CS446: Machine Learning (D)	**Spring 2017**
March 16^{th}, 2017	

- 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%.
- Please check that the exam booklet contains **17** pages, with the appendix at the end.
- 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.
- A list of potentially useful functions has been provided in the appendix at the end.
- **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)

PAC Learning		/25
SVM		/25
Kernel		/25
Decision Trees		/24
Total		/100

1 PAC Learning and VC dimension (25 points)

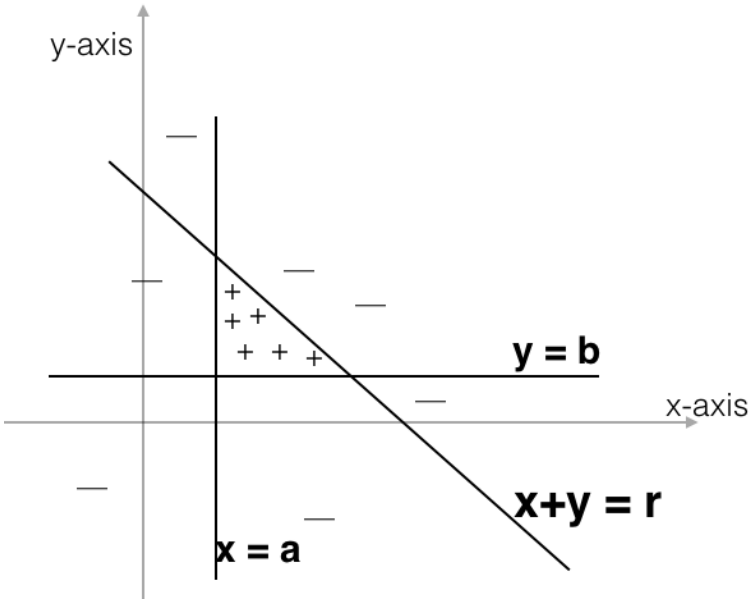
Consider the hypothesis space of "top right facing right angled isosceles triangles", the set of all isosceles triangles with two sides that are parallel to the x and y axes respectively, in the two dimensional plane.
Any such triangle can be created by the intersection of the straight lines $x = a$, $y = b$ and $x + y = r$, where $r > a$ and $r > b$. Formally, this class can be represented as

$$\mathcal{H}_\Delta = \{h_{a,b,r} : a, b, r \in \mathbb{R}, \quad r > a, \quad r > b\}$$

where

$$h_{a,b,r} = \begin{cases} 1 & \text{if } x \geq a \text{ and } y \geq b \text{ and } x + y \leq r \\ -1 & \text{otherwise} \end{cases}$$

That is, each $h_{a,b,r}$ assigns a positive label to any point lying in the interior or on the edge of the triangle formed by the straight lines $x = a$, $y = b$ and $x + y = r$. $h_{a,b,r}$ assigns a negative label to any point lying outside the triangle. Diagrammatically, $h_{a,b,r}$ looks like this



(a) (i) (1 point) The size of the hypothesis space $\mathcal{H}_\Delta$ is (b). Choose one of the following options to fill the blank:

(a) Finite (b) Infinite

(ii) (3 points) Consider the following data set:

2D Points (x,y)	Label
(1, 1)	+1
(1, 2)	+1
(5, 5)	-1
(-2, -1)	-1

We want to choose a hypothesis in $\mathcal{H}_\Delta$ that is consistent with this data set. Circle one of the four triples (a, b, r) below that defines a hypothesis consistent with the given data.

(a) $a = -2$, $b = -1$, $r = 11$ (b) $a = 0$, $b = 0$, $r = 3$

(c) $a = -1$, $b = 1$, $r = 10$ (d) $a = -2$, $b = -1$, $r = 10$

(b)

(b) (i) (5 points) Assume a sample of m points is drawn I.I.D. from some distribution $\mathcal{D}$ and that the labels are provided from some target function $h^*_{a^*,b^*,r^*} \in \mathcal{H}_\Delta$. Describe an algorithm that takes a training sample of m such points and returns a hypothesis $\hat{h}_{a,b,r} \in \mathcal{H}_\Delta$ that is *consistent* with the training sample.

Find the minimum x coordinate, minimum y coordinate, and maximum (x+y) value for positive points

(ii) (2 points) State the running time of your algorithm in terms of m, with justification.

O(m)

(c) (i) (1 point) Explain **briefly**: how does one prove, for a given hypothesis class, that:

$$\text{VC dimension} \geq k$$

When you can shatter k points

(ii) (4 points) Show that : VC dimension of $\mathcal{H}_\Delta \geq 3$.
Show a set of 3 points with all labellings paired with shattering functions

(d) (i) (3 points) Explain **briefly**: how does one prove, for a given hypothesis class, that:

$$\text{VC dimension} = k$$

When you can shatter k points, but cannot shatter any arrangement of $k+1$ points

(ii) (6 points) Consider a subset $\mathcal{H}_{\subset}$ of our original hypothesis class $\mathcal{H}_{\Delta}$, where we fix a and b to be 0. That is:

$$\mathcal{H}_{\subset} = \{h_{a,b,r} \in \mathcal{H}_{\Delta} : a = 0 \text{ and } b = 0\}.$$

Find the VC dimension of $\mathcal{H}_{\subset}$.

VC dimension of this class is 1. Its easy to show that the class can shatter 1 point. For the case of two points, consider any two points in the first quadrant (x_1, y_1) and (x_2, y_2). (We did not choose points from the other quadrants, since they will always be negative, hence, cannot be shattered). Let $r_1 = x_1 + y_1$ and $r_2 = x_2 + y_2$. If $r_1 = r_2$, you cannot shatter because both points will always have the same label. If $r_1 < r_2$, you can never label (x_1, y_1) to be negative, and (x_2, y_2) positive. A similar reasoning for the $r_1 > r_2$ case.

2 Hinge Loss and SVM (25 points)

Given a dataset D $= \{(x_i, y_i)\}, x_i \in R^k, y_i \in= \{-1, +1\}, 1 \leq i \leq N$.

(a) (3 points) Which of the following is the Hinge Loss of the example (x_i, y_i)? (Circle the correct answer).

(A) $max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i)^2$ **(B)** $min(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i)$

(C) $max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i)$ **(D)** $max(0, y_i \mathbf{w}^T \mathbf{x}_i)$

(E) $max(0, y_i - \mathbf{w}^T \mathbf{x}_i)$ **(F)** $max(0, -y_i \mathbf{w}^T \mathbf{x}_i)$

C

In the next few questions we will address formulations of SVMs. Recall the formulation of Hard (Linear) SVM:

$$\min_w \frac{1}{2} w^T w$$
$$\text{s.t } \forall i, y_i w^T x_i \geq 1$$

(b) (4 points) Complete the formulation of soft SVM:

$$\min_{w, \xi_i} \frac{1}{2} w^T w + C \sum_{i \in [1,N]} \xi_i$$
$$\text{s.t } \underline{\quad \forall i, y_i w^T x_i \geq 1 - \xi_i \quad}$$
$$\underline{\quad \forall i, \xi_i \geq 0 \quad}$$

(c) (4 points) Complete: If C = $\underline{\quad \infty \quad}$, soft SVM will behave exactly as hard SVM.

(d) (4 points) In order to reduce over-fitting, one should *decrease*/*increase* the value of C. (circle the correct answer). Briefly justify your answer?

Decrease. It will tolerant more error, and hence increase the regularization strength.

(e) (5 points) Derive the SGD update for the soft SVM algorithm. Write down the exact procedure and update rule. (Hint : First reduce the optimization problem to an unconstrained optimization problem, then apply SGD)

$$\min \frac{1}{2} w^T w + C \sum max(0, 1 - y_i w^T x_i)$$

With some learning rate r, and regularization constant C',

Then for each example (x_i, y_i):

if $max(0, 1 - y_i w^T x_i) == 0$:

$w = w - rw$

else:

$w = w + C' r y_i x_i - rw$

(f) (5 points) We are given the dataset in Figure 1 below, where the positive examples are represented as black circles and negative points as white squares. (The same data is also provided in Table 1 for your convenience).

Recall that the equation of the separating hyperplane is $\hat{y} = w^T x - \theta$.

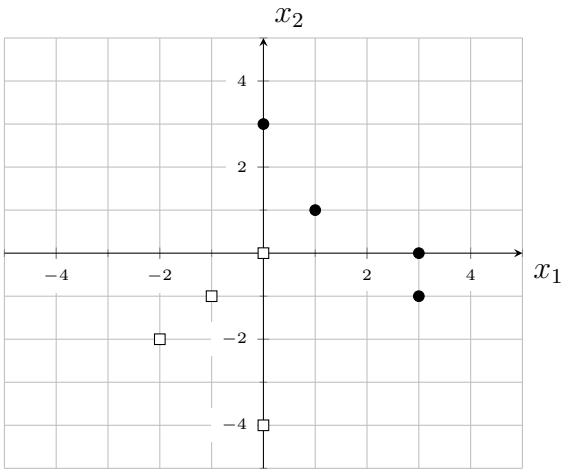


Figure 1: Linear SVM

	Attributes		
	x_1	x_2	y
1	0	0	(-)
2	0	-4	(-)
3	-1	-1	(-)
4	-2	-2	(-)
5	3	0	(+)
6	0	3	(+)
7	1	1	(+)
8	3	-1	(+)

Table 1: The data set S.

(i) **Draw** the hard SVM decision boundary for the dataset in Figure 1. **Write down** the parameter for the learned linear decision function.

W $= (w_1, w_2) =$ (1, 1) . $\theta =$ 1

(ii) **Circle** the support vectors in Figure 1.

Point 1,7,8

3 Perceptron, Kernels, Boosting (25 pts)

In the first four parts of this problem we will address the Perceptron with Margin algorithm and Kernel Perceptron. The final part is about Boosting.
In answering the first two questions, you will choose options from this table.

(a) -1	(b) 0	(c) 1	(d) θ
(e) y_i	(f) $\mathbf{x}_i$	(g) $y_i\mathbf{x}_i$	(h) $-y_i\mathbf{x}_i$
(i) $\|\mathbf{x}'\|_2$	(j) $[x'^2_1, x'^2_2]^T$	(k) $e^{\|\mathbf{x}'\|_2}$	(m) $\max(0, 1 - \|\mathbf{x}'\|_2)$
(n) Do nothing.	(p) Shuffle($\mathcal{S}$)		

(a) (5 pts)

We are given a training sample of size n, $\mathcal{S} = \{(\mathbf{x}'_i, y_i)\}_{i=1}^m$, where $\mathbf{x}'_i \in \mathbb{R}^2$ is the feature vector and $y_i \in \{\pm 1\}$ is the label, $i = 1, 2, \ldots, m$.

We want to train a linear classifier, i.e., $h_{\mathbf{u},\theta}(\mathbf{x}') = \text{sgn}(\mathbf{u}^T\mathbf{x}' - \theta)$, $\mathbf{u} \in \mathbb{R}^2$, $\theta \in \mathbb{R}$, on $\mathcal{S}$ using the **perceptron with margin** algorithm.

Define the augmented weight vector $\mathbf{w} \in \mathbb{R}^3$ and feature vector $\mathbf{x} \in \mathbb{R}^3$ such that $\mathbf{u}^T\mathbf{x}' - \theta \equiv \mathbf{w}^T\mathbf{x}$. (2 pts)

$$\mathbf{w} = [\mathbf{u}^T, \underbrace{\text{(d) } \theta}_{\text{Choose from (a) to (d)}}]^T, \quad \mathbf{x}_i = [\mathbf{x}'^T_i, \underbrace{\text{(a) } -1}_{\text{Choose from (a) to (d)}}]^T, \ \forall i$$

(d) and (a) cannot be flipped here, although it still satisfies $\mathbf{u}^T\mathbf{x}' - \theta \equiv \mathbf{w}^T\mathbf{x}$. The reason is that x is the feature vector, and it remains the same during training. Training is to adjust the weight vector (of course, θ as well) but not to adjust the feature vector.

Complete the perceptron with margin algorithm below.

Algorithm 1 Perceptron algorithm **with margin**

w $\leftarrow$ **0**
for each example $(\mathbf{x}_i, y_i) \in \mathcal{S}$ **do**
  if $y_i(\mathbf{w}^T\mathbf{x}_i) <$ **(c) 1. "with margin"** (Choose from (a) to (c)) **then**
    $\mathbf{w} \leftarrow \mathbf{w} - r \cdot$ (**(h) $-y_i\mathbf{x}_i$. The gradient of hinge loss has this minus sign.**) (Choose from (e) to (h))
  ▷ Learning rate r is a positive constant
  else
    (n) Do nothing. Shuffle should be applied outside this for-loop. (Choose from (n) to (p))
  end if
end for
Return w

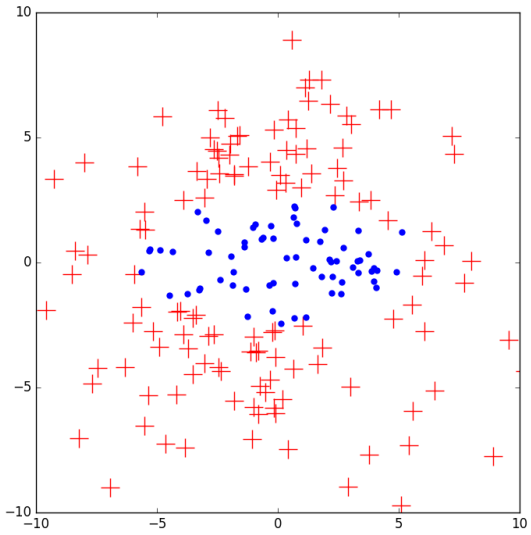


Figure 2: The visualization of $\mathcal{S}$, where the plus/dot signs represent examples of label "+1"/"-1" and the two axes represent the two dimensions of $\mathbf{x}'$.

(b) (6 pts)

(i) The data set $\mathcal{S}$ is given in Fig. 2. Can you directly use the version of the perceptron given above to learn a good hypothesis for this data? **NO**
Choose YES or NO

(ii) Assume that you want to use the primal version of Perceptron to learn a good hypothesis for this data. Suggest a new set of features $\tilde{\mathbf{x}}'$.

$\tilde{\mathbf{x}}' =$ **(j) $[x'^2_1, x'^2_2]^T$; it fits an ellipse. (i) and (k) can only fit circles.**
Choose from (i) to (m)

(c) (4 pts)

We will use now the dual version of Perceptron. Which of the following four options is a correct dual representation of linear classifiers (i.e., $h_{\mathbf{w}}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x})$)? **(iv)**

(i) $\mathbf{w} = w_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + w_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + w_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \ w_1, w_2, w_3 \in \mathbb{R}$

(ii) $\mathbf{w} = \sum_{i=1}^{m} \mathbf{x}_i$

(iii) $\mathbf{w} = \sum_{i=1}^{m} y_i \mathbf{x}_i$

(iv) $\mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i, \ \alpha_i \in \mathbb{R}$

Briefly **justify** your answer.

In the case of Algorithm 1, w is initialized to be the zero vector. Whenever it is updated, a re-scaled $\mathbf{x}_i$ is added to w, so w is a linear combination of $\mathbf{x}_i$'s. Many of you have chosen (iii), but (iii) is a actually constant and you cannot learn anything using it. (i) is a correct representation but not in the dual form. Some of you got points deducted because of mentioning α_i is the number of mistakes, which is not accurate.

(d) (4 pts)

(i) Determine which of the following functions are valid kernels for $\mathbf{x}, \mathbf{z} \in \mathbb{R}^3$. Note that x_i and z_i are the ith dimension of $\mathbf{x}$ and $\mathbf{z}$, respectively. Circle "TRUE" if one is, or "FALSE" otherwise. There could be multiple correct options; choose all of them.

i.	$K(\mathbf{x}, \mathbf{z}) = \mathbf{x} - \mathbf{z}$	TRUE	FALSE✓
ii.	$K(\mathbf{x}, \mathbf{z}) = x_1 z_1$	TRUE✓	FALSE
iii.	$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z} + c)^2, \ c < 0$	TRUE	FALSE✓
iv.	$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^2$	TRUE✓	FALSE
v.	$K(\mathbf{x}, \mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|_2^2}{2\sigma^2}}, \ \sigma > 0$	TRUE✓	FALSE

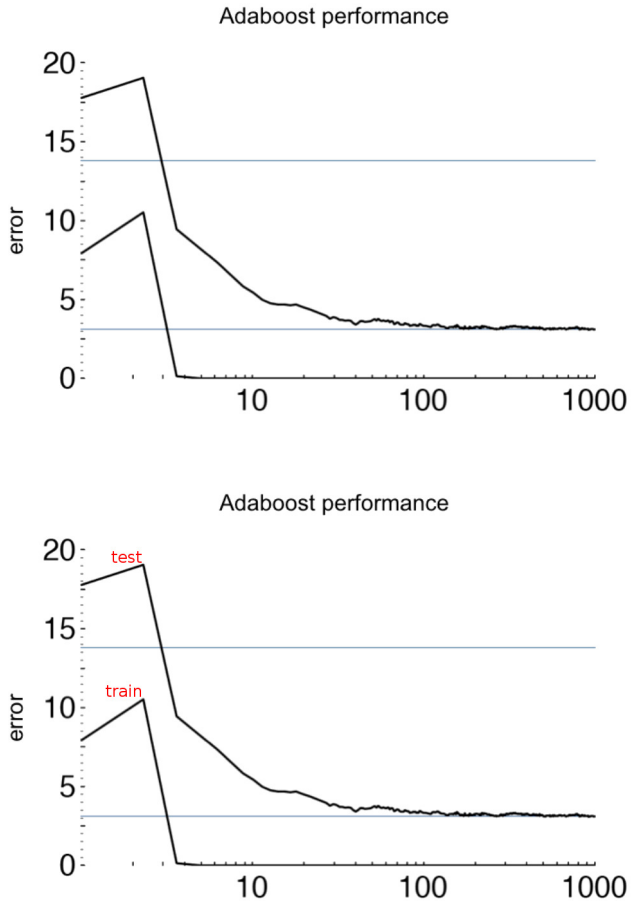
Kernel is a mapping to scalar, so (i) is wrong. (ii) is obviously an inner product and is TRUE. (iii) $= (\mathbf{x} \cdot \mathbf{z})^2 + 2c\mathbf{x} \cdot \mathbf{z} + c^2$ and when $c < 0$, it's not guaranteed to be positive definite. (v) is the RBF kernel. Each one worth 0.5 pts here and up to 2 pts can be deducted.

(ii) Among the functions above for which you circled "TRUE", which would be good choices to use for the data in Fig. 2. List all options that apply.

(iv) $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^2$ and (v) RBF. 1 pt each.

(e) (6 pts)

The following graph illustrating the performance of Adaboost appeared in Schapire et. al. ICML 1997:



These results were obtained by boosting Quinlan's C4.5 decision tree learning algorithm on a UCI data set, varying the number of boosting rounds and recording the error of Adaboost's hypothesis on both the training set and a separate testing set.

(i) Which curve on the graph corresponds to training error and which to testing error? Put the labels "train" and "test" on the graph as your response.

(ii) What did learning theorists find unusual or interesting about these results?

The testing error continues to decrease even after the training error is 0.

(iii) How can this interesting phenomenon be explained? Use evidence from the Adaboost learning algorithm and give a concise answer.

i. Adjustments made by Adaboost during each round of boosting are not a function of final, combined hypothesis' error.

ii. They are a function of the weak learner's error, which continues to be non-zero after the training error of the combined hypothesis reaches zero.

iii. Thus, the distribution kept by the algorithm continues to be modified, and useful features continue to be added to the combined hypothesis.

4 Decision Trees (24 points)

In this problem we will use the data set S described by the table below and the ID3 algorithm to learn a decision tree.
All computations in this problem are simple and only require the use of fractions; see the **appendix** for useful formulas and approximations.

	Attributes					
	x_1	x_2	x_3	x_4	x_5	y
1	1	0	1	1	1	(-)
2	1	0	0	1	1	(-)
3	1	1	1	0	0	(+)
4	1	0	1	0	1	(+)
5	0	1	1	0	0	(+)
6	0	0	0	1	1	(-)
7	1	0	0	1	0	(-)
8	1	1	0	1	1	(+)

Table 2: The data set S.

(a) (2 pts) Calculate the entropy of the label y.
$-\frac{1}{2}\log_2(-\frac{1}{2}) - \frac{1}{2}\log_2(-\frac{1}{2}) = 1$

(b) (2 pts) Compute $Gain(S, x_4)$ and complete the table below.
$H(S) - (\frac{5}{8}H(y|x_4 = 1) + \frac{3}{8}H(y|x_4 = 0)) = \frac{5}{8}$

$Gain(S, x_1)$	0
$Gain(S, x_2)$	$\frac{1}{2}$
$Gain(S, x_3)$	$\frac{1}{5}$
$Gain(S, x_4)$	
$Gain(S, x_5)$	$\frac{1}{10}$

Table 3: Information gains of each attribute.

(c) (9 pts) In this question you will learn the **minimal** decision tree that is consistent with S. Use the data in Table 3 to answer the following questions:

- What is the root node?
  x_4

- What subsets of the data will the algorithm consider once the root node has been chosen? Use the indices of the examples in the table to answer this question.
  $\{1, 2, 6, 7, 8\}$,$\{3, 4, 5\}$

- Continue to construct the minimal decision tree you can find that is consistent with the data. (Note: you do not need to do more information gain calculations.)

if(x_4):
    if(!x_2):
        (-)
    else:
        (+)
else:
    (+)

(d) (2 pts)

Write down the *simplest* Boolean function you can that is identical to the decision tree you have written above.

$\neg x_4 \vee x_2$

(e) (4 pts) Rewrite the Boolean function as a linear threshold unit over the features $\{x_1, x_2, x_3, x_4, x_5\}$.

$x_2 - x_4 + 1 > 0$

(f) (5 pts) Suppose that the data set we have is much larger than the example above, call it S_{large}. After learning a decision tree over S_{large} we find that the test error is significantly larger than training error. Suggest at least two strategies we might try to address this issue given the constraint that there is no more data. Please be brief and clear.

Possible answers:

(a) Decrease the max height of the tree

(b) Prune

(c) Boosting

(d) Random forests

Appendix

Some useful formulas...

1. $Entropy(S) = -p_+ log_2(p_+) - p_- log_2(p_-)$
2. $Gain(S, a) = Entropy(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$
3. $log_2(3) \approx 3/2$
   $log_2(5) \approx 11/5$