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)

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 $0 \leq p(x) \leq 1$ for all x.

False: the value of the probabilistic density function does not need to be smaller than 1. It's just that the area under the curve equals 1.

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

False: kNN is a non-linear classifier where the decision boundaries resemble Voronoi diagrams.

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

True: Logistic Regression is a probabilistic model and we use the binary cross-entropy loss which is derived from MLE.

(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: Even with a classifier which always predicts "cancer-free", one could get 99% 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. It is possible that the classifier is overfitted.

(f) **(3 pts)** A decision tree is learned by minimizing information gain.
False: Nope, maximizing information gain or minimizing entropy.

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?

SGD updates the model after seeing one example, while GD updates the model after computing the gradient using the entire dataset. SGD usually takes more iterations to converge although each iteration takes less time.

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

Development set gives us a good estimate of the model performance on unseen examples.

(c) **(3 pts)** Describe the differences between linear regression and logistic regression (in less than two sentences). Logistic regression predicts the probability $P(y = 1 \mid x)$; therefore the output value is restricted in [0,1], while the output of linear regression can be any real value.

(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, being a probabilistic model can be used for predicting the probability of an event.

(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 $sign(w \cdot x)$.

Lets define the class labels $y \in \{1, +1\}$ and the matrix X such that each of the n rows is one of the n dimensional feature vectors. Then we want to find a w such that $sgn(Xw) = y$. We know that if $Xw = y$ then $sign(Xw) = y$. Since X is composed of linearly independent rows, we can invert X to obtain $w = X^{-1}y$. Therefore we can construct a linear classifier that separates all n points. Interestingly, if we add an additional constant term to the features, we can separate $n + 1$ linearly independent points in n dimensions.

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.

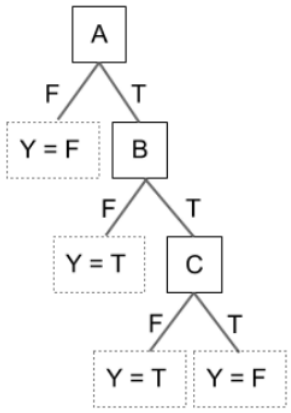
(a) We will use the dataset below to learn a decision tree which predicts the output Y, given by the binary values of A, B, C.

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

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

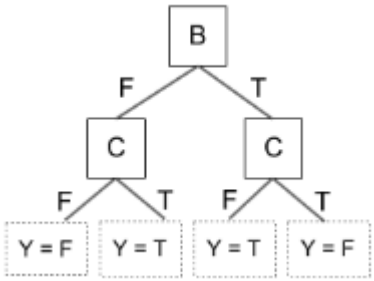
$$H(Y) = -\left[\frac{2}{4}\log(\frac{2}{4}) + \frac{2}{4}\log(\frac{2}{4})\right]$$

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



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.

although we get better information gain by splitting on A, Y is just a function of B and C i.e. $Y = B$ XOR C, hence the best tree is:



(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?
Using the entropy criterion we can determine the better split:

$$H(S_1) = -[\frac{3}{4}\log(\frac{3}{4}) + \frac{1}{4}\log(\frac{1}{4})]$$

$$H(S_2) = -[\frac{3}{4}(\frac{1}{3}\log(\frac{1}{3}) + \frac{2}{3}\log(\frac{2}{3})) + \frac{1}{4}(\frac{1}{1}\log(1))] = -[\frac{3}{4}(\frac{1}{3}\log(\frac{1}{3}) + \frac{2}{3}\log(\frac{2}{3}))]$$

As $H(S_1) > H(S_2)$, S_2 is the better split.

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)

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?

Assume that the model digests the data samples in the given order, it will update the parameters at points 2,4,5,6.
When $y(w^T x) \leq 0$, the model is making a mistake. Then there will be an update.
(If you update only when $y(w^T x) < 0$, you will get answer: 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 :

$w \leftarrow w + \eta y x$, where $\eta \geq \frac{-y(w^T x)}{||x||^2}$

Prove: Let w_i be the weight before making the mistake and w_{i+1} be the updated weight.
We want $y w_i^T x < 0$ but $y w_{i+1}^T x \geq 0$
$w_{i+1} = w_i + \eta y x$
$y w_{i+1}^T x = y w_i^T x + \eta y^2 ||x||^2 \geq 0$
$\eta \geq \frac{-y(w_i^T x)}{||x||^2}$
(suggesting any η satisfying this condition will get full scores.)

(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.
Add one more feature $x_1 \cdot x_2$, then the data is linear separable.
(There are many possible solutions).

(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

$w_0 = w_1 = w_2 = 1$ is one possible solution. This is not the only one.
Any solutions are correct when $y(w_0 + w_1 x_1 + w_2 x_2) > \gamma$, where $\gamma \geq 0$

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

When $w_0 = w_1 = w_2 = 1$, the model has a largest margin, and the distance between closest point to the separating hyper-plane is $\gamma = \frac{1}{\sqrt{2}}$

5 Logistic Regression[19 pts]

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

$$P(Y = 1|X, w_0, w_1, w_2) = \sigma(w_0 + w_1X_1 + w_2X_2)$$

(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 $x = (1, 2)$?

$$\begin{aligned}
\exp(-z) &= \exp\{\ln(4) - \ln(2) + 2\ln(3)\} \\
&= \exp(\ln(4)) \exp(\ln(2^{-1})) \exp(\ln(3^2)\} \\
&= 4 \times 1/2 \times 9 = 18 \\
\sigma(z) &= \frac{1}{1 + \exp(-z)} \\
&= \frac{1}{1+18} = \frac{1}{19}
\end{aligned}$$

As $P(Y = 1|X, w_0, w_1, w_2) = \sigma(Z) < 1/2$, the prediction is $y = -1$.

(b) **(6 pts)** Is logistic regression a linear or non-linear classifier? Prove your answer.
Given x, we predict $y = 1$ if $P(y = 1|x, w) = \sigma(w^T x) \geq 1/2$. This reduces to $w^T x \geq 0$ which is a linear classifier.

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

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

Derive its gradient.

Solution:

Let's denote

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

Recall that we have $\sigma\prime(z) = \sigma(z)(1 - \sigma(z))$. Thus, we have

$$\begin{aligned}
\nabla J(w) &= \sum_{i=1}^{m} \left(\frac{y_i}{\sigma(w^T x_i)} \sigma\prime(w^T x_i) + \frac{1 - y_i}{1 - \sigma(w^T x_i)} \sigma\prime(1 - w^T x_i) \right) \\
&= \sum_{i=1}^{m} \left(\frac{y_i}{\sigma(w^T x_i)} \sigma(w^T x_i)(1 - \sigma(w^T x_i)) x_i - \frac{1 - y_i}{1 - \sigma(w^T x_i)} \sigma(w^T x_i)(1 - \sigma(w^T x_i)) x_i \right) \\
&= \sum_{i=1}^{m} \left(y_i (1 - \sigma(w^T x_i)) x_i - (1 - y_i) \sigma(w^T x_i) x_i \right) \\
&= \sum_{i=1}^{m} \left(y_i - \sigma(w^T x_i) \right) x_i
\end{aligned}$$