CIS 520, Machine Learning, Fall 2017: Assignment 5
Due: Monday, October 30th, 11:59pm
[100 points]

Instructions. Please write up your responses to the following problems clearly and concisely. We encourage you to write up your responses using LaTeX; we have provided a LaTeX template, available on Canvas, to make this easier. **Submit your answers in PDF form to Canvas. We will not accept paper copies of the homework.**

Collaboration. You are allowed and encouraged to work together. You may discuss the homework to understand the problem and reach a solution in groups up to size **two students.** However, *each student must write down the solution independently, and without referring to written notes from the joint session.* **In addition, each student must write on the problem set the names of the people with whom you collaborated.** You must understand the solution well enough in order to reconstruct it by yourself. (This is for your own benefit: you have to take the exams alone.)

1 Perceptron vs. Winnow [25 points]

For online binary classification problems, you saw the Perceptron algorithm in class. Another algorithm that is used for such problems is the *Winnow* algorithm, which also maintains a linear classification model $\mathbf{w}_t$, but makes *multiplicative updates* to $\mathbf{w}_t$ rather than additive ones (such multiplicative updates now play an important role in many modern optimization algorithms). In this case, the weight vectors $\mathbf{w}_t$ always have positive entries that add up to 1:

Algorithm **Winnow**

Learning rate parameter $\eta > 0$
Initial weight vector $\mathbf{w}_1 = (\frac{1}{d}, \ldots, \frac{1}{d}) \in \mathbb{R}^d$
For $t = 1, \ldots, T$:
- Receive instance $\mathbf{x}_t \in \mathbb{R}^d$
- Predict $\widehat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$
- Receive true label $y_t \in \{\pm 1\}$
- Update: If $\widehat{y}_t \neq y_t$ then

  For each $i \in \{1, \ldots, d\}$: $\quad w_{t+1,i} \leftarrow \dfrac{w_{t,i} \exp(\eta\, y_t x_{t,i})}{Z_t}$

  where $Z_t = \sum_{j=1}^{n} w_{t,j} \exp(\eta\, y_t x_{t,j})$

  else

  $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t$

For examples that are linearly separable by a non-negative weight vector, the Winnow algorithm is known to have the following mistake bound:

Theorem (Winnow mistake bound). *Suppose that the examples seen in T trials are linearly separable by a non-negative weight vector, i.e. that there exists a weight vector $\mathbf{u} \in \mathbb{R}^d_+$ and $\gamma > 0$ such that*

$$y_t(\mathbf{u}^\top \mathbf{x}_t) \geq \gamma \quad \text{for all } t \in \{1, \ldots, T\}.$$

Also suppose $\|\mathbf{x}_t\|_\infty \leq R_\infty$ for all t. If $\|\mathbf{u}\|_1$, γ, and R_∞ are known, then one can select the learning rate parameter η in a way that the number of mistakes in the T trials is at most

$$2\left(\frac{R_\infty^2 \|\mathbf{u}\|_1^2}{\gamma^2}\right)\ln(d)\,.$$

(a) Sparse target vector u, dense feature vectors $\mathbf{x}_t$. [10 points]
Suppose you are in a setting with high-dimensional features (large d), and that all features are of roughly constant magnitude; for simplicity, suppose $\mathbf{x}_t \in \{\pm 1\}^d$ for all t. Suppose you are told that the examples in T trials are linearly separable by a sparse weight vector $\mathbf{u} \in \{0, 1\}^d$ which has only $k \ll d$ non-zero entries, and that you are given $\gamma > 0$ such that $y_t(\mathbf{u}^\top \mathbf{x}_t) > \gamma$ for all t. Calculate upper bounds on the numbers of mistakes that would be made by both Perceptron and Winnow. Which algorithm would be a better choice here?

(b) Dense target vector u, sparse feature vectors $\mathbf{x}_t$. [10 points]
Suppose you are in a setting with high-dimensional features (large d), and that the feature vectors are sparse; for simplicity, suppose $\mathbf{x}_t \in \{0, -1, +1\}^d$ for all t and that each $\mathbf{x}_t$ has $k \ll d$ non-zero entries. Suppose you are told the examples in T trials are linearly separable by a dense weight vector $\mathbf{u} \in \mathbb{R}^d_+$ with $\|\mathbf{u}\|_1 = d$ and $\|\mathbf{u}\|_2 \leq 2\sqrt{d}$, and that you are given $\gamma > 0$ such that $y_t(\mathbf{u}^\top \mathbf{x}_t) > \gamma$ for all t. Calculate upper bounds on the numbers of mistakes that would be made by both Perceptron and Winnow. Which algorithm would be a better choice here?

(c) If your problem has non-negative feature vectors $\mathbf{x}_t \in \mathbb{R}^d_+$, is the Winnow algorithm a meaningful choice? Why or why not? **[5 points]**

2 Multiclass Boosting [25 points]

In this problem you will analyze the AdaBoost.M1 algorithm, a multiclass extension of AdaBoost. Given a training sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$, where x_i are instances in some instance space $\mathcal{X}$ and y_i are multiclass labels that take values in $\{1, \ldots, K\}$, the algorithm maintains weights $D_t(i)$ over the examples (x_i, y_i) as in AdaBoost, and on round t, gives the weighted sample (S, D_t) to the weak learner. The weak learner returns a multiclass classifier $h_t : \mathcal{X} \to \{1, \ldots, K\}$ with weighted error less than $\frac{1}{2}$; here the weighted error of h_t is measured as

$$\mathrm{er}_t = \sum_{i=1}^{m} D_t(i) \cdot \mathbf{1}(h_t(x_i) \neq y_i)\,.$$

Note that the assumption on the weak classifiers is stronger here than in the binary case, since we require the weak classifiers to do more than simply improve upon random guessing (there are other multiclass boosting

algorithms that allow for weaker classifiers; you will analyze the simplest case here). For convenience, we will encode the weak classifier h_t as $\widetilde{h}_t : \mathcal{X} \to \{\pm 1\}^K$, where

$$\widetilde{h}_{t,k}(x) = \begin{cases} +1 & \text{if } h_t(x) = k \\ -1 & \text{otherwise.} \end{cases}$$

In other words, $\widetilde{h}_t(x)$ is a K-dimensional vector that contains $+1$ in the position of the predicted class for x and -1 in all other $(K-1)$ positions. On each round, AdaBoost.M1 re-weights examples such that examples misclassified by the current weak classifier receive higher weight in the next round. At the end, the algorithm combines the weak classifiers h_t via a weighted majority vote to produce a final multiclass classifier H:

Algorithm **AdaBoost.M1**

Inputs: Training sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \{1, \ldots, K\})^m$
Number of iterations T

Initialize: $D_1(i) = \frac{1}{m} \quad \forall i \in [m]$

For $t = 1, \ldots, T$:

- Train weak learner on weighted sample (S, D_t); get weak classifier $h_t : \mathcal{X} \to \{1, \ldots, K\}$
- Set $\alpha_t \leftarrow \dfrac{1}{2} \ln\left(\dfrac{1 - \mathrm{er}_t}{\mathrm{er}_t}\right)$
- Update:

$$D_{t+1}(i) \leftarrow \frac{D_t(i) \exp(-\alpha_t \widetilde{h}_{t,y_i}(x_i))}{Z_t}$$

where $Z_t = \sum_{j=1}^m D_t(j) \exp(-\alpha_t \widetilde{h}_{t,y_j}(x_j))$

Output final hypothesis:

$$H(x) \in \arg\max_{k \in \{1,\ldots,K\}} \underbrace{\sum_{t=1}^T \alpha_t \widetilde{h}_{t,k}(x)}_{F_{T,k}(x)}$$

You will show, in five parts below, that if all the weak classifiers have error er_t at most $\frac{1}{2} - \gamma$, then after T rounds, the training error of the final classifier H, given by

$$\mathrm{er}_S[H] = \frac{1}{m} \sum_{i=1}^m \mathbf{1}(H(x_i) \neq y_i)\,,$$

is at most $e^{-2T\gamma^2}$ (which means that for large enough T, the final error $\mathrm{er}_S[H]$ can be made as small as desired).

(a) [5 points] Show that

$$D_{T+1}(i) \;=\; \frac{\frac{1}{m}\, e^{-F_{T,y_i}(x_i)}}{\prod_{t=1}^T Z_t}\,.$$

(b) [5 points] Show that

$$\mathbf{1}(H(x_i) \neq y_i) \;\leq\; \mathbf{1}\big(F_{T,y_i}(x_i) < 0\big)\,.$$

(*Hint:* Consider separately the two cases $H(x_i) \neq y_i$ and $H(x_i) = y_i$, and note that $\sum_{k=1}^K F_{T,k}(x_i) = -(K-2)\sum_{t=1}^T \alpha_t$.)

(c) [5 points] Show that

$$\mathrm{er}_S[H] \;\leq\; \frac{1}{m}\sum_{i=1}^{m} e^{-F_{T,y_i}(x_i)} \;=\; \prod_{t=1}^{T} Z_t\,.$$

(*Hint:* For the inequality, use the result of part (b) above, and the fact that $\mathbf{1}(u<0) \leq e^{-u}$; for the equality, use the result of part (a) above.)

(d) [5 points] Show that for the given choice of α_t, we have

$$Z_t \;=\; 2\sqrt{\mathrm{er}_t(1-\mathrm{er}_t)}\,.$$

(e) [5 points] Suppose $\mathrm{er}_t \leq \frac{1}{2}-\gamma$ for all t (where $0<\gamma\leq\frac{1}{2}$). Then show that

$$\mathrm{er}_S[H] \;\leq\; e^{-2T\gamma^2}\,.$$

3 Loss-Based Performance Measures [25 points]

1. **Binary classification with asymmetric costs. [5 points]**

   Consider a binary classification problem in which the cost of a false positive is 10 and that of a false negative is 40, so that we have the following loss:

		$\widehat{y}$	
		-1	$+1$
y	-1	0	10
	$+1$	40	0

   Say you have learned a class probability estimation (CPE) model $\widehat{\eta}:\mathcal{X}\to[0,1]$, which given an instance $x\in\mathcal{X}$, estimates the probability of x having a true label $+1$. How would you use this to build a classification model $h:\mathcal{X}\to\{-1,+1\}$ for the above problem? Give your answer as a decision rule that explains when $h(x)$ should be -1 or $+1$ (as a function of $\widehat{\eta}(x)$).

2. **Binary classification with abstain option. [10 points]**

   Consider a binary classification problem in which the classifier is allowed to 'abstain' on instances it is not sure about (such instances could then be sent to a human expert to classify, with some associated cost). So here the true label y can take 2 possible values, -1 and $+1$, while the predicted label $\widehat{y}$ can take 3 possible values: -1, $+1$, and '?' ('abstain'). Suppose that the cost of misclassifying an instance is 1 as in the usual 0-1 loss, and that the cost of abstaining is 0.4, so that we have the following loss:

		$\widehat{y}$		
		-1	$+1$	'?'
y	-1	0	1	0.4
	$+1$	1	0	0.4

   Say you have learned a class probability estimation (CPE) model $\widehat{\eta}:\mathcal{X}\to[0,1]$, which given an instance $x\in\mathcal{X}$, estimates the probability of x having a true label $+1$. How would you use this to build a classification model $h:\mathcal{X}\to\{-1,+1,\text{'?'}\}$ for the above problem? Give your answer as a decision rule that explains when $h(x)$ should be -1, $+1$, or '?' (as a function of $\widehat{\eta}(x)$). Explain your derivation. If the cost of abstaining changes from 0.4 to 0.2, how will your classification model change? Will it abstain more frequently or less frequently?

3. **Multiclass classification. [10 points]**

You are collaborating with a cancer treatment center and are trying to help them predict which patients will respond well to a particular cancer drug. You are given clinical data for patients they have given the drug to in the past; for each such patient, the data contains measurements from the patient's tumor biopsy together with a class label indicating whether the patient was a complete responder (CR) to the drug, a partial responder (PR), or a non-responder (NR). Your goal is to predict the response category (CR, PR, or NR) for new patients based on their tumor biopsy measurements. You are given the following loss for this problem:

		$\widehat{y}$ = NR	$\widehat{y}$ = PR	$\widehat{y}$ = CR
y	NR	0	4	5
	PR	9	0	1
	CR	10	1	0

Here mis-predicting a PR case as CR or vice versa incurs relatively little cost, since in both cases the patient is given the drug. Mis-predicting a PR or CR case as NR is very costly, since in this case a patient who could benefit from the drug does not receive treatment. Mis-predicting an NR case as PR or CR is also costly, since it involves unnecessary expense and side effects for a patient who doesn't benefit from the drug, but is less costly than errors in the other direction.

Suppose that, using the training data provided to you, you have trained a CPE model, and that for 2 new patients with measurement vectors x_1 and x_2, the model estimates the following probabilities for the 3 classes:

$$\text{Patient 1: } \begin{pmatrix} \widehat{\eta}_{\text{NR}}(x_1) \\ \widehat{\eta}_{\text{PR}}(x_1) \\ \widehat{\eta}_{\text{CR}}(x_1) \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.3 \\ 0.1 \end{pmatrix}; \quad \text{Patient 2: } \begin{pmatrix} \widehat{\eta}_{\text{NR}}(x_2) \\ \widehat{\eta}_{\text{PR}}(x_2) \\ \widehat{\eta}_{\text{CR}}(x_2) \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.3 \\ 0.6 \end{pmatrix}.$$

Which response category would you predict for each patient, and how do these categories differ from what you would have predicted under the 0-1 loss? Explain your answers.

4 Learning Theory [25 points]

1. In a supervised learning problem, the true quantity we really want to minimize is
   (a) the training error
   (b) the test error
   (c) the generalization error
   (d) the cross-validation error
   (e) none of the above

2. Which of the following serve to provide an estimate of the generalization error? Pick all that apply.
   (a) training error
   (b) test error
   (c) cross-validation error
   (d) VC-complexity bound
   (e) none of the above

3. Which of the following tasks can VC-complexity bounds be useful for? Pick all that apply.

   (a) obtaining high-confidence bounds on generalization error

   (b) choosing a training set

   (c) choosing folds for cross-validation

   (d) model selection

   (e) none of the above

4. For binary classification with 0-1 loss, an algorithm that always chooses an optimal linear classifier on the training data (with smallest 0-1 error on the training data) is

   (a) never consistent

   (b) consistent for some data distributions

   (c) universally consistent

   (d) an optimal algorithm for 0-1 loss

   (e) none of the above

5. For binary classification with 0-1 loss, which of the following algorithms are universally consistent? Pick all that apply.

   (a) SVM with quadratic kernel and suitably chosen C parameter

   (b) SVM with RBF kernel and suitably chosen C parameter

   (c) Linear logistic regression with L_2 regularization and suitably chosen λ parameter

   (d) Logistic regression with RBF kernel, RKHS regularization, and suitably chosen λ parameter

   (e) none of the above