

CM146 Midterm

TOTAL POINTS

76 / 77

QUESTION 1

1 Name+ID 2 / 2

✓ - 0 pts Correct

QUESTION 2

T/F 16 pts

2.1 1-4 8 / 8

- 2 pts 1 incorrect

- 2 pts 2 incorrect

✓ - 0 pts 3 actually correct

- 2 pts 4 incorrect

- 0 pts correct

2.2 5-8 8 / 8

- 2 pts 5 incorrect

- 2 pts 6 incorrect

- 2 pts 7 incorrect

- 2 pts 8 incorrect

✓ - 0 pts Correct

QUESTION 3

Multiple Choice 32 pts

3.1 9 4 / 4

✓ - 0 pts Correct

- 1 pts include d

- 1 pts not include b

- 1 pts include c

- 1 pts include a

3.2 10 4 / 4

- 0 pts Correct

- 0 pts not include b

- 1 pts include a

✓ - 0 pts not include d

- 1 pts not include c

3.3 11 4 / 4

✓ - 0 pts Correct

- 4 pts wrong

3.4 12 4 / 4

✓ - 0 pts Correct

- 1 pts choose c

- 1 pts not choose a

- 1 pts not choose b

- 1 pts choose d

3.5 13 4 / 4

✓ - 0 pts Correct

- 1 pts choose b

- 1 pts choose c

- 1 pts not choosing d

- 1 pts choose a

3.6 14 3 / 4

- 0 pts Correct

✓ - 1 pts not choosing b

- 1 pts not choosing c

- 1 pts choose d

- 1 pts not choosing a

3.7 15 4 / 4

✓ - 0 pts Correct

- 4 pts Incorrect

3.8 16 4 / 4

✓ - 0 pts Correct

- 4 pts Incorrect

QUESTION 4

Decision Tree 12 pts

4.1 Entropy of Y 2 / 2

- ✓ - 0 pts Correct
- 2 pts Incorrect

4.2 Information Gain 6 / 6

- ✓ - 0 pts Correct
- 1 pts Incorrect information gain with correct X1 and X2
- 2 pts Incorrect information gain
- 4 pts Incorrect X1 or X2
- 6 pts Incorrect

4.3 Root choice 1 / 1

- ✓ - 0 pts Correct
- 1 pts Incorrect

4.4 Zero training error 3 / 3

- ✓ - 0 pts Correct
- 3 pts Improper rationale

QUESTION 5

MLE Pareto 8 pts

5.1 Log likelihood 3 / 3

- ✓ - 0 pts Correct
- 2 pts Assumed α was also raised to the power
- 1.5 pts Dropped a log
- 2 pts Assumed $x_1 = x_2 = x_3 = \dots = x_n = x$
- 3 pts Incorrect
- 1.5 pts Didn't raise α to the Nth power
- 2 pts Didn't find likelihood of all the data
- 1 pts Miscellaneous mistake

5.2 Derivative 3 / 3

- ✓ - 0 pts Correct
- 1.5 pts Dropped the sum
- 1 pts Miscellaneous mistakes
- 1.5 pts Dropped a log
- 3 pts Incorrect

5.3 MLE 2 / 2

- ✓ - 0 pts Correct
- 1 pts Didn't solve
- 1 pts Miscellaneous mistakes
- 2 pts No answer
- 1 pts Wrong answer
- 1.5 pts Dropped the sum

QUESTION 6

Least absolute errors 7 pts

6.1 Log likelihood 4 / 4

- ✓ - 0 pts Correct
- 2 pts $\log(ab) = \log(a) + \log(b)$
- 0.5 pts 1 negative sign error
- 2 pts extra y_n
- 3 pts incorrect
- 1 pts Missing term
- 4 pts no attempt

6.2 Equivalence 3 / 3

- ✓ - 0 pts Correct
- 0.5 pts slightly incorrect
- 3 pts incorrect
- 1 pts wrong steps/incorrect derivation

Midterm

Feb. 11th, 2019

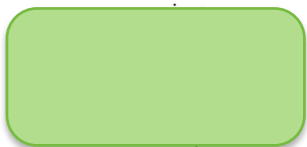
- Please do not open the exam unless you are instructed to do so.
- This is a closed book and closed notes exam.
- Everything you need in order to solve the problems is supplied in the body of this exam OR in a cheatsheet at the end of the exam.
- Mark your answers ON THE EXAM ITSELF. If you make a mess, clearly indicate your final answer (box it).
- For true/false questions, CIRCLE True OR False and provide a brief justification for full credit.
- Unless otherwise instructed, for multiple-choice questions, CIRCLE ALL CORRECT CHOICES (in some cases, there may be more than one) and provide a brief justification if the question asks for one.
- If you think something about a question is open to interpretation, feel free to ask the instructor or make a note on the exam.
- If you run out of room for your answer in the space provided, please use the blank pages at the end of the exam and indicate clearly that you've done so.
- Do NOT put answers on the back of any page of the exam.
- You may use scratch paper if needed (provided at the end of the exam).
- You have 1 hour 45 minutes.
- 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!

Legibly write your name and UID in the space provided below to earn 2 points.

Name:

UID:



Name and UID		/2
True/False		/16
Multiple choice		/32
Decision tree		/12
Maximum likelihood		/8
Least absolute errors		/7
Total		/77

True/False (16 pts)

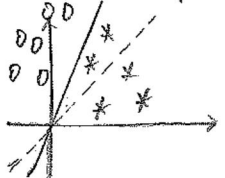
1. (2 pts) The training error of a learning algorithm is an accurate estimate of its generalization error.

True False
 Overfitting the training dataset can reduce the training error, ("memorize" data) but may poorly generalize to future testing data.

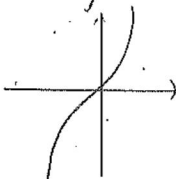
2. (2 pts) We are trying to use logistic regression to predict disease vs healthy ($y \in \{0, 1\}$ where $y = 1$ refers to the disease class) by measuring how many steps a person walked in a week: $x \in \mathbb{R}$. From the training data, we learn a weight $w = -1$ for the feature x and an intercept $b = 1$. This model will predict that the probability of disease increases as a person walks more steps. $y = -x + 1$

True False
 As x increases, \hat{y} decreases due to negative weight. The more negative \hat{y} gets, the more likely we predict y to be 0, which means more healthy. (if use sigmoid for probability).

3. (2 pts) For a binary classification problem, we use the perceptron algorithm to learn weights w on a training dataset (assume the intercept $b = 0$). The error of the perceptron on a test dataset is unchanged if we then rescale the weights w so that they sum to 1.

True False
 Consider a simple example where $\vec{w} \in \mathbb{R}$.

 The solid line is $y = wx = 2x$, and the dotted line is $y = w'x = x$. Clearly, before changing w , there's no error, but after rescaling, there're 2 errors.

4. (2 pts) The function $f(x) = x^3$ is convex over the set of all real numbers.

True False
 $f''(x) = 6x$, which is negative for $x < 0$.


5. (2 pts) To predict y from \mathbf{x} where $y \in \{0, 1\}$, $\mathbf{x} \in \mathbb{R}^D$, we transform \mathbf{x} by a function $\phi(\mathbf{x}) = \mathbf{C}\mathbf{x} + \mathbf{d}$ where \mathbf{C} is a $M \times D$ matrix and $\mathbf{d} \in \mathbb{R}^M$. A logistic regression model with $\phi(\mathbf{x})$ as input can learn a non-linear decision boundary.

True

False

kernel ϕ is a linear transformation. ($\phi(\mathbf{x} + \mathbf{y}) = \phi(\mathbf{x}) + \phi(\mathbf{y})$, $\phi(k\mathbf{x}) = k\phi(\mathbf{x})$).
 So for nonlinear distribution, after ϕ , the data is still nonlinear. So the logistic regression is still not learning the non-linear boundary.

6. (2 pts) The K-nearest neighbor algorithm often uses Euclidean distance as the default distance metric: $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$. Suppose we instead use a new distance metric: $d(\mathbf{x}_i, \mathbf{x}_j) = \log(1 + \|\mathbf{x}_i - \mathbf{x}_j\|_2)$. The classification results will change as a result of this distance metric.

True

False

$d' = \log(1 + \|\mathbf{x}_i - \mathbf{x}_j\|_2) = \log(1 + d)$, which is monotonic increasing.
 So if $d(A, B) < d(A, C)$, then $d'(A, B) < d'(A, C)$ still holds.
 So the k nearest neighbors are still the same.

7. (2 pts) We have a convex function for which we would like to find the minimum. For any choice of step size, gradient descent applied to our problem is always guaranteed to converge to the minimum.

True

False

For convex functions, it is guaranteed that $\nabla = 0$ gives the global minimum, but in practice with gradient descent method, if step size is too large, it may oscillate near the minimum without converging to it.

8. (2 pts) Stochastic gradient descent is guaranteed to converge to the minimum of a convex function faster than batch gradient descent.

True

False

① Stochastic gradient descent is not guaranteed to converge.

② For one iteration, Stochastic gradient descent $O(D)$
 Batch gradient descent $O(ND)$.

but SGD may take more iterations to approach minimum than BGD.

Multiple choice (32 pts)

CIRCLE ALL CORRECT CHOICES (in some cases, there may be more than one)

9. (3 pts) You are given a training dataset for a binary classification problem: $\{(x_1, y_1), \dots, (x_N, y_N)\}$ where $(x_n, y_n), n \in \{1, \dots, N\}$ is an instance-label pair. You use a learning algorithm to build a binary classifier D_1 . You then take one of the instance-label pairs, add a copy of it to the training set (so your new training set will have $N + 1$ instances), and rerun the same learning algorithm to create D_2 (assume any random numbers used in each case are the same). D_1 and D_2 will always have identical decision boundaries when the learning algorithm used is:

(a) Decision Tree (Entropy may change)

(b) 1-Nearest Neighbor ($k=1$ is just returning itself in training set)

(c) 3-Nearest Neighbor (consider $* \begin{matrix} 0 \\ 0 \end{matrix} \Rightarrow ? = 0$, but $(*) \begin{matrix} 0 \\ 0 \end{matrix} \Rightarrow ? = *$)

(d) Perceptron (distribution shape changed, although the original separation boundary still works here, but not guaranteed the algorithm still finds that old line).

10. (4 pts) We want to deploy a machine learning algorithm on a website to predict what ad to show to a visitor to the website based on their view history. To learn this model, we have a training dataset of a million instances and 10 features. Which of the following learning models would be practical for this application?

(a) K-Nearest Neighbors (too many users, finding closest distance too expensive)

(b) Logistic regression (may be okay because weight $\vec{w} \in \mathbb{R}^{10}$, stochastic gradient descent may be practical,

(c) Decision tree (features are not that diverse, only 10, so a tree is manageable) but not guaranteed

(d) Perceptron (too many users, may need too many iterations to converge)

11. (4 pts) Which of the following phenomenon is called over-fitting?

(a) low training error, low test error

(b) low training error, high test error

(c) high training error, low test error

(d) high training error, high test error

over-fitting implies low training error.
if training error is high, then
it's more likely to be under-fitting
(or the model is wrong).

12. (4 pts) In which of these settings is over-fitting more likely?

- (a) Increasing the number of features (*make the model too complicated*)
(b) Increasing the complexity of the hypothesis space (*the model may become too complicated*)
(c) Increasing the value of the regularization hyperparameter
(d) Increasing the number of training examples
- lower over-fitting risk* {

13. (4 pts) Which of the following is true of the Perceptron classifier ?

- (a) If the Perceptron learning algorithm finds a hypothesis that achieves zero training error, this hypothesis also achieves zero test error.
(b) If the Perceptron learning algorithm does not converge after MaxIter iterations, the problem is not linearly separable.
(c) Computational cost of classifying a test instance increases with the size of the training data. (*may need more iterations, but for a single iteration, computational complexity is the same*)
(d) Computational cost of classifying a test instance increases with the number of features. $h = w^T x_i + b$ for example. *then it's $O(D)$. If D is large, then calculation more expensive.*

14. (4 pts) Which of the following methods can achieve zero training error on any linearly separable dataset?

- (a) Perceptron
(b) Logistic regression
(c) 1-Nearest Neighbor
(d) 3-Nearest Neighbor

15. (4 pts) The entropy of a distribution over a set of 4 items with probability mass function p is defined as $-\sum_{k=1}^4 p(k) \log_2 p(k)$. Which of the following distributions has the largest entropy?

- (a) (0, 1, 0, 0) $E = 0$
(b) (1, 0, 0, 0) $E = 0$
(c) (0.25, 0.25, 0.25, 0.25) $\Rightarrow E = 1$ (*uniform distribution*)
(d) (0.5, 0.2, 0.2, 0.1) $E < 1$.

16. (4 pts) Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ be the design matrix with each row corresponding to the features of an example and $\mathbf{y} \in \mathbb{R}^N$ be a vector of all the labels. The OLS solution is $\theta_{OLD} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$. Which of the following is the OLS solution θ_{NEW} if we scale each feature by 2 (i.e., the new dataset is $2\mathbf{X}$)?

- (a) $2\theta_{OLD}$
(b) $4\theta_{OLD}$
(c) $\frac{1}{2}\theta_{OLD}$
(d) θ_{OLD}

$$\begin{aligned}\theta_{new} &= (2\mathbf{X}^T, 2\mathbf{X})^{-1} (2\mathbf{X})^T \mathbf{y} \\ &= \frac{1}{4} (\mathbf{X}^T \mathbf{X})^{-1} \cdot 2\mathbf{X}^T \mathbf{y} \\ &= \frac{1}{2} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \frac{1}{2} \theta_{old}.\end{aligned}$$

Decision Tree (12 pts)

Decision tree learning

Given the following set of training observations with two features (X_1, X_2) and the response variable Y , we would like to learn a decision tree using information gain to choose nodes. Recall that the information gain is defined as $Gain = H[Y] - H[Y|X]$, where $H[Y] = -E[\log_2 P(Y)]$ is the entropy. See cheatsheet at the end of this exam for entropy values.

X_1	X_2	Y
0	1	1
1	1	0
0	0	1
1	0	1
0	1	0
0	0	0

(a) (2 pts) What is the entropy of Y ?

$$H[Y] = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = 1$$

- (b) (6 pts) What is the information gain of each of the attributes X_1 and X_2 relative to Y ?

$$\begin{aligned} H[\text{Data} | X_1] &= \frac{2}{6} H[\text{Data} | X_1=1] + \frac{4}{6} H[\text{Data} | X_1=0] \\ &= \frac{1}{3} \cdot 1 + \frac{2}{3} \cdot 1 \\ &= 1 \end{aligned}$$

$$\text{Info Gain on } X_1 = 1 - 1 = 0$$

$$\begin{aligned} H[\text{Data} | X_2] &= \frac{3}{6} H[\text{Data} | X_2=1] + \frac{3}{6} H[\text{Data} | X_2=0] \\ &= \frac{1}{2} \left(-\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \right) + \frac{1}{2} \left(-\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \right) \\ &= -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \\ &\approx 0.92 \end{aligned}$$

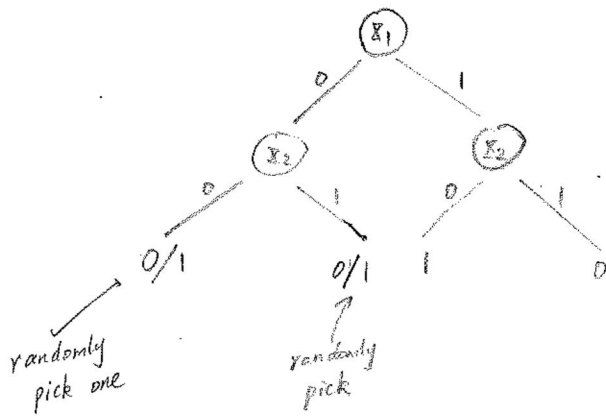
$$\text{Info Gain on } X_2 = 1 - 0.92 = 0.08$$

- (c) (1 pts) Using information gain, which attribute will the ID3 decision tree learning algorithm choose as the root?

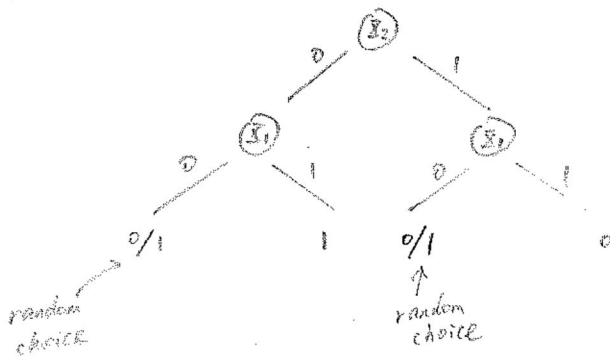
Choose the one with largest info gain.

So pick X_2 as root attribute.

(d) (3 pts) Can we construct a decision tree with zero training error on this training data? If yes, provide an example. If no, justify.



cannot achieve $err = 0$.



cannot.

Because for the same (X_1, X_2) tuple, it's possible in the given training set that the Y 's are different.
 So no D-tree can classify with no errors.

Maximum Likelihood (8 pts)

Let X_1, \dots, X_N be i.i.d. random variables where $X_n \sim \text{Pareto}(\alpha), n \in \{1, \dots, N\}$. The probability density function for $X \sim \text{Pareto}(\alpha)$ is:

$$f(x; \alpha) = \begin{cases} \alpha x^{-(\alpha+1)}, & \text{if } x \geq 1 \\ 0, & \text{if } x < 1 \end{cases}$$

- (a) (3 pts) Give an expression for the log likelihood $l(\alpha)$ as a function of α given this specific dataset. You may assume all values, $X_1, \dots, X_N \geq 1$.

$$\begin{aligned} \log l(\alpha) &= \log \prod_{i=1}^N P(X_i = x_i) \\ &= \log \prod_{i=1}^N \alpha x_i^{-\alpha-1} \\ &= \sum_{i=1}^N \log \alpha x_i^{-\alpha-1} \\ &= \sum_{i=1}^N \log \alpha - (\alpha+1) \log x_i \\ &= N \log \alpha - (\alpha+1) \sum_{i=1}^N \log x_i \quad (\text{convex}). \end{aligned}$$

(b) (3 pts) Compute the derivative of the log likelihood for this specific dataset.

$$\frac{d \ell \ell}{d \alpha} = \frac{N}{\alpha} - \sum_{i=1}^N \log x_i$$

(c) (2 pts) What is the maximum likelihood estimate $\hat{\alpha}$ of α ?

$$\text{Set } \frac{d \ell \ell}{d \alpha} = 0 \Rightarrow \frac{N}{\alpha} = \sum_{i=1}^N \log x_i$$
$$\hat{\alpha} = \frac{N}{\sum_{i=1}^N \log x_i}$$

$$\text{for } \alpha < \hat{\alpha}, \quad \ell \ell' > 0.$$

$$\alpha > \hat{\alpha}, \quad \ell \ell' < 0.$$

So $\hat{\alpha}$ is at maximum point.

Least absolute errors (7 pts)

In class, we showed how linear regression (ordinary least squares) can be interpreted as a probabilistic model. In this problem, we consider an alternative model for regression. We have a training set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ where $\mathbf{x}_n \in \mathbb{R}^D$ and $y_n \in \mathbb{R}$. Now we model our target y_n as distributed according to the following distribution:

$$p(y_n | \mathbf{x}_n; \theta) = \frac{1}{2b} \exp\left(-\frac{1}{b} |y_n - \theta^T \mathbf{x}_n|\right)$$

Here $|z|$ is the absolute value of z and $b > 0$ is a constant that is assumed to be known.

- (a) (4 pts) Write the log likelihood of the parameters $l(\theta)$. Express your answer in terms of $y_n, \mathbf{x}_n, \theta$.

$$\begin{aligned} & \log P(Y | X; \theta) \\ &= \log \prod_{n=1}^N p(y_n | \mathbf{x}_n; \theta) \\ &= \log \prod_{n=1}^N \frac{1}{2b} e^{-\frac{1}{b} |y_n - \theta^T \mathbf{x}_n|} \\ &= \sum_{n=1}^N \log \frac{1}{2b} e^{-\frac{1}{b} |y_n - \theta^T \mathbf{x}_n|} \\ &= \sum_{n=1}^N \log \frac{1}{2b} - \frac{1}{b} |y_n - \theta^T \mathbf{x}_n| \\ &= \text{const} - \frac{1}{b} \sum_{n=1}^N |y_n - \theta^T \mathbf{x}_n| \end{aligned}$$

- (b) (3 pts) Show that finding the maximum likelihood estimate of θ leads to the same answer as finding the θ that minimizes the cost function (which is the sum of absolute errors):

$$J(\theta) = \sum_{n=1}^N |y_n - \theta^T x_n|$$

$$\begin{aligned} \operatorname{argmax}_{\theta} \ell &= \operatorname{argmax}_{\theta} \text{const} - \frac{1}{b} \sum_{n=1}^N |y_n - \theta^T x_n| \\ &= \operatorname{argmin}_{\theta} \frac{1}{b} \sum_{n=1}^N |y_n - \theta^T x_n|, \quad \text{because of the negative sign.} \\ &= \operatorname{argmin}_{\theta} \sum_{n=1}^N |y_n - \theta^T x_n|, \quad \text{because } \frac{1}{b} \text{ is positive constant.} \\ &= \operatorname{argmin}_{\theta} J(\theta). \end{aligned}$$

Identities

Probability density/mass functions for some distributions

$$\text{Normal : } P(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

$$\text{Multinomial : } P(\mathbf{x}; \boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{x_k}$$

\mathbf{x} is a length K vector with exactly one entry equal to 1 and all other entries equal to 0

$$\text{Poisson : } P(x; \lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}$$

Matrix calculus

Here $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$. \mathbf{A} is symmetric.

$$\begin{aligned}\nabla \mathbf{x}^T \mathbf{A} \mathbf{x} &= 2\mathbf{A} \mathbf{x} \\ \nabla \mathbf{b}^T \mathbf{x} &= \mathbf{b}\end{aligned}$$

Entropy

The entropy $H(X)$ of a Bernoulli random variable $X \sim \text{Bernoulli}(p)$ for different values of p :

p	$H(X)$
$\frac{1}{2}$	1
$\frac{1}{3}$	0.92
$\frac{1}{4}$	0.81
$\frac{1}{5}$	0.73
$\frac{2}{5}$	0.97

You may use this page for scratch space.