

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	
<b>Total</b>		<b>60</b>	

1. (6 pts) Machine Learning Basics

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

- (1) Supervised: the inputs are data with feature and given labels, and we want a model that works well predicting the label of feature [instances]. We train such a model from the given input data, so that it can be used to predict data not in the input training data.
- (2) Unsupervised: the inputs are data with feature but NO labels. We want to learn the hidden model ~~and~~ <sup>pattern</sup> behind those data.

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

Classification: put the input into some fixed categories.  
i.e. the result <sub>output</sub> is discrete.

Regression: the result <sub>output</sub> is ~~usually~~ continuous.

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

- ① so that training and testing data never overlap to make sure <sup>the</sup> the model has no idea about testing data before actually testing, so that it can't "remember" the test and do well.
- ② split from the same data set. so the data ~~will~~ will have <sup>the</sup> same distribution, so that our training and testing make sense (train with it, and also test <sup>on</sup> what has been trained)

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. because the model of its outputs is the result of a  $\sigma$  function, which is a probability (between 0 and 1)

(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

- ① we prefer models with higher test data accuracy
- ② 100% accuracy on training set has a large possibility of overfitting

## True/False

3. (2 pts) You are given a training dataset with attributes  $A_1, \dots, A_m$  and instances  $x^{(1)}, \dots, x^{(n)}$  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

ID3 based information gain. By putting a copy to the training set, the information gain of the feature related has a large possibility of being changed. So ~~the~~  $D_1, D_2$  not guaranteed to be the same.

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

True

False

In one iteration, SGD updates based on one data point. While BGD looks at the whole data set to perform an update. Specifically, SGD upgrades in  $O(1)$  while BGD in  $O(n)$ .

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

It is possible that it will converge after 100 iterations and reduce the training error to 0% - which also means the data is linearly separable.

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

True False

By doing this transformation, we transform this problem into a linear regression problem, where closed form solution (optima) can be found  $\Rightarrow$  convex

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

We always do data normalization for k-NN problem, so that the distribution of data points actually reflects the distance relation. In this sense, if ~~on~~ the <sup>unit of</sup> area changes, ~~the~~ unit of our prediction model changes correspondingly, which ~~will~~ affect the distribution thus the distance order is the same  $\Rightarrow$  prediction will not change

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

$f(x) = 2(2x-3)^2 = 8x^2 - 12x + 18$   
 $f'(x) = 8 > 0 \therefore f(x)$  is convex

$\therefore$  gradient descent will always reach a local ~~minima~~ <sup>minima</sup>, in this case a global minima. (convex)

## 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. (if we deal with tie appropriately)
- (d) For larger values of  $k$ , it is more likely that the classifier will overfit than underfit.



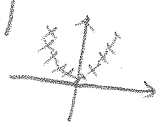

10. (2 pts) Assume we are given a set of one-dimensional inputs and their corresponding output (that is, a set of  $\{(x_i, y_i)\}, x_i \in \mathbb{R}, y_i \in \mathbb{R}$ ). We would like to compare the following two models on our input dataset where  $\theta \in \mathbb{R}$ :

$$A : y = \theta^2 x$$

$$B : y = \theta x$$

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.

A model is a 'parabola' , B is a line.   
 So if the data in a data set is in the shape of parabola:   
 then fits A better: if a line , then B better.

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

→ note at the beginning when new instances come, training error will increase but the final result is unknown.

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

## Maximum likelihood

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  $\theta$  (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  $\theta$  given this specific dataset. [2 pts]

$$\begin{aligned} P(X_i; \theta) &= \theta^{x_i} \cdot (1-\theta)^{(1-x_i)} \\ \Rightarrow L(\theta) &= \prod_{i=1}^4 \theta^{x_i} \cdot (1-\theta)^{(1-x_i)} \\ \Rightarrow \log(L(\theta)) &= \sum_{i=1}^4 [x_i \cdot \log(\theta) + (1-x_i) \cdot \log(1-\theta)] \\ l(\theta) &= \log \theta + \log \theta + \log(1-\theta) + \log \theta \\ &= 3 \log \theta + \log(1-\theta) \end{aligned}$$

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

$$\begin{aligned} \frac{d l(\theta)}{d \theta} &= \frac{3}{\theta} + (-1) \cdot \frac{1}{1-\theta} \\ &= \frac{3}{\theta} - \frac{1}{1-\theta} \end{aligned}$$

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

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

$$\text{let } \frac{d(l(\theta))}{d\theta} = 0$$

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

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



# 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):

$$p = \frac{1}{2} : H(X) = 1$$

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

- (a) What is the entropy of  $Y$ ? [1 pts]  $T:3 \quad F:3 \Rightarrow p = \frac{1}{2}$

$$\begin{aligned} H[Y] &= -\frac{1}{2} \log_2\left(\frac{1}{2}\right) - \frac{1}{2} \log_2\left(\frac{1}{2}\right) \\ &= 1 \end{aligned}$$

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

For  $a_1$ : T:  $H = -\frac{1}{3} \log(\frac{1}{3}) - \frac{2}{3} \log(\frac{2}{3}) \approx 0.92$

F:  $H = -\frac{1}{3} \log(\frac{1}{3}) - \frac{2}{3} \log(\frac{2}{3}) \approx 0.92$

$\therefore H(Y|a_1) = \frac{1}{2} \cdot 0.92 + \frac{1}{2} \cdot 0.92 = 0.92$

Information Gain:  $H(Y) - H(Y|a_1) = 1 - 0.92 = 0.08$

For  $a_2$ : T:  $H = -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1$

F:  $H = -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1$

$H(Y|a_2) = \frac{4}{6} \times 1 + \frac{2}{6} \times 1 = 1$

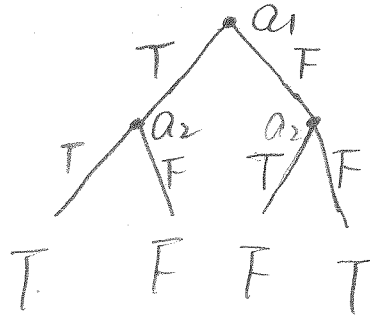
Information gain =  $H(Y) - H(Y|a_2) = 1 - 1 = 0$

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

ID3 chooses attribute with largest information gain

so choose  $a_1$ .

(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]

Instance 1 = change  $a_2$  to F.

so <sup>instance</sup> 1:  $a_1=T, a_2=F \Rightarrow T$ .

While instance 3:  $a_1=T, a_2=F \Rightarrow F$ .

no zero training error tree

## 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  $\theta$  that minimizes the cost function

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

Here  $w_n > 0$ ,  $x_n \in \mathbb{R}^{D+1}$ ,  $\theta \in \mathbb{R}^{D+1}$ .  $X = \begin{pmatrix} x_1^T \\ \vdots \\ x_N^T \end{pmatrix} \in \mathbb{R}^{N \times (D+1)}$ ,  $y \in \mathbb{R}^N$ . For

this problem, assume that the intercept term is included in the  $\theta$  and that the linear regression solution exists in this setting.

**Questions:**

- (a) Show that  $J(\theta)$  can also be written as:

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

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

$$\begin{aligned} \text{RHS} &= \begin{pmatrix} y_1 - x_1^T \theta \\ \vdots \\ y_N - x_N^T \theta \end{pmatrix}^T \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_N \end{pmatrix} \begin{pmatrix} y_1 - x_1^T \theta \\ \vdots \\ y_N - x_N^T \theta \end{pmatrix} \\ &= (y_1 - x_1^T \theta \quad y_2 - x_2^T \theta \quad \dots \quad y_N - x_N^T \theta) \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_N \end{pmatrix} \begin{pmatrix} y_1 - x_1^T \theta \\ \vdots \\ y_N - x_N^T \theta \end{pmatrix} \\ &= (w_1 (y_1 - x_1^T \theta) \quad \dots \quad w_N (y_N - x_N^T \theta)) \begin{pmatrix} y_1 - x_1^T \theta \\ \vdots \\ y_N - x_N^T \theta \end{pmatrix} \\ &= \sum_{i=1}^N w_i (y_i - x_i^T \theta)^2 = \text{LHS} = J(\theta) \end{aligned}$$

( $\theta^T \cdot x_n$  is scalar)  
 $\therefore (\theta^T \cdot x_n)^T = x_n^T \cdot \theta = \theta^T \cdot x_n$

$$y^T \quad (1 \times N) \times (N \times N) \cdot N \times 1$$

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

$$\begin{aligned} f(x) &= b^T x & \nabla f(x) &= b \\ f(x) &= x^T A x & \nabla f(x) &= 2Ax \end{aligned}$$

[5 pts]

$$\begin{aligned} J(\theta) &= (y - X\theta)^T W (y - X\theta) \\ &= (y^T - (X\theta)^T) W (y - X\theta) \\ &= (y^T - \theta^T X^T) \cdot W (y - X\theta) \\ &= y^T \cdot W (y - X\theta) - \theta^T X^T W (y - X\theta) \\ &= y^T \cdot W \cdot y - y^T W \cdot X\theta - \theta^T X^T W y + \theta^T X^T W X \theta \\ &= \text{const} - y^T W \cdot X\theta - \theta^T X^T W y + \theta^T X^T W X \theta \end{aligned}$$

$$\therefore y^T W (X\theta) = (1 \times N) (N \times N) (N \times 1) = (1 \times 1)$$

$$\therefore y^T W (X\theta) = (y^T W X \theta)^T = \theta^T X^T W^T y$$

$$\text{and since } W \text{ is diagonal: } W^T = W \quad \therefore \Rightarrow \theta^T X^T W^T y$$

$$\therefore J(\theta) = \text{const} - 2(y^T W X \theta) + \theta^T (X^T W X) \theta$$

$$\therefore \nabla J(\theta) = -2 \cdot (y^T W X)^T + 2 X^T W X \theta = 0$$

$$\Rightarrow X^T W X \theta = X^T W^T y = X^T W y \quad [\text{since } W^T = W]$$

$$\therefore \text{if } (X^T W X) \text{ is invertible } \Rightarrow \hat{\theta} = (X^T W X)^{-1} \cdot X^T W y \quad \square$$

- (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

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

In this model, each  $y_n$  is drawn from a normal distribution with mean  $\theta^T x_n$  and variance  $\sigma_n^2$ . The  $\sigma_n^2$  are **known**. Write the log likelihood of this model as a function of  $\theta$ . [3 points]  $\rightarrow L(y_n; x_n, \theta)$

$$\begin{aligned} l(\theta) &= \log\left(\prod_{n=1}^N P(y_n | x_n, \theta)\right) \quad (\because \text{independently drawn}) \\ &= \sum_{n=1}^N \log P(y_n | x_n, \theta) = \sum_{n=1}^N \left[ \log\left(\frac{1}{\sqrt{2\pi\sigma_n^2}}\right) + \log\left(\exp\left(-\frac{(y_n - \theta^T x_n)^2}{2\sigma_n^2}\right)\right) \right] \\ &= \sum_{n=1}^N \left[ -\log(\sqrt{2\pi\sigma_n^2}) - \frac{(y_n - \theta^T x_n)^2}{2\sigma_n^2} \right] \end{aligned}$$

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

We want  $\operatorname{argmax}_{\theta} l(\theta)$ , same as finding  $\theta$  that minimize  $(-l(\theta))$

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

since  $\sigma_n^2$  is known.

$\log \sqrt{2\pi\sigma_n^2}$  is known/fixed,  $\ominus$

$\therefore$  minimizing  $(-l(\theta))$  is the same as minimizing:

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

If we let  $\boxed{\frac{1}{2\sigma_n^2} = w_n} \forall n$ , then  $\sum_{n=1}^N w_n \cdot (y_n - \theta^T x_n)^2 = J(\theta)$ .

So it's exactly the same as minimizing  $J(\theta)$ , therefore the answer will be the same.

(Blank page provided for your work)

$$y - X\theta \quad (1 \times N) \cdot (N \times N) \cdot (N \times 1)$$
$$= \begin{pmatrix} y_1 - x_1^T \theta \\ \vdots \\ y_N - x_N^T \theta \end{pmatrix}^T \cdot \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ & & 0 & w_N \end{pmatrix}$$

$$= (y_1 - x_1^T \theta \quad \dots) \begin{pmatrix} \phantom{y_1 - x_1^T \theta} \\ \phantom{y_1 - x_1^T \theta} \\ \phantom{y_1 - x_1^T \theta} \end{pmatrix} \begin{pmatrix} y_1 - x_1^T \theta \\ \phantom{y_1 - x_1^T \theta} \\ \phantom{y_1 - x_1^T \theta} \end{pmatrix}$$

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

$$\begin{aligned} & \cancel{(\theta^T X^T - y^T)} \\ & y^T - X^T \theta^T \end{aligned}$$