# 10-701/15-781 Machine Learning

# Mid-term Exam Solution

Your Name: ______________________________

Your Andrew ID: _________________________

# 1 True or False (Give one sentence explanation) (20%)

1. (F) For a continuous random variable $x$ and its probability distribution function $p(x)$, it holds that $0 \leq p(x) \leq 1$ for all $x$.

2. (F) Decision tree is learned by minimizing information gain.

3. (F) Linear regression estimator has the smallest variance among all unbiased estimators.

4. (T) The coefficients $\alpha$ assigned to the classifiers assembled by AdaBoost are always non-negative.

5. (F) Maximizing the likelihood of logistic regression model yields multiple local optimums.

6. (F) No classifier can do better than a naive Bayes classifier if the distribution of the data is known.

7. (F) The back-propagation algorithm learns a globally optimal neural network with hidden layers.

8. (F) The VC dimension of a line should be at most 2, since I can find at least one case of 3 points that cannot be shattered by any line.

9. (F) Since the VC dimension for an SVM with a Radial Base Kernel is infinite, such an SVM must be worse than an SVM with polynomial kernel which has a finite VC dimension.

10. (F) A two layer neural network with linear activation functions is essentially a weighted combination of linear separators, trained on a given dataset; the boosting algorithm built on linear separators also finds a combination of linear separators, therefore these two algorithms will give the same result.

## 2 Linear Regression (10%)

We are interested here in a particular 1-dimensional linear regression problem. The dataset corresponding to this problem has $n$ examples $(x_1; y_1), \ldots, (x_n; y_n)$ where $x_i$ and $y_i$ are real numbers for all $i$. Let $\mathbf{w}^* = [w_0^*, w_1^*]^T$ be the least squares solution we are after. In other words, $\mathbf{w}^*$ minimizes

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2.$$

You can assume for our purposes here that the solution is unique.

1. (5%) Check each statement that must be true if $\mathbf{w}^* = [w_0^*, w_1^*]^T$ is indeed the least squares solution.

   $$\begin{array}{ll}
   \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)y_i = 0 & (\quad) \\
   \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(y_i - \bar{y}) = 0 & (\quad) \\
   \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) = 0 & (**) \\
   \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(w_0^* + w_1^* x_i) = 0 & (**)
   \end{array}$$

   where $\bar{x}$ and $\bar{y}$ are the sample means based on the same dataset. (hint: take the derivative of $J(\mathbf{w})$ with respect to $w_0^*$ and $w_1^*$)

   (sol.) Taking the derivative with respect to $w_1$ and $w_0$ gives us the following conditions of optimality

   $$\frac{\partial}{\partial w_0}J(\mathbf{w}) = \frac{2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) = 0$$
   $$\frac{\partial}{\partial w_1}J(\mathbf{w}) = \frac{2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i = 0$$

   This means that the prediction error $(y_i - w_0 - w_1 x_i)$ does not co-vary with any linear function of the inputs (has a zero mean and does not co-vary with the inputs). $(x_i - \bar{x})$ and $(w_0^* + w_1^* x_i)$ are both linear functions of inputs.

2. (5%) There are several numbers (statistics) computed from the data that we can use to estimate $\mathbf{w}^*$. There are

   $$\begin{array}{ll}
   \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i & (\quad) \\
   \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i & (\quad) \\
   C_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 & (**) \\
   C_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) & (**) \\
   C_{yy} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 & (\quad)
   \end{array}$$

   Suppose we only care about the value of $w_1^*$. We'd like to determine $w_1^*$ on the basis of ONLY two numbers (statistics) listed above. Which two numbers do we need for this? (hint: use the answers to the previous question) (sol.) We need $C_{xx}$ (spread of $x$) and $C_{xy}$ (linear dependence between $x$ and y). No justification was necessary as these basic points have appeared in the course. If we want to derive these more mathematically, we can, for example, look at one of the answers to the previous question:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) = 0,$$

which we can rewritte as

$$\left[\frac{1}{n}\sum_{i=1}^{n} y_i(x_i - \bar{x})\right] - w_0^*\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})\right] - w_1^*\left[\frac{1}{n}\sum_{i=1}^{n} x_i(x_i - \bar{x})\right] = 0$$

By using the fact that $1/n\sum_i(x_i - \bar{x}) = 0$ we see that

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} y_i(x_i - \bar{x}) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = C_{xy}\\ \frac{1}{n}\sum_{i=1}^{n} x_i(x_i - \bar{x}) &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x}) = C_{xx}\end{aligned}$$

Substituting these back into our equation above gives

$$C_{xy} - w_1^* C_{xx} = 0$$

## 3 AdaBoost (15%)

Consider building an ensemble of decision stumps $G_m$ with the AdaBoost algorithm,

$$f(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right).$$

Figure 1 dispalys a few labeled point in two dimensions as well as the first stump we have chosen. A stump predicts binary ±1 values, and depends only on one coordinate value (the split point). The little arrow in the figure is the normal to the stump decision boundary indicating the positive side where the stump predicts +1. All the points start with uniform weights.

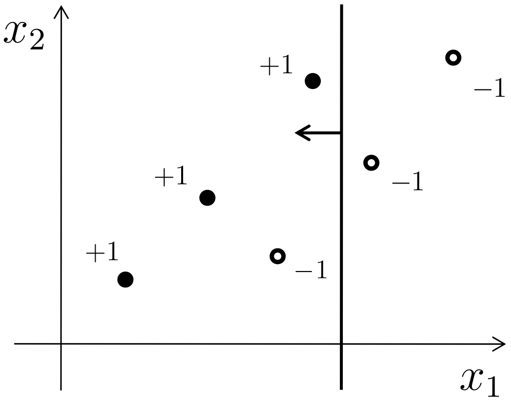


Figure 1: Labeled points and the first decision stump. The arrow points in the positive direction from the stump decision boundary.

1. (5%) Circle all the point(s) in Figure 1 whose weight will increase as a result of incorporating the first stump (the weight update due to the first stump).

   (sol.) The only misclassified negative sample.

2. (5%) Draw in the same figure a possible stump that we could select at the next boosting iteration. You need to draw both the decision boundary and its positive orientation.

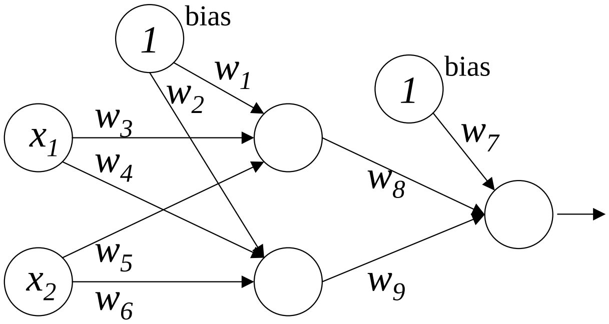
   (sol.) The second stump will also be a vertical split between the second positive sample (from left to right) and the misclassified negative smaple, as drawn in the figure.

3. (5%) Will the second stump receive higher coefficient in the ensemble than the first? In other words, will $\alpha_2 > \alpha_1$? Briefly explain your answer. (no calculation should be necessary).

   (sol.) $\alpha_2 > \alpha_1$ because the point that the second stump misclassifies will have a smaller relative weight since it is classified correctly by the first stump.

# 4 Neural Nets (15%)

Consider a neural net for a binary classification which has one hidden layer as shown in the figure. We use a linear activation function $h(z) = cz$ at hidden units and a sigmoid activation function $g(z) = \frac{1}{1+e^{-z}}$ at the output unit to learn the function for $P(\mathbf{y} = 1|\mathbf{x}, \mathbf{w})$ where $\mathbf{x} = (x_1, x_2)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_9)$.



1. (5%) What is the output $P(\mathbf{y} = 1 \mid \mathbf{x}, \mathbf{w})$ from the above neural net? Express it in terms of $x_i, c$ and weights $w_i$. What is the final classification boundary?

   (sol.)

$$\begin{aligned} & g(w_7 + w_8 h(w_1 + w_3 x_1 + w_5 x_2) + w_9 h(w_2 + w_4 x_1 + w_6 x_2)) \\ = & \frac{1}{1 + \exp(-(w_7 + c w_8 w_1 + c w_9 w_2 + (c w_8 w_3 + c w_9 w_4) x_1 + (c w_8 w_5 + c w_9 w_6) x_2))} \end{aligned}$$

   The classification boundary is :

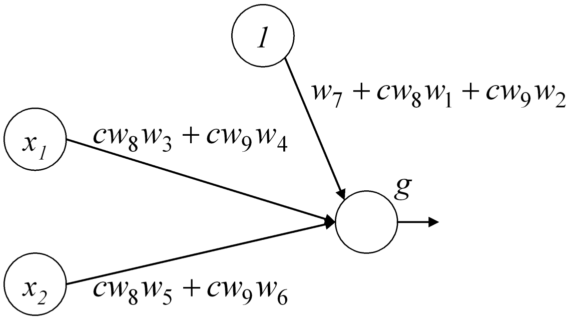
$$w_7 + c w_8 w_1 + c w_9 w_2 + (c w_8 w_3 + c w_9 w_4) x_1 + (c w_8 w_5 + c w_9 w_6) x_2 = 0$$

2. (5%) Draw a neural net with no hidden layer which is equivalent to the given neural net, and write weights $\tilde{\mathbf{w}}$ of this new neural net in terms of $c$ and $w_i$.

   (sol.)

3. (5%) Is it true that any multi-layered neural net with linear activation functions at hidden layers can be represented as a neural net without any hidden layer? Briefly explain your answer.

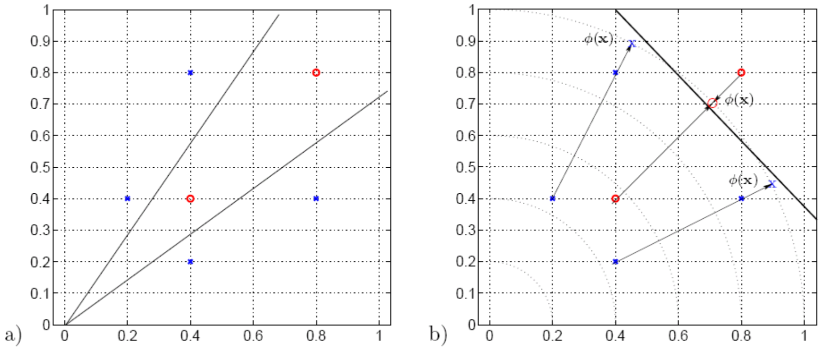
   (sol.) Yes. If linear activation functions are used for all the hidden units, output from hidden units will be written as linear combination of input features. Since these intermediate output serves as input for the final output layer, we can always find an equivalent neural net which does not have any hidden layer as seen in the example above.


$1$
$x_1$
$x_2$
$w_7 + cw_8 w_1 + cw_9 w_2$
$cw_8 w_3 + cw_9 w_4$
$cw_8 w_5 + cw_9 w_6$
$g$

# 5 Kernel Method (20%)

Suppose we have six training points from two classes as in Figure (a). Note that we have four points from class 1: $(0.2, 0.4), (0.4, 0.8), (0.4, 0.2), (0.8, 0.4)$ and two points from class 2: $(0.4, 0.4), (0.8, 0.8)$. Unfortunately, the points in Figure (a) cannot be separated by a linear classifier. The kernel trick is to find a mapping of $\mathbf{x}$ to some feature vector $\phi(\mathbf{x})$ such that there is a function $K$ called kernel which satisfies $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$. And we expect the points of $\phi(\mathbf{x})$ to be linearly separable in the feature space. Here, we consider the following normalized kernel:

$$K(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{||\mathbf{x}^T|| ||\mathbf{x}'||}$$



1. (5%) What is the feature vector $\phi(\mathbf{x})$ corresponding to this kernel? Draw $\phi(\mathbf{x})$ for each training point $\mathbf{x}$ in Figure (b), and specify from which point it is mapped.

$$\phi(\mathbf{x}) = \frac{\mathbf{x}}{||\mathbf{x}||}$$

2. (5%) You now see that the feature vectors are linearly separable in the feature space. The maximum-margin decision boundary in the feature space will be a line in $\mathbb{R}^2$, which can be written as $w_1 x + w_2 y + c = 0$. What are the values of the coefficients $w_1$ and $w_2$? (Hint: you don't need to compute them.)

   (sol.)

$$(w_1, w_2) = (1, 1)$$

3. (3%) Circle the points corresponding to support vectors in Figure (b).

4. (7%) Draw the decision boundary in the original input space resulting from the normalized linear kernel in Figure (a). Briefly explain your answer.

## 6 VC Dimension and PAC Learning(10%)

The VC dimension, $VC(H)$, of hypothesis space $H$ defined over instance space $X$ is the size of the largest largest number of points (in some configuration) that can be shattered by $H$. Suppose with probability $(1-\delta)$, a PAC learner outputs a hypothesis within error $\epsilon$ of the best possible hypothesis in $H$. It can be shown that the lower bound on the number of training examples $m$ sufficient for successful learning, stated in terms of $VC(H)$ is

$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon)).$$

Consider a learning problem in which $X = \mathcal{R}$ is the set of real numbers, and the hypothesis space is the set of intervals $H = \{(a < x < b) | a, b \in \mathcal{R}\}$. Note that the hypothesis labels points inside the interval as positive, and negative otherwise.

1. (5%) What is the VC dimension of $H$?

   (sol.) $VC(H) = 2$. Suppose we have two points $x_1$ and $x_2$, and $x_1 < x_2$. They can always be shattered by $H$, no matter how they are labled.

   (a) if $x_1$ positive and $x_2$ negative, choose $a < x_1 < b < x_2$;

   (b) if $x_1$ negative and $x_2$ positive, choose $x_1 < a < x_2 < b$;

   (c) if both $x_1$ and $x_2$ positive, choose $a < x_1 < x_2 < b$;

   (d) if both $x_1$ and $x_2$ negative, choose $a < b < x_1 < x_2$;

   However, if we have three points $x_1 < x_2 < x_3$ and if they are labeled as $x_1$ (positive) $x_2$ (negative) and $x_3$ (positive), then they cannot be shattered by $H$.

2. (5%) What is the probability that a hypothesis consistent with $m$ examples will have error at least $\epsilon$?

   (sol.) Use the above result. Substitute $VC(H) = 2$ into the inequality

   $$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8 * 2\log_2(13/\epsilon)),$$

   we have

   $$\begin{aligned} \epsilon m &\geq 4\log_2(2/\delta) + 8 * 2\log_2(13/\epsilon) \\ \epsilon m - 16\log_2(13/\epsilon) &\geq 4\log_2(2/\delta) \\ \frac{2^{\epsilon m/4}}{(13/\epsilon)^4} &\geq 2/\delta \\ \delta &\geq \frac{\left(\frac{13}{\epsilon}\right)^4}{2^{\epsilon m/4-1}} \end{aligned}$$

## 7 Logistic Regression (10%)

We consider the following models of logistic regression for a binary classification with a sigmoid function $g(z) = \frac{1}{1+e^{-z}}$:

- Model 1: $P(Y = 1 \mid X, w_1, w_2) = g(w_1 X_1 + w_2 X_2)$
- Model 2: $P(Y = 1 \mid X, w_1, w_2) = g(w_0 + w_1 X_1 + w_2 X_2)$

We have three training examples:

$$\begin{array}{lll} x^{(1)} = [1,1]^T & x^{(2)} = [1,0]^T & x^{(3)} = [0,0]^T \\ y^{(1)} = 1 & y^{(2)} = -1 & y^{(3)} = 1 \end{array}$$

1. (5%) Does it matter how the third example is labeled in Model 1? i.e., would the learned value of $\mathbf{w} = (w_1, w_2)$ be different if we change the label of the third example to -1? Does it matter in Model 2? Briefly explain your answer. (Hint: think of the decision boundary on 2D plane.)

   (sol.) It does not matter in Model 1 because $x^{(3)} = (0,0)$ makes $w_1 x_1 + w_2 x_2$ always zero and hence the likelihood of the model does not depend on the value of $\mathbf{w}$. But it does matter in Model 2.

2. (5%) Now, suppose we train the logistic regression model (Model 2) based on the $n$ training examples $x^{(1)}, \ldots, x^{(n)}$ and labels $y^{(1)}, \ldots, y^{(n)}$ by maximizing the penalized log-likelihood of the labels:

   $$\sum_i \log P(y^{(i)} \mid x^{(i)}, \mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||^2 = \sum_i \log g(y^{(i)} \mathbf{w}^T x^{(i)}) - \frac{\lambda}{2}||\mathbf{w}||^2$$

   For large $\lambda$ (strong regularization), the log-likelihood terms will behave as linear functions of $\mathbf{w}$.

   $$\log g(y^{(i)} \mathbf{w}^T x^{(i)})) \approx \frac{1}{2} y^{(i)} \mathbf{w}^T x^{(i)}$$

   Express the penalized log-likelihood using this approximation (with Model 1), and derive the expression for MLE $\hat{\mathbf{w}}$ in terms of $\lambda$ and training data $\{x^{(i)}, y^{(i)}\}$. Based on this, explain how $\mathbf{w}$ behaves as $\lambda$ increases. (We assume each $x^{(i)} = (x_1^{(i)}, x_2^{(i)})^T$ and $y^{(i)}$ is either 1 or -1 )

   (sol.)

   $$\begin{aligned} \log l(\mathbf{w}) &\approx \sum_i \frac{1}{2} y^{(i)} \mathbf{w}^T x^{(i)} - \frac{\lambda}{2}||w||^2 \\ \frac{\partial}{\partial w_1} \log l(\mathbf{w}) &\approx \frac{1}{2} \sum_i y^{(i)} x_1^{(i)} - \lambda w_1 = 0 \\ \frac{\partial}{\partial w_2} \log l(\mathbf{w}) &\approx \frac{1}{2} \sum_i y^{(i)} x_2^{(i)} - \lambda w_2 = 0 \\ \therefore \quad \mathbf{w} &= \frac{1}{2\lambda} \sum_i y^{(i)} \mathbf{x}^{(i)} \end{aligned}$$