CS M146 Midterm

Vincent Cheung

TOTAL POINTS

77.5 / 100

QUESTION 1

1 True/false 15 / 18

- **0 pts** Correct
- ✓ **- 3 pts (a) incorrect (e.g., saying p(x) is probability)**
- **3 pts** (b) incorrect
- **3 pts** (c) incorrect
- **3 pts** (d) incorrect
- **3 pts** (e) incorrect
- **3 pts** (f) incorrect
- **2 pts** (a) partial points for showing how to use integral to get probability from p(x) and arguing 0 \leq integral p(x) \leq 1 (but we are asking p(x), not integral p(x))

QUESTION 2

Short Question 23 pts

2.1 (a)-(d) 13 / 13

- ✓ **- 0 pts Correct**
- **4 pts** (a) incorrect
- **3 pts** (b) incorrect
- **3 pts** (c) incorrect
- **3 pts** (d) incorrect
- **2 pts** (a) partially correct
- **1.5 pts** (b) partially correct
- **1.5 pts** (c) partially correct
- **1.5 pts** (d) partially correct
- **0 pts** (b) should specify tuning "hyper-parameter"

2.2 (e) 0 / 10

- **0 pts** Correct
- **1 pts** Answer correct but missed one/two steps while proving
- **2 pts** Some minor mistakes/missed a important step
- **5 pts** Major mistakes, but mentioned some important points like solving a linear system Xw. E.g., trying to solve Xw = 0 instead of Xw = y or mention X is invertible
- **8 pts** only mentioned definition of linear independence
- ✓ **- 10 pts incorrect**

QUESTION 3

Decision tree 15 pts

3.1 (a) i, ii 7 / 7

- ✓ **- 0 pts Correct**
- **2 pts** a) i. incorrect
- **0.5 pts** a) i. partially incorrect
- **5 pts** a) ii. incorrect
- **2.5 pts** a) ii. partially incorrect

3.2 (a) iii 0 / 3

- **0 pts** Correct
- **1.5 pts** a) iii. Partially incorrect
- ✓ **- 3 pts a) iii) incorrect**

3.3 (b) 2.5 / 5

- **0 pts** Correct
- ✓ **- 2.5 pts partially incorrect**
- **5 pts** incorrect

QUESTION 4

Perceptron 23 pts

4.1 (a) (answer 2,4,5,6; 4,5,6; 2,4,6; 4,6; are all okay) 4 / 4

- **4 pts** Totally wrong
- **2 pts** Partially Correct
- ✓ **- 0 pts Correct**

4.2 (b) 4 / 8

- ✓ **- 4 pts did mention yx or mention learning rate, but got totally wrong with the constraint of the learning rate**
- **0 pts** correct

- **2 pts** made tiny mistakes on the constraint of the learning rate
- **8 pts** did not mention yx or learning rate (yx is the basic and necessary component when updating the weights)

4.3 (c),(d) 6 / 6

- **3 pts** c is wrong
- **3 pts** d is wrong
- **6 pts** both c and d are wrong

✓ **- 0 pts** all correct

- **1 pts** c is partially correct: mention "adding dimension" without specific solutions or with wrong solutions
- **1 pts** d is partially correct: A. wrong w0w1w2 B.neglect the question "only solution"

4.4 (e) 5 / 5

- **2 pts** partially correct, e.g. draw a correct diagram

✓ **- 0 pts** correct

- **5 pts** wrong

QUESTION 5

19 pts

5.1 (a) 3 / 3

✓ **- 0 pts** Correct

- **1 pts** No Y prediction
- **1 pts** Incorrect Prediction
- **1.5 pts** Wrong calculation & not finished; no Y prediction
- **1.5 pts** Incomplete & wrong calculation
- **0.5 pts** Wrong calculation
- **0.5 pts** No Y prediction after calculating probabilities
- **1.5 pts** Wrong calculation & wrong prediction
- **1 pts** Wrong formula is used
- **0 pts** Slight mistake in calculation
- **1.5 pts** Not finished; no Y prediction
- **1 pts** Your calculation is wrong & how you get Y? See solution
- **0.5 pts** You need to show how you get Y
- **1 pts** Wrong calculation & prediction is wrong
- **3 pts** No answer
- **2 pts** Unfinished

5.2 (b) 6 / 6

✓ **- 0 pts** Correct

- **2 pts** But you need to prove it.
- **1 pts** You need to show that the other form of this classifier is w^Tx= 0
- **6 pts** Wrong answer
- **0.5 pts** See the solution in CCLE
- **1 pts** See the solution in CCLE
- **2 pts** Your proof is not correct
- **3 pts** Wrong perception ; see the solution on CCLE
- **2 pts** I did not understand what have you written. Assuming you have written 'linear classifier' I have graded. You need to prove it. Please the the solution on CCLE

5.3 (c) 10 / 10

✓ **- 0 pts** Correct

- **0 pts** You forgot to mention the sum
- **2 pts** Please see the solution on CCLE
- **10 pts** No answer
- **5 pts** Unfinished
- **8 pts** Wrong answer
- **0 pts** Slight mistake
- **2 pts** How??
- **9 pts** No answer
- **8 pts** Not finished
- **0 pts** Mistake
- **3 pts** Please see the solution on CCLE
- **5 pts** Not correct.

QUESTION 6

6 name 2 / 2

✓ **- 0 pts** Correct

CM146: Introduction to Machine Learning	Fall 2018

Midterm

Nov. 5^{th}, 2018

- This is a closed book exam. Everything you need in order to solve the problems is supplied in the body of this exam.
- This exam booklet contains **Five** problems.
- You have 90 minutes to earn a total of 100 points.
- **Besides having the correct answer, being concise and clear is very important. For full credit, you must show your work and explain your answers.**

Good Luck!

Name and ID: (2 Point) Vincent Cheung 404 751 330

Name		/2
True/False Questions		/18
Short Questions		/23
Decision Tree		/15
Perceptron		/23
Regression		/19
Total		/100

1 True/False Questions (Add a 1 sentence justification.) [18 pts]

(a) **(3 pts)** For a continuous random variable x and its probability density function $p(x)$, it holds that $0 \le p(x) \le 1$ for all x.

True. The total probabilities must sum to 1, so each individual probability density must be between 0 and 1.

(b) **(3 pts)** K-NN is a linear classification model.

False. K-NN can classify data that is not linearly separable, like XOR.

(c) **(3 pts)** Logistic regression is a probabilistic model and we use the maximum likelihood principle to learn the model parameters.

True. Logistic regression outputs a probability between 0 and 1 using the sigmoid of $w^T x_i$, where w^T maximizes the likelihood function.

(d) **(3 pts)** Suppose you are given a dataset with 990 cancer-free images and 10 images from cancer patients. If you train a classifier which achieves 98% accuracy on this dataset, it is a reasonably good classifier.

False. A high training accuracy can mean overfitting, which could cause low testing accuracy.

(e) **(3 pts)** A classifier that attains 100% accuracy on the training set is always better than a classifier that attains 70% accuracy on the training set.

False. A high training accuracy can mean overfitting, which could cause low testing accuracy.

(f) **(3 pts)** A decision tree is learned by minimizing information gain.

False. It is learned by minimizing entropy, or maximizing information gain.

2 Short Questions [23 pts]

(a) **(4 pts)** What is the main difference between gradient descent and stochastic gradient descent (in one sentence)? Which one require more iterations to converge, why?

GD. computes the error of every data point before updating the weight vector, whereas stochastic GD. uses just one data point at a time. The stochastic GD. takes more iterations to converge since we are updating the weight vector only using one data point each time.

(b) **(3 pts)** What is the motivation to have a development set?

The development set is used to tune the hyperparameters to find the best ones (the ones with the least error on the development set).

(c) **(3 pts)** Describe the differences between linear regression and logistic regression (in less than two sentences).

Linear regression creates a linear function that outputs a continuous number in $\mathbb{R}$ that predicts a value, whereas logistic regression creates a non-linear function (eg sigmoid) that outputs a probability for a classification.

(d) **(3 pts)** Consider the models that we have discussed in lecture: decision trees, k-NN, logistic regression, Perceptrons. If you are required to train a model that predicts the probability that the patient has cancer, which of these would you prefer, and why?

Logistic regression, since it uses a non-linear function to output a probability for a classification.

(e) **(10 pts)** Given n linearly independent feature vectors in n dimensions, show that for any assignment to the binary labels you can always construct a linear classifier with weight vector w which separates the points. Assume that the classifier has the form $sign(w \cdot x)$. Hint: a set of vectors are linearly independent if no vector in the set can be defined as a linear combination of the others.

Let $x_1, \ldots, x_n \in \mathbb{R}^n$ be the n feature vectors, $w \in \mathbb{R}^n$ be the weight. Since $x_1, \ldots, x_n$ are linearly independent, they must have $\gamma > 0$ and since n is finite, we have the max length R as finite. Using perceptron, we can separate the vectors after R^2/γ^2 mistakes. Since we may have less than R^2/γ^2 vectors, they are definitely able to be separated.

3 Decision Trees [15 pts]

For this problem, you can write your answers using $\log_2$, but it may be helpful to note that $\log_2 3 \approx 1.6$ and entropy $H(S) = -\sum_{v=1}^{K} P(S=v)\log_2 P(S=v)$. The information gain of an attribute A is $G(S,A) = H(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} H(S_v)$, where S_v is the subset of S for which A has value v.

(a) We will use the dataset below to learn a decision tree which predicts the output Y, given by the binary values of A, B, C.

A	B	C	Y
F	F	F	F
T	F	T	T
T	T	~~T~~ F	T
T	T	T	F

i. **(2 pts)** Calculate the entropy of the label y.

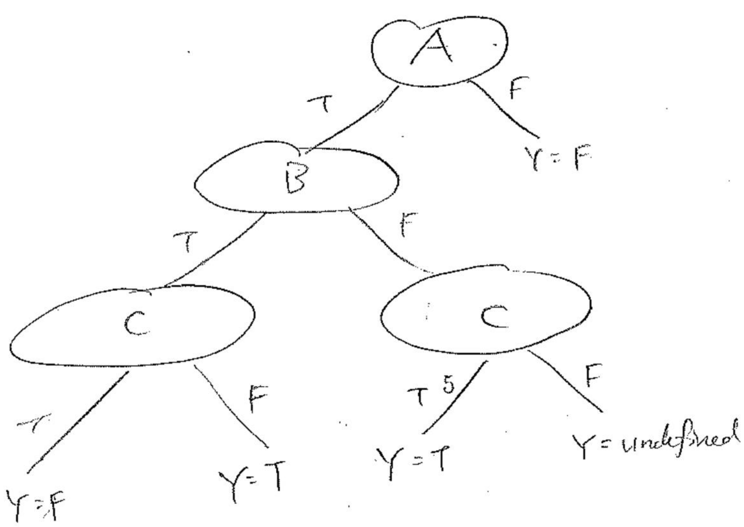
$$H(y) = -\tfrac{1}{2}\log\tfrac{1}{2} - \tfrac{1}{2}\log\tfrac{1}{2} = \tfrac{1}{2}\log 2 + \tfrac{1}{2}\log 2 = 1$$

ii. **(5 pts)** Draw the decision tree that will be learned using the ID3 algorithm that achieves zero training error.

$$G(y,A) = H(y) - \tfrac{1}{4}(0) - \tfrac{3}{4}\left(-\tfrac{2}{3}\log\tfrac{2}{3} - \tfrac{1}{3}\log\tfrac{1}{3}\right) = 1 + \tfrac{1}{2}\log\tfrac{2}{3} + \tfrac{1}{4}\log\tfrac{1}{3}$$

$$G(y,B) = H(y) - \tfrac{1}{2}\left(-\tfrac{1}{2}\log\tfrac{1}{2} - \tfrac{1}{2}\log\tfrac{1}{2}\right) - \tfrac{1}{2}\left(-\tfrac{1}{2}\log\tfrac{1}{2} - \tfrac{1}{2}\log\tfrac{1}{2}\right) = 1 - \tfrac{1}{2} - \tfrac{1}{2} = 0$$

$$G(y,C) = G(y,B) = 0.$$



iii. **(3 pts)** Is this tree optimal (i.e. does it get minimal training error with minimal depth?) explain in two sentences, and if it isn't optimal draw the optimal tree.

It is optimal because ID3 produces the optimal tree based on information gain. Here, the training error is 0 and depth is 3, both of which are minimal.

(b) **(5 pts)** You have a dataset of 400 positive examples and 400 negative examples. Now suppose you have two possible splits. One split results in (300+, 100-) and (100+, 300-). The other choice results in (200+, 400-), and (200+, 0). Which split is most preferable and why?

The first split (300+, 100-) and (100+, 300-).

We can use the (300+, 100-) part to train and the (100+, 300-) part to test both positive and negative examples.

With the other split, the (200+, 0) part will only be able to train or test with positive examples.

4 Perceptron Algorithm [23 pts]

(a) **(4 pts)** Assume that you are given training data $(x, y) \in R^2 \times \{\pm 1\}$ in the following order:

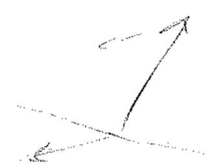
Instance	1	2	3	4	5	6	7	8
Label y	+1	−1	+1	−1	+1	−1	+1	+1
Data (x_1, x_2)	(10, 10)	(0, 0)	(8, 4)	(3, 3)	(4, 8)	(0.5, 0.5)	(4, 3)	(2, 5)
w	(1, 1)	(1, 1)	(1, 1)	(-2, -2)	(2, 6)	(1.5, 5.5)	(1.5, 5.5)	(1.5, 5.5)

We run the Perceptron algorithm on all the samples once, starting with an initial set of weights $w = (1, 1)$ and bias $b = 0$. On which examples, the model makes an update?

2, 4, 5, 6

(b) **(8 pts)** Suggest a variation of the Perceptron update rule which has the following property: If the algorithm sees two consecutive occurrences of the same example, it will never make a mistake on the second occurrence. (Hint: determine an appropriate learning rate that guarantees this property). Prove your answer is correct.

The update rule is :



$w \leftarrow w + \underline{\; r\, y_i x_i \;}$

learning rate

Let $r = |w|$. Then $w \leftarrow w - w = 0$.

(c) **(3 pts)** Linear separability is a pre-requisite for the Perceptron algorithm. In practice, data is almost always inseparable, such as XOR.

x_1	x_2	y
-1	-1	-1
-1	$+1$	$+1$
$+1$	-1	$+1$
$+1$	$+1$	-1

Provide a solution to convert the inseparable data to be linearly separable. The XOR can be used for the illustration.

Use the product $(-x_1 x_2)$ $\Longrightarrow$

$-x_1x_2$	y
-1	-1
$+1$	$+1$

now it is linearly separable

if $y_i(-x_1^{(i)} x_2^{(i)}) < 0$:

update $w \leftarrow w + y_i(-x_1^{(i)} x_2^{(i)})$;

(d) **(3 pts)** Design (specify w_0, w_1, w_2 for) a two-input Perceptron (with an additional bias or offset term) that computes "OR" Boolean functions. Is your answer the only solution?

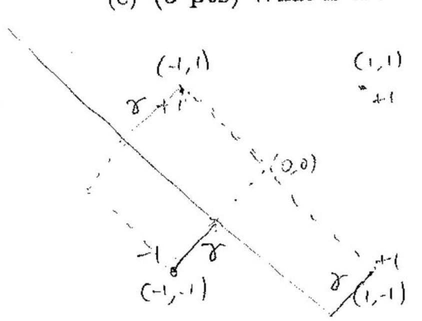
x_1	x_2	y
-1	-1	-1
1	-1	1
1	1	1
-1	1	1

Use $(w_2, w_1, w_0) = (1, 1, \frac{1}{2})$. This is not the only solution.

$\uparrow$ bias.

There are infinitely many solutions; e.g. $(k, k, \frac{1}{2}k)$ for $k > 0$.

(e) **(5 pts)** What is the maximal margin γ in the above OR dataset.



$\Rightarrow$ Maximal margin $\gamma = \frac{1}{2}$ distance $[(0,0), (-1,-1)] = \boxed{\frac{\sqrt{2}}{2}}$

5 Logistic Regression[19 pts]

Considering the following model of logistic regression for a binary classification, with a sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$:

$$P(Y = 1|X, w_0, w_1, w_2) = \sigma(w_0 + w_1 X_1 + w_2 X_2)$$

(a) **(3 pts)** Suppose we have learned that for the logistic regression model, $(w_0, w_1, w_2) = (-\ln(4), \ln(2), -\ln(3))$. What will be the prediction ($y = 1$ or $y = -1$) for the given $x = (1, 2)$?

$$\sigma(-\ln 4 + 1 \cdot \ln 2 - 2\ln 3) = \sigma(-2 + 1 - 2\ln 3)$$

$$= \sigma(-1 - 2\ln 3)$$

$$= \sigma(z) \text{ where } z < 0$$

$$< \frac{1}{2}$$

$\Rightarrow$ We will predict $\boxed{y = -1.}$

(b) **(6 pts)** Is logistic regression a linear or non-linear classifier? Prove your answer.

Logistic regression is a linear classifier because it has a linear decision boundary:

$$\sigma(w^T x_i) = \frac{1}{2} \quad \text{decision boundary}$$

$$\Rightarrow \frac{1}{1 + \exp(-w^T x_i)} = \frac{1}{2}$$

$$\Rightarrow 2 = 1 + \exp(-w^T x_i)$$

$$\Rightarrow 1 = \exp(-w^T x_i)$$

$$\Rightarrow w^T x_i = 0 \quad \text{linear.}$$

(c) **(10 pts)** In the hoemwork, we mention an alternative formulation of learning a logistic regression model when $y \in \{1, 0\}$

$$\arg\min_{w} \sum_{i=1}^{m} y_i \log \sigma(w^T x_i) + (1 - y_i) \log(1 - \sigma(w^T x_i))$$

. Derive its gradient.

$$\sum_i \left[\frac{y_i}{\sigma(w^T x_i)} \sigma(w^T x_i)(1-\sigma(w^T x_i))\, x_{ij} + \frac{1-y_i}{1-\sigma(w^T x_i)} \cdot -\sigma(w^T x_i)(1-\sigma(w^T x_i))\, x_{ij} \right]$$

$$= \sum_i \left[y_i(1-\sigma(w^T x_i))x_{ij} - (1-y_i)\,\sigma(w^T x_i)\, x_{ij} \right]$$

$$= \sum_i \left[y_i x_{ij} - \sigma(w^T x_i) y_i x_{ij} - \sigma(w^T x_i) x_{ij} + \cancel{\sigma(w^T x_i)}\, y_i x_{ij} \right]$$

$$= \sum_{i=1}^{m} \left(y_i - \sigma(w^T x_i)\right) x_{ij}$$

$\Rightarrow$ Call the above expression D_j. The gradient vector is composed of n of these terms, where n is the dimension of w. $\langle D_1, \ldots, D_n \rangle$.