an output. Please note that the parameters of the network (weight matrices) are already estimated. The weight matrix W that connects input to the hidden layer is $\begin{bmatrix} -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$. The weight matrix V that connects the hidden layer to the output is $\begin{bmatrix} 1 & 2 \end{bmatrix}$. ReLu function is used as the activation function for the hidden and the output layer.



Let's assume that you are given the following input data:

$$x = (x_1, x_2, x_3) = (2, 1, 2)$$

What is the output value of this neural network given this input? Please also report the output of h_1 and h_2. Explain your answer, and provide step-by-step formulation.

Important note: 1) Please ignore the bias term, and 2) during the lecture (week 6), we have studied the mathematical formulation of a neural network.

for h_1:

Inputs: $x_1 : -2$ $x_2 : 2$ $x_3 : -2$

output: 2

for h_2:

Inputs: $x_1 : 0$ $x_2 : -1$ $x_3 : -2$

output: 0

Final output = 2