CM146 Midterm

Jonathan (nyan) Tun

TOTAL POINTS

54 / 60

QUESTION 1

1 Machine Learning Basics **6 / 6**

✓ **- 0 Correct**

QUESTION 2

2 Logistic Regression **4 / 4**

✓ **- 0 Correct**

QUESTION 3

3 True/False **8 / 12**

✓ **- 2 Q4 Incorrect**

✓ **- 2 Q6 Incorrect**

QUESTION 4

Multiple Choice 7 pts

4.1 **2 / 2**

✓ **- 0 Correct**

4.2 **0 / 2**

✓ **- 2 incorret/not answered**

4.3 **3 / 3**

✓ **- 0 Correct**

QUESTION 5

5 Maximum Likelihood **5 / 5**

✓ **- 0 Correct**

QUESTION 6

Decision Trees 10 pts

6.1 Entropy Y **1 / 1**

✓ **- 0 Correct**

6.2 Information Gain **4 / 4**

✓ **- 0 Correct**

6.3 Root split **1 / 1**

✓ **- 0 Correct**

6.4 Zero training error tree **2 / 2**

✓ **- 0 Correct**

6.5 Change instance **2 / 2**

✓ **- 0 Correct**

QUESTION 7

Weighted Linear Regression 16 pts

7.1 objective function **3 / 3**

✓ **- 0 Correct**

7.2 optimal value **5 / 5**

✓ **- 0 Correct**

7.3 OLS log likelihood **3 / 3**

✓ **- 0 Correct**

7.4 MLE; variance and weight relation **5 / 5**

✓ **- 0 Correct**

CM 146 — Machine Learning: Midterm

Fall 2017

Name: Nyan Tun

UID: 204607621

Instructions:

1. This exam is CLOSED BOOK and CLOSED NOTES.
2. You may use scratch paper if needed.
3. The time limit for the exam is 1hour, 45 minutes.
4. Mark your answers ON THE EXAM ITSELF. If you make a mess, clearly indicate your final answer (box it).
5. For true/false questions, CIRCLE True OR False and provide a brief justification for full credit.
6. 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.
7. If you think something about a question is open to interpretation, feel free to ask the instructor or make a note on the exam.

Q	Problem	Points	Score
1	ML basics	6	
2	Application	4	
3	True/False	12	
4	Multiple choice	7	
5	Maximum likelihood	5	
6	Decision Trees	10	
7	Regression	16	
	Total	60	

1. (6 pts) **Machine Learning Basics**

(a) (2 pts) Consider supervised and unsupervised learning. What is the main difference in the inputs <u>and</u> the goals?

For supervised learning the inputs are labelled whereas for unsupervised, the inputs are not. The goal for supervised learning is to be able to predict which class a new instance falls under given a training data set with labels. The goal for unsupervised is to find common patterns between inputs that were not known prior

(b) (2 pts) What is the main difference between classification and regression?

Classification is concerned with placing a new test instance in a set of given classes (could be binary or multiclass) This usually involves finding a hyperplane that separates the data. As for regression, it is a way of evaluating how the output is correlated to a given training set X. A common method is linear regression, which uses the OLS method.

(c) (2 pts) What is the motivation to separate the available data into training and test data?

Without splitting the data into test and train, there is no way of assessing how well a model performs. With test and train, a model can be fitted to the training data. Then h(x) values can then be predicted using the model for the test data. How far off h(x) is from y gives an accuracy score of the model.

2. (4 pts) **Application** Suppose you are given a dataset of cellular images from patients with and without cancer.

(a) (2 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, it predicts the probability that an instance is within a given class rather than the class itself.

(b) (2 pts) A model that attains 100% accuracy on the training set and 70% accuracy on the test set is better than a model that attains 80% accuracy on the training set and 75% accuracy on the test set.

True

False

Model A - 100% accuracy train, 70% accuracy test

Model B - 80% accuracy train, 75% accuracy test

There is the possibility that Model A may be overfitting. This occurs when the model is trained to fit the training data exactly such that the model generalizes poorly. This could explain why Model B has lower training accuracy but performs better on test than A does.

Overfitted models are bad.

$H[X] = \sum_{k=1}^{K} P(X=a_k) \log P(X=a_k)$

True/False

3. (2 pts) You are given a training dataset with attributes $A_1, \ldots, A_m$ and instances $x^{(1)}, \ldots, x^{(n)}$ and you use the ID3 algorithm to build a decision tree D_1. You then take one of the instances, add a copy of it to the training set (so your new training set will have $n+1$ instances), and rerun the decision tree learning algorithm (with the same random seed) to create D_2. D_1 and D_2 are necessarily identical decision trees.

   True   False

   Repeating an instance will change the weight of the feature with that value ($p(X=a_k)$) and also change the weight of its corresponding output ($p(Y=c \mid X=a_k)$)

   Thus, entropy and info. gain could be changed to the point where the hierarchy is different.

4. (2 pts) Stochastic Gradient Descent is faster per iteration than Batch Gradient Descent.

   True   False

   SGD is faster than GD because it updates the step size using only one point whereas GD uses all N data points. ∴ SGD = O(D) and GD is O(ND). Each iteration is O(D) for both.

5. (2 pts) You run the PerceptronTrain algorithm with $maxIter = 100$. The algorithm terminates at the end of 100 iterations with a classifier that attains a training error of 1%. This means that the training data is not linearly separable.

   True   False

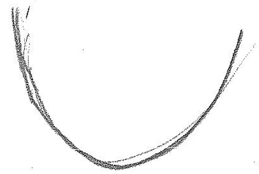
   Even if the data is lin. separable, if the maxIter is low, then after the iterations, the model may not have reached convergence yet.

   e.g maxIter < N

6. (2 pts) We want to learn a non-linear regression function to predict y from $\boldsymbol{x}$ where $y \in \mathbb{R}, \boldsymbol{x} \in \mathbb{R}^D$ given training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. To do so, we transform $\boldsymbol{x}$ by a function $\boldsymbol{\phi}(\boldsymbol{x})$ and minimize the residual sum of squares objective function on the transformed features: $\sum_{i=1}^n \left(y_i - \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_i)\right)^2$. This optimization problem is convex.

True

False



The optimization may be convex or not depending on $\phi(x)$

7. (2 pts) We want to use 1-Nearest Neighbors (1-NN) to classify houses into one of two classes (cheap vs expensive) given a single feature that measures the area of the house. The predictions made by the 1-NN classifier data can change if the area of the house is measured in square metres instead of square feet. (You can neglect the effect of ties *i.e.*, two training instances that are both nearest neighbors to a test instance.)

True

False

Adjusting the units will change the distance computed for all points, keeping the distance rankings the same.

8. (2 pts) You run gradient descent to minimize the function $f(x) = (2x-3)^2$. Assume the step size has been chosen appropriately and you run gradient descent till convergence. Then gradient descent will return the global minimum of f.

True

False

$f'(x) = 2(2x-3) = 4x-6$

$f''(x) = 4$ ← convex $f''(x) > 0$

Multiple choice

9. (2 pts) In k-nearest neighbor classification, which of the following statements are true? (circle all that are correct)

   (a) The decision boundary is smoother with smaller values of k.

   (b) k-NN does not require any parameters to be learned in the training step (for a fixed value of k and a fixed distance function).

   (c) If we set k equal to the number of instances in the training data, k-NN will predict the same class for any input.

   × (d) For larger values of k, it is more likely that the classifier will overfit than underfit.

10. (2 pts) Assume we are given a set of one-dimensional inputs and their corresponding output (that is, a set of $\{(x_i, y_i)\}, x_i \in \mathbb{R}, y_i \in \mathbb{R}$). We would like to compare the following two models on our input dataset where $\theta \in \mathbb{R}$:

$$\mathrm{A}: y = \theta^2 x$$
$$\mathrm{B}: y = \theta x$$

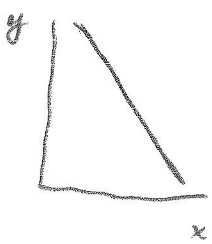
For each model, we split into training and testing set to evaluate the learned model. Which of the following is correct? Choose the answer that best describes the outcome, and provide justification.

   (a) There are datasets for which A would be more *accurate* than B.

   (b) There are datasets for which B would be more *accurate* than A.

   (c) Both (a) and (b) are correct.

   (d) They would perform equally well on all datasets.



Both A and B are linear models where the ideal weights are iteratively solved for. For any given data set, the weights for B will be $= \theta_A^2$

$\theta_A^2 = \theta_B$, ∴ both will work equally.

11. (3 pts) If your model is overfitting, increasing the training set size (by drawing more instances from the underlying distribution) will tend to result in which of the following? (circle the best answer for each)

   (a) training error will ... increase / decrease / unknown

   (b) test error will ... increase / decrease / unknown

   (c) overfitting will ... increase / decrease / unknown

For these problems, you must show your work to receive credit!

Maximum likelihood

$(P(X_n = 1) = \theta)$

12. We observe the following data consisting of four independent random variables $X_n, n \in \{1, \dots, 4\}$ drawn from the same Bernoulli distribution with parameter θ (*i.e.*, $P(X_n = 1) = \theta$): $(X_1, X_2, X_3, X_4) = (1, 1, 0, 1)$.

(a) Give an expression for the log likelihood $l(\theta)$ as a function of θ given this specific dataset. [2 pts]

$$L(\theta) = \prod_{n=1}^{\{1,\dots,4\}} \begin{cases} \theta & \text{if } X_n = 1 \\ 1-\theta & \text{if } X_n = 0 \end{cases}$$

α = # of times $X_n = 1$

β = # of times $X_n = 0$

$$L(\theta) = \theta^{\alpha}(1-\theta)^{\beta}$$

$$\ell(\theta) = \alpha \log\theta + \beta \log(1-\theta)$$

$$\ell(\theta) = 3\log\theta + \log(1-\theta)$$

for (X_1, X_2, X_3, X_4)

(b) Give an expression for the derivative of the log likelihood for this specific dataset. [2 pts]

$$\ell(\theta) = 3\log\theta + \log(1-\theta)$$

$$\ell'(\theta) = \frac{\partial\, 3\log\theta}{\partial\theta} + \frac{\partial \log(1-\theta)}{\partial\theta}$$

$$= \frac{3}{\theta} - \frac{1}{1-\theta}$$

$$\ell'(\theta) = \frac{3}{\theta} - \frac{1}{1-\theta}$$

(c) What is the maximum likelihood estimate $\hat{\theta}$ of θ? [1 pts]

$$\ell'(\theta) = \frac{3}{\theta} - \frac{1}{1-\theta} = 0$$

$$\frac{3}{\theta} = \frac{1}{1-\theta}$$

$$3(1-\theta) = \theta$$

$$3 - 3\theta = \theta$$

$$3 = 4\theta$$

$$\hat{\theta} = \frac{3}{4}$$

$$\hat{\theta} = \frac{3}{4}$$

Decision Trees

13. We would like to learn a decision tree given the following pairs of training instances with attributes (a_1, a_2) and target variable Y.

Instance number	a_1	a_2	Y
1	T	T	T
2	T	T	T
3	T	F	F
4	F	F	T
5	F	T	F
6	F	T	F

For reference, for a random variable X that takes on two values with probability p and $1-p$, here are some values of the entropy function (we use **log to the base 2** in this question):

$$p = \tfrac{1}{2} : H(X) = 1 \qquad\qquad p \in \{\tfrac{1}{3}, \tfrac{2}{3}\} : H(X) \approx .92$$

(a) What is the entropy of Y? [1 pts]

$$H[Y] = -\sum_{k=1}^{k} P(Y = a_k) \log P(Y = a_k)$$

$$= -\left(P(Y=T) \log P(Y=T) + P(Y=F) \log P(Y=F)\right)$$

$$= -\left(\tfrac{1}{2} \log \tfrac{1}{2} + \tfrac{1}{2} \log \tfrac{1}{2}\right)$$

$$= 1$$

(b) What is the information gain of each of the attributes a_1 and a_2 relative to Y? [4 pts]

$$H[Y|a_1] = -\sum_{k=1}^{K} P(a_1 = a_k) H[Y|a_1 = a_k]$$

a_1

$$H[Y|a_1] = -(P(a_1=T)H[Y|a_1=T] + P(a_1=F)H[Y|a_1=F])$$

$$H[Y|a_1=T] = -(\tfrac{2}{3}\log\tfrac{2}{3} + \tfrac{1}{3}\log\tfrac{1}{3}) \approx 0.92$$

$$H[Y|a_1=F] = -(\tfrac{1}{3}\log\tfrac{1}{3} + \tfrac{2}{3}\log\tfrac{2}{3}) \approx 0.92$$

$$H[Y|a_1] = \tfrac{1}{2}(0.92) + \tfrac{1}{2}(0.92) = 0.92$$

info gain $a_1 = H[Y] - H[Y|a_1]$
$= 1 - 0.92$
$\approx \underline{0.08}$

a_2

$$H[Y|a_2=T] = -(\tfrac{1}{2}\log\tfrac{1}{2} + \tfrac{1}{2}\log\tfrac{1}{2}) = 1$$

$$H[Y|a_2=F] = -(\tfrac{1}{2}\log\tfrac{1}{2} + \tfrac{1}{2}\log\tfrac{1}{2}) = 1$$

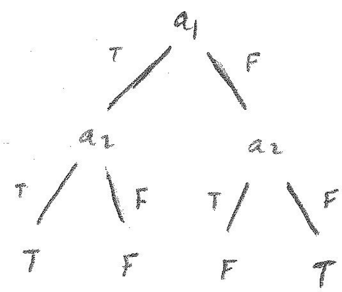
$$H[Y|a_2] = \tfrac{1}{2}(1) + \tfrac{1}{2}(1) = 1$$

info gain $a_2 = H[Y] - H[Y|a_2]$
$= 1 - 1$
$= \underline{0}$

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

$\underline{a_1}$ $\quad 0.08 > 0$

(d) Construct a decision tree with zero training error on this training data. [2 pts]



(e) Change exactly one of the instances (by changing either the attributes or labels but not both) so that **no decision tree can attain zero training error** on this dataset (indicate the instance number and the change). [2 pts]

Change instance number 5

a_2 value to F

Weighted linear regression

14. In the problem set, we considered weighted linear regression where the input features are 1-dimensional. We now extend this to D-dimensional features. Thus, we want to find $\boldsymbol{\theta}$ that minimizes the cost function

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} w_n (y_n - \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x}_n)^2$$

Here $w_n > 0$, $\boldsymbol{x}_n \in \mathbb{R}^{D+1}, \boldsymbol{\theta} \in \mathbb{R}^{D+1}$. $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_N^{\mathrm{T}} \end{pmatrix} \in \mathbb{R}^{N\times(D+1)}$, $\boldsymbol{y} \in \mathbb{R}^N$. For this problem, assume that the intercept term is included in the $\boldsymbol{\theta}$ and that the linear regression solution exists in this setting.

Questions:

(a) Show that $J(\boldsymbol{\theta})$ can also be written as:

$$J(\boldsymbol{\theta}) = \underbrace{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^{\mathrm{T}}}_{1\times N} \underbrace{\boldsymbol{W}}_{N\times N} \underbrace{(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})}_{N\times 1} \quad N\times 1$$

Here $\boldsymbol{W}$ is a diagonal matrix where the entry on the diagonal on row n, column n is w_n. [3 pts]

$J(\theta) = (y - X\theta)^T W (y - X\theta)$

$(y - X\theta) = \begin{pmatrix} y_1 - x_1^T\theta \\ y_2 - x_2^T\theta \\ \cdots \\ y_n - x_n^T\theta \end{pmatrix}$ $\quad (y - X\theta)^T = (y_1 - x_1^T\theta \quad y_2 - x_2^T\theta \ \ldots \ y_n - x_n^T\theta)$

$W = \begin{bmatrix} w_1 & 0 & 0 & 0 \to \\ 0 & w_2 & & \\ \downarrow & & w_3 & \\ & & & w_4 \ \ldots \\ & & & & w_n \end{bmatrix}$

$W(y - X\theta) = \begin{bmatrix} w_1(y_1 - x_1^T\theta) \\ w_2(y_2 - x_2^T\theta) \\ \cdots \\ w_3(y_n - x_n^T\theta) \end{bmatrix}$

$(y - X\theta)^T W (y - X\theta) = w_1(y_1 - x_1^T\theta)^2 + w_2(y_2 - x_2^T\theta)^2 + w_3(y_3 - x_3^T\theta)^2$

$= \sum_{n=1}^{N} w_n (y_n - x_n^T\theta)^2 = \sum_{n=1}^{N} w_n (y_n - \theta^T x_n)^2$

(b) Show that the optimal value for $\widehat{\theta} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{y}$. For reference, here are some useful gradient identities (where $\boldsymbol{x}, \boldsymbol{b}$ are vectors and $\boldsymbol{A}$ is a symmetric matrix).

$$f(\boldsymbol{x}) = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} \qquad \nabla f(\boldsymbol{x}) = \boldsymbol{b}$$
$$f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} \qquad \nabla f(\boldsymbol{x}) = 2\boldsymbol{A}\boldsymbol{x}$$

[5 pts]

$$J(\theta) = (y - X\theta)^T W (y - X\theta)$$

$$\nabla J(\theta) = 2W(y - X\theta) \cdot \frac{\partial(y - X\theta)}{\partial\theta}$$

$$= 2X^T W(y - X\theta)$$

$$= 2X^T W y - 2X^T W X\theta$$

$$\nabla J(\theta) = 0$$

$$X^T W y - X^T W X \theta = 0$$

$$X^T W X \theta = X^T W y$$

$$\therefore \hat{\theta} = (X^T W X)^{-1} X^T W y$$

//

(c) In class, we provided a probabilistic interpretation of ordinary least squares (OLS). We now try to provide a probabilistic interpretation of weighted linear regression. Consider a model where each of the N samples is independently drawn according to a normal distribution

$$P(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(y_n - \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n)^2}{2\sigma_n{}^2}\right) \qquad N(\theta^T x_n, \sigma_n^2)$$

In this model, each y_n is drawn from a normal distribution with mean $\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n$ and variance $\sigma_n{}^2$. The $\sigma_n{}^2$ are **known**. Write the log likelihood of this model as a function of $\boldsymbol{\theta}$. [3 points]

$$L(\theta) = \prod_{n=1}^{N} P(y_n|x_n, \theta)$$

$$\ell(\theta) = \sum_{n=1}^{N} \log P(y_n|x_n,\theta) = \sum_{n=1}^{N} \left\{ \log \frac{1}{\sqrt{2\pi\sigma_n^2}} - \frac{(y_n - \theta^T x_n)^2}{2\sigma_n^2} \right\}$$

$$\ell(\theta) = -\frac{1}{2} \sum_{n=1}^{N} \left\{ \log(2\pi\sigma_n^2) + \frac{1}{\sigma_n^2}(y_n - \theta^T x_n)^2 \right\}$$

(d) Show that finding the maximum likelihood estimate of $\boldsymbol{\theta}$ leads to the same answer as solving a weighted linear regression. How do $\sigma_n{}^2$ relate to w_n? [5 points]

$$\ell'(\theta) = -\frac{1}{2} \sum_{n=1}^{N} \frac{\{2(y_n - \theta^T x_n)\} x_n}{\sigma_n^2}$$

$$= -\sum_{n=1}^{N} \frac{1}{\sigma_n^2}(y_n - \theta^T x_n) x_n$$

$$\frac{1}{\sigma_n^2}(y_n - \theta^T x_n)^2$$

$$\downarrow$$

$$\frac{2}{\sigma_n^2}(y_n - \theta^T x_n) \cdot \frac{\partial(y_n - \theta^T x_n)}{\partial\theta}$$

$$= \frac{2}{(\sigma_n^2)}(y_n - \theta^T x_n) \cdot x_n$$

same form

$$J(\theta) = \sum_{n=1}^{N} w_n (y_n - \theta^T x_n)^2$$

$$J'(\theta) = 2\sum_{n=1}^{N} w_n (y_n - \theta^T x_n) x_n$$

$$w_n (y_n - \theta^T x_n)^2 \downarrow$$

$$2w_n(y_n - \theta^T x_n) \cdot \frac{\partial(y_n - \theta^T x_n)}{\partial\theta}$$

$$= 2w_n(y_n - \theta^T x_n) \cdot x_n$$

$$2w_n = -\frac{1}{\sigma_n^2} \rightarrow \boxed{w_n = -\frac{1}{2\sigma_n^2}}$$

(Blank page provided for your work)