| **CS446: Machine Learning** | **Fall 2015** |
|---|---|
| October $27^{th}$, 2015 | |

- 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 **four** problems. You need to solve all problems to get 100%.
- The exam ends at 1:45 PM. You have 75 minutes to earn a total of 100 points.
- Answer each question in the space provided. If you need more room, write on the reverse side of the paper and indicate that you have done so.
- **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 (NetID):** (1 Point)

| | | |
|---|---|---|
| Short Questions | | /29 |
| Kernels | | /25 |
| Online Learning | | /25 |
| Decision Trees | | /20 |
| **Total** | | /100 |

**Short Questions** [29 points]

(a) **(5 points)** In the following definition of PAC learning we left a few blank fields. Fill in the blanks by choosing, for each empty field, one of the options given below. Note that under each line defining a blank we provided a small set of options for you to choose from.

| | | | | | |
|---|---|---|---|---|---|
| (a) $\delta$ | (b) $\epsilon$ | (c) $1/\delta$ | (d) $1/\epsilon$ | (e) $1-\delta$ | (f) $1-\epsilon$ |
| (g) $m$ | (h) $n$ | (i) $n\epsilon/\delta$ | | (j) size($\mathbf{H}$) | |
| (k) number of examples | | (l) instance size | | (m) computation time | |
| (n) linear | | (o) polynomial | | (p) exponential | |
| (q) $\frac{1}{2}-\gamma$ | (r) $\frac{1}{2}+\gamma$ | (s) $1-\gamma$ | | | |

A concept class $\mathbf{C}$ defined over the instance space $\mathbf{X}$ (with instances of length $n$) is *strongly* PAC learnable by learner $\mathbf{L}$ using a hypothesis space $\mathbf{H}$ if for all concepts $f \in \mathbf{C}$, for all distributions $\mathbf{D}$ on $\mathbf{X}$, and for all fixed $\delta, \epsilon \in [0,1]$, given a sample of $m$ examples sampled independently according to the distribution $\mathbf{D}$, the learner $\mathbf{L}$ produces with a probability _____ {at least | at most | equal to} _____ {one of (a) to (f)} a hypothesis $g \in \mathbf{H}$ with error _____ {at least | at most | equal to} _____ {one of (a) to (f)} where the _____ {one of (k) to (m)} is _____ {one of (n) to (p)} in _____, _____, _____, and _____ {four of (a) to (j)}.

(b) **(8 points)** Consider the following algorithm $\mathcal{B}$ for functions in $C$. $\mathcal{B}$ has the property that given any polynomial size sample of $m$ labeled examples $\{(x_1, c(x_1)), \ldots (x_m, c(x_m))\}$ according to some $c \in C$, $\mathcal{B}$ outputs a hypothesis $h \in H$ that is incorrect on at most one of the $m$ examples.

Then, the concept class $C$ _____ {PAC learnable | not PAC learnable} .

Justify your answer.

(c) **(8 points)**

In this question we consider the Winnow algorithm.

**(1)** In class we proved that Winnow can learn $k$-monotone disjunctions over a domain of $n$ variables. State the upper bound on the number of examples Winnow will make.

**(2)** We consider a learning problem of the domain $X = \{1, 2, \ldots, N\}^d$. A $d$-dimensional hyper-rectangle over this domain $X$ is a subset $c \subseteq X$ defined by $2d$ values: $1 \leq a_i \leq b_i \leq N$, for $i = 1, \ldots, d$.
The subset is

$$c = \{(x_1, \ldots, x_d) \in X : a_i \leq x_i \leq b_i \forall i = 1, \ldots, d\}.$$

Let RECT denote the class of all such $d$-dimensional hyper-rectangles over $X$.

i. Show that you can use Winnow to learn the concept class RECT. Justify your answer.

ii. What is the mistake bound of your algorithm? Explain.

(d) **(8 points)** We define a set of concepts

$$H = \{sgn(ax^2 + bx + c); a, b, c, \in R\},$$

where $sgn(z)$ is 1 when $z$ is positive, and 0 otherwise. What is the VC dimension of $H$? Prove your claim.

**Kernels** [25 points]

In this question we will define kernels, study some of their properties and develop one specific kerenl.

(a) **(1 point)** Choose one of the options below:
A function $K(x, z)$ is a valid kernel if it corresponds to an _____ in some feature space, between feature representations that correspond to $x$ and $z$.
{"inner(dot) product" | "sum"}

(b) **(12 points)** In the next few questions we guide you to prove the following property of kernels:
*Linear Combination Property:* if $\forall i$, $k_i(x, x')$ are valid kernels, and $c_i > 0$ are constants, then $k(x, x') = \sum_i c_i k_i(x, x')$ is a valid kernel.

i. Given a valid kernel $k_1(x, x')$ and a constant $c > 0$, use the definition in(a) to show that $k(x, x') = ck_1(x, x')$ is also a valid kernel.

ii. Given valid kernels $k_1(x, x')$ and $k_2(x, x')$, use the definition in (a) to show that $k(x, x') = k_1(x, x') + k_2(x, x')$ is also a valid kernel.

iii. Conclude that the Linear Combination Property holds.

(c) **(12 points)** In order to learn functions over sets, we represent sets as feature vectors in an $n$ dimensional space, and define a kernel in this space. Our instances are all subsets of a set $S$, of size $|S| = m$. The $n$ dimensional feature representation of $A \subseteq S$ is:

$$\varphi(A) = (\phi_{U_1}(A), \phi_{U_2}(A), \ldots, \phi_{U_n}(A)),$$

where $U_1, U_2, \ldots, U_n$ are all the subsets of $S$, and the coordinates are defined using the following feature mapping function:

$$\phi_U(A) = \begin{cases} 1, & \text{if } U \subseteq A \\ 0, & \text{otherwise.} \end{cases}$$

That is, $\varphi(A) \in \{0,1\}^n$, where $n = 2^m$.

Let $A, B$ be subsets of $S$. We define the kernel

$$K(A, B) = \varphi(A)^\top \varphi(B).$$

**Show that** $K(A, B)$ can be computed in time that is polynomial in $m$.

**On-Line Learning** [25 points]

In this problem we will consider a few variations of the Perceptron Learning algorithm and their properties. Recall the perceptron algorithm,

```
1: for (x, y) ∈ {x_i, y_i}_{i=1}^T do
2:     if y(w^T x) ≤ 0 then
3:         w' ← w + ηyx
4:         w ← w'
5:     end if
6: end for
7: return w
```

(a) **(10 points)** Suggest a variation of the preceptron update rule (line 3) 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). Your work will result in one of the options below.
Which one? _____
{one of (a) to (d)}

(a) $\eta \geq \frac{1-y(w^\intercal x)}{\|x\|^2+1}$ (b) $\eta \leq \frac{-y(w^\intercal x)}{\|x\|^2+1}$ (c) $\eta \leq \frac{-y(w^\intercal x)}{\|x\|^2}$ (d) $\eta \geq \frac{-y(w^\intercal x)}{\|x\|^2}$

Prove that your proposed variation satisfies the property stated above (that is, show how you derive the learning rate you chose).

(b) **(5 points)** Consider the following variant of perceptron: Instead of *returning* $w$ in line 7, return $\frac{w}{\|w\|}$.
Choose one of the following options and justify your answer: The decision boundary of the variant and the standard perceptron are _____ because:
{same | different}

(c) **(5 points)** Consider the following variant of perceptron: *Following each mistake*, instead of the usual update in line 3, perform the following update,

$$w' \leftarrow \frac{w + \eta y x}{\|w + \eta y x\|}.$$

Choose one of the following options and justify your answer: The decision boundary of the variant and the standard perceptron are _____ because:
{same | different}

(d) **(5 points)** Derive the SGD update for the SVM algorithm, which has the following loss:

$$\|w\|^2 + \sum_{i=1}^{M} \max(0, 1 - y_i(w^\intercal x_i))$$

(disregard the single point where the objective function is not differentiable. )

**Decision Trees** [20 points]

(a) **(10 points)** Let $x$ be a vector of $n$ Boolean variables and let $k \leq n$ be an integer. We define the class $P_k$ of $k$-parity functions over the $n$ variables. A $k$-parity function $f_S$ is defined as follows: a set $S \subseteq \{x_1, \ldots, x_n\}$ is chosen such that $|S| = k$. Let $x = (x_1, x_2, \ldots x_n) \in \{0,1\}^n$. Then $f_S(x) = 1$ iff an odd number of variables in $S$ are set to 1 in $x$.

**Example:** Let $n = 3$ and consider functions in $P_2$. Let $S = \{x_1, x_2\}$. Then $f_S(100) = f_S(101) = f_S(010) = f_S(011) = 1$, and $f_S(000) = f_S(001) = f_S(110) = f_S(111) = 0$.

i. Fix $S \subseteq \{x_1, \ldots, x_n\}$, and assume that you are using ID3 to learn a decision tree from data that is consistent with $f_S$. Consider two variables, $x_1, x_2$ such that $x_1 \in S$ and $x_2 \notin S$. Which of these variables is more likely to be the root of the decision tree for $f_S$? $\underset{\{x_1\ ;\ x_2\}}{\_\_\_\_\_}$

Justify your answer:

ii. Consider $f_S \in P_k$. State the *depth* of the smallest possible consistent decision tree for $f_S$ in terms of $n$ and $k$. Describe the shape of the decision tree for $f_S$.

Justify your answer.

(b) **(10 points)** Assume that you are using an implementation of ID3 that takes an upper bound on the depth of the output decision tree as a parameter. You will use this implementation of ID3 to learn from a dataset $D_{train}$, compute the empirical error, and then evaluate the learned tree on a test set $D_{test}$. You will learn two trees, $T_k$, learned with bound $k$ on the depth, and $T_m$, learned with a depth bound $m$.
Assume $\mathbf{k < m}$. (But note that we say nothing about the size of $k$; it could be a very small number or a very large number).

i. Which tree, $T_k$ or $T_m$, is likely to have *larger* empirical error (that is, error on $D_{train}$)? _____
{$T_k$ ; $T_m$ ; impossible to tell}
Justify your answer:

ii. Which tree, $T_k$ or $T_m$, is likely to have larger error when tested on $D_{test}$? _____
{$T_k$ ; $T_m$ ; impossible to tell}
Justify your answer:

**This page was intentionally left blank.**