Introduction to Machine Learning (236756)

Spring 2014

Final Exam A

Lecturers: Nati Srebro-Bartom, Nir Ailon

TA: Lior Friedman

Instructions:

1. This exam contains 8 question pages, as well as 4 additional pages, including this one. Please make sure you have all 12 pages.
2. The length of the exam is three hours (180 minutes).
3. You may not use any additional material.
4. You may write in a blue or black pen, or use a pencil.
5. All answers should be written on the exam form, **and you must return it at the end of the exam.**
6. You may use the backs of pages, as well as the last two pages as draft paper.
7. You must answer all questions.
8. **Answer only within the answer boxes.**
9. Mark or circle the correct option, **do not mark incorrect options.**
10. Please write clearly and legible. **Illegible answer will not be checked.**
11. **Only answer the questions asked—there is no need for additional explanations or details.**

Brevity is the soul of wit

"Hamlet" act 2 scene 2

12. Do not tear off pages from the exam form.

Good Luck!

Question 1-True/False (10 points):

Mark the correct box. (Leave the other one unmarked)

True [✓ False] a. If we are not concerned with computational issues, it is always better to use a higher value of k when using k-Nearest-Neighbors prediction.

True False b. When performing PCA over $\mathbb{R}^{10}$, the third principal component is always orthogonal to the first principal component.

True [✓ False] c. Consider a training set S and some feature map $\phi(x) \in \{-1,+1\}^{17}$, and using AdaBoost on the training set with each of the features as a weak classifier. If the data set is linearly separable then after some finite number of iterations, AdaBoost will find a linear predictor w with $L_S^{01}(w) = 0$.

True False d. Consider learning whether a person is sick based on 20 binary symptoms. If we know that the symptoms are independent given that the person is sick, and also independent given that the person is not sick, then it is better to use Naïve Bayes then to use logistic regression.

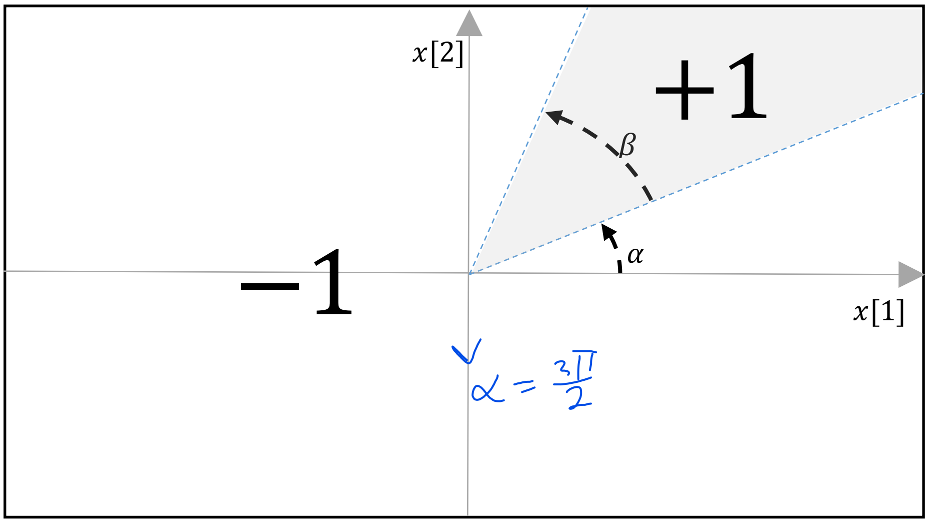
True [✓ False] e. There exists a finite m and a learning rule A such that for any distribution over binary labeled strings, if we run A on m randomly drawn examples, with probability at least 99% it will return a predictor with generalization error at most 0.01 more than the best error we can get using any Python program.

Question 2 (20 points):

Consider the space $\mathcal{X} = \mathbb{R}^2$ and the hypothesis class of "wedges" around the origin:

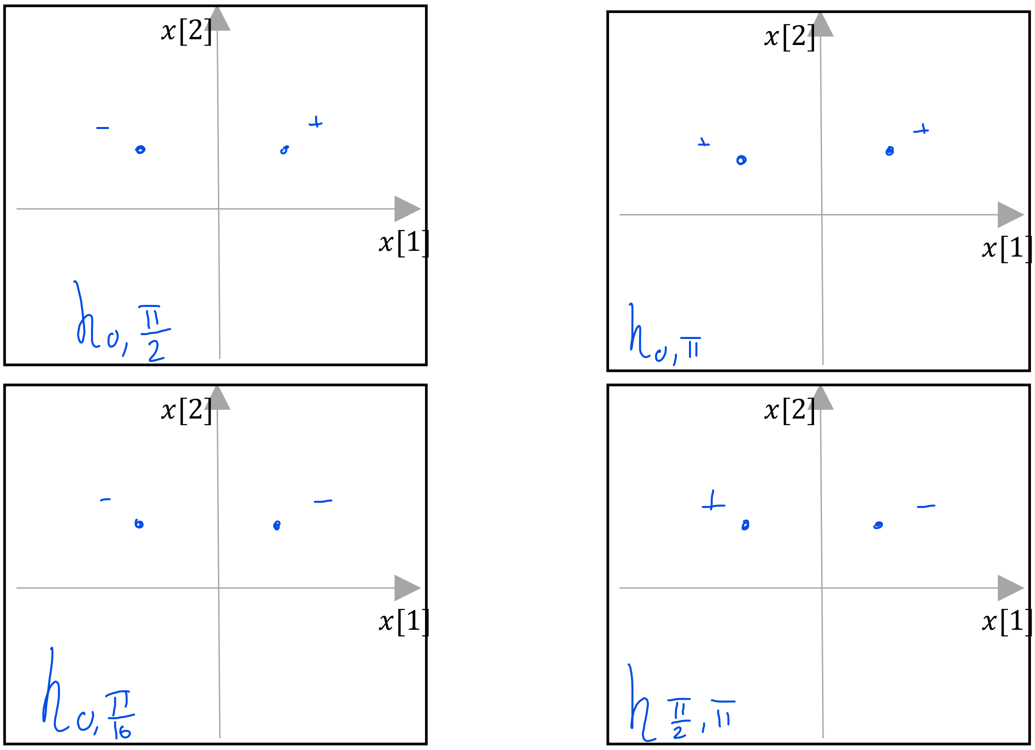
$$\mathcal{H}_{pizza} = \{\, h_{\alpha,\beta}(x) = [[\angle(\vec{\alpha}, x) \le \beta]] \mid \alpha, \beta \in [0, 2\pi] \,\}$$

Where $\angle(u, v)$ is the angle, going counter-clockwise, from the vector u to the vector v, and $\vec{\alpha} = (\cos\alpha, \sin\alpha)$ is a vector with angle α from the $x[1]$ axis. For example:



Note that that it's possible for the "wedge" to cross back into the first orthant, e.g. if $\alpha = 3\pi/2$ and $\beta = \pi$.

a. Give an example of a set of two points that can be shattered, and show how these points can be shattered: Do this by showing, in the four diagrams below, how all label combinations can be attained by hypotheses in the class:

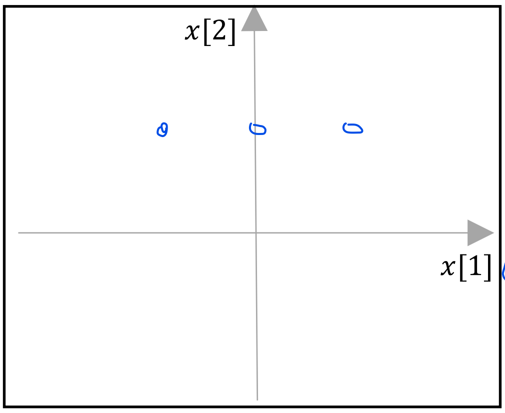


b. Give an example of a set of two points that **cannot** be shattered, and corresponding labels for these two points that cannot be attained by the hypothesis class:

$x_1 = (\boxed{1\,,\,1})$ $y_1 = \boxed{+}$

$x_2 = (\boxed{2\,,2})$ $y_2 = \boxed{-}$

c. Give an example of a set of three points that **can** be shattered: (just show the points, no need to show how to shatter them)



H_{pizza}:

$\left\{ H_{\alpha,\beta}(x) = [[\langle \tilde{\alpha}, x \rangle < \beta]] \;\middle|\; \alpha,\beta \in [0, 2\pi] \right\}$

d. We can write $\mathcal{H}_{pizza} = \{\, h_{a,b,c}(x) = [[\, a \cdot \phi(x) + b \cdot \phi(x)^2 > c]] \,|a, b, c \in \mathbb{R}\,\}$. We can do so using the following feature:

$\phi(x) = (\sin x, \cos x)$ $\left(x_1^2, x_2^2, 2x_1x_2 \right)$

e. What is the VC-dimension of $\mathcal{H}_{pizza}$? $\boxed{3}$

f. Suppose we knew the correct labeling is exactly specified by a pizza wedge in $\mathcal{H}_{pizza}$. We have access to training data $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ and to an implementation of the perceptron algorithm, which performs updates of the form $w \leftarrow w + y \cdot \psi(x)$. What feature map should we use? **You may write your answer in terms of $\boldsymbol{\phi(x)}$.**

$\psi(x) =$

g. If our implementation was kernalized, and accepted only a Kernel function, what kernel function should we use? Write a function which is **as short as possible**. You may write your answer in terms of $\phi(x)$.

$K(x, x') =$

Question 3 (15 points):

Consider the following training set of pairs $(x, y), x \in \mathbb{R}^4, y \in \{\pm 1\}$:

	X[1]	X[2]	X[3]	X[4]	Y	
x_1	2	1	10	7	+1	1/14
x_2	2	1	10	7	+1	
x_3	1	5	10	1	+1	
x_4	1	5	2	1	+1	1/2
x_5	1	5	3	7	-1	
x_6	1	5	4	7	-1	1/14
x_7	3	1	5	7	-1	
x_8	3	1	6	7	-1	

X[4] < 6

x[1] < 3

What happens when we run the AdaBoost algorithm on this training set, with Decision Stumps of the form $[[x[j] < \theta]]$ or $[[x[j] > \theta]]$ as weak predictors? For your convenience and comfort, we include the code for the AdaBoost algorithm, but **you do not need to calculate any roots or logarithms or perform any other complex calculations.**

$\alpha_1 = \frac{1}{2} \ln(8-1)$

$\alpha_2 = \frac{1}{2} \ln(7-1)$

a. What is the first decision stump chosen? (hint: only one point misclassified)

$[[\, x[3] > 9 \quad]]$

b. What is the weight of x_1 after the first weight update? $D_1^{(2)} = \frac{1}{14}$

c. What is the weight of x_4 after the first weight update? $D_4^{(2)} = 1/2$

(hint: remember what the weighted 01 error of the weak hypothesis should be after the weighting)

d. What is the second decision stump chosen? (hint: two points misclassified)

$[[\, x[1] < 3 \quad]]$

e. What is the weighted error ε_2 of this weak classifier? $\epsilon_2 = \frac{2}{14}$

f. What points are classified **incorrectly** by the combined classifier h_s created by AdaBoost after $T = 2$ iterations?

x_1	x_2	x_3	[✓ x_4]	x_5	x_6	x_7	x_8
✓ ✓	✓ ✓	✓ ✓	✗ ✓	✓ ✗	✓ ✗	✓ ✓	✓ ✓

Question 4 (30 points):

ABG Marketing is attempting to train a predictor that would predict, based on the text of a product review, whether it is positive or negative. For this purpose they collected 100,000 reviews (which we will assume were independently and randomly drawn from all reviews of interest) for which they also have labels indicating whether the review is positive or not.

Avi, an employee, learns a linear predictor over word counts. That is, he uses the feature representation $\phi(x) \in \mathbb{R}^{987}$, where $\phi_A(x)[i]$ is the number of appearance of word i, for the 987 distinct words that appeared in the reviews. He splits the data randomly into S_1 of size 90,000 and S_2 of size 10,000, and learns a predictor of the form $x \mapsto \langle w, \phi_A(x)\rangle$ by minimizing the hinge loss over S_1:

$$\hat{w}_A = \arg\min_{w} L_{S_1}^{hinge}(x \mapsto \langle w, \phi_A(x)\rangle)$$

Avi found that $\boldsymbol{L_{S_2}^{01}(\hat{w}_A) = 0.37}$. This was not good enough for the company.

Benny, another employee, instead suggests using as features counts of three-word-phrases. That is, using $\phi_B(x) \in \mathbb{R}^{91{,}529}$, where $\phi_B(x)[i]$ is the number of appearances in review x of one of 91,529 distinct phrases of **up to** three consecutive words (e.g. “is not good” or “also yellow” or “my”). He repeated the same protocol as Avi, learning $\hat{w}_B = \arg\min_{w} L_{S_1}^{hinge}(x \mapsto \langle w, \phi_B(x)\rangle)$ and found that $\boldsymbol{L_{S_2}^{01}(\hat{w}_B) = 0.51}$.

a. Circle the correct answer: The increase in error is mostly due to an increase in the-

Approximation Error \ [✓ Estimation Error] \ Optimization Error

b. Given the above data, which of the following might reasonably help Benny improve over Avi’s result? **(Mark all that apply)**

☑ 1. Add a regularization term and learn again using regularization.

☑ 2. Use only phrases of up to two words. (No regularization)

☒ 3. Use four-word phrases as well. (No regularization)

☑ 4. Use a significantly larger training set. (No regularization)

Benny finally settled on using regularization and learning:

$$\hat{w}_{B,\lambda} = \arg\min_{w} L_{S_1}^{hinge}(x \mapsto \langle w, \phi_B(x)\rangle) + {\uparrow}\lambda\|w\|_1 \downarrow$$

He used $\lambda = 0.0001$ and got $\boldsymbol{L_{S_2}^{01}(\hat{w}_{B,\lambda}) = 0.47}$. To better understand what was going on, he checked $\widehat{w_{B,\lambda}}$ on S_1 and got $\boldsymbol{L_{S_1}^{01}(\hat{w}_{B,\lambda}) = 0.45}$. underfitting

c. Given the above data, which of the following might reasonably help Benny **improve over Avi’s result**? **(Mark all that apply)**

☒ 1. Use a larger value of λ.

☒ 2. Use a smaller value of λ.

☑ 3. Use the same λ, but only use phrases of up to two words.

☒ 4. Use the same λ, but also use four-word phrases.

☑ 5. Use the same λ, but use a significantly larger training set.

d. Galit, an expert hired by the company, offered Benny to use a cross-validation procedure in order to set the value of λ. Complete the code in the best manner possible:

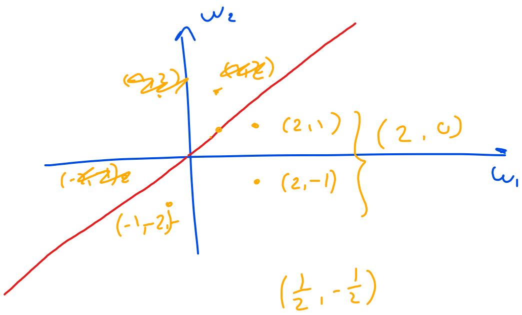
```
Randomly split $S$ into five sets $S_1', \dots, S_5'$, each of size 20,000.
For each $\lambda = 10 ** (-8: 0.2: -1)$ :
        For each $i = 1..5$:
                $\hat{w}_{\lambda,i} \leftarrow \arg\min_w L^{hinge}_{\boxed{S_{train}}}(w) + \lambda \|w\|_1$
$S_{train}$ = 4 subsets in the i-th fold used for training
        Calculate $e_\lambda$= $\boxed{\frac{1}{5}\sum_{i=1}^{5} L^{hinge}_{S_i}(\hat{w}_{\lambda_i})}$
Select $\lambda^* = \arg\min_\lambda e_\lambda$
Output $\hat{w}_G = \arg\min_w L_S^{hinge}(w) + \lambda^* \|w\|_1$
```

Question 5 (10 points):

Which of the following sets X are Convex? If the set is convex, mark so. If not, give two points $x_1, x_2 \in X$ such that their midpoint is not in X: $\frac{x_1+x_2}{2} \notin X$

a. $\{ w \in \mathbb{R}^2 \mid \|w\|_1 \leq 3, w[2] > 1 \}$	[✓ CONVEX]	$x_1 = (\quad , \quad)$	$x_2 = (\quad , \quad)$
b. $\{ w \in \mathbb{R}^2 \mid \|w\|_1 \geq 3, w[2] < 1 \}$	CONVEX	$x_1 = (5 , 0)$	$x_2 = (-5 , 0)$
c. $\{ w \in \mathbb{R}^2 \mid \|w\|_1 \geq 3, w[1] > 0, w[2] > 0 \}$	[✓ CONVEX]	$x_1 = (\quad , \quad)$	$x_2 = (\quad , \quad)$
d. $\{ w \in \mathbb{R}^2 \| w \textit{ has at most } 1 \textit{ nonzero entry} \}$	CONVEX	$x_1 = (5 , 0)$	$x_2 = (-5 , 0)$
e. $\{ w \in \mathbb{R}^2 \mid w[1] > w[2] \}$	[✓ CONVEX]	$x_1 = (\quad , \quad)$	$x_2 = (\quad , \quad)$

$\forall u,v \in C, \alpha \in [0,1]$ $\quad \alpha u + (1-\alpha) v \in C$

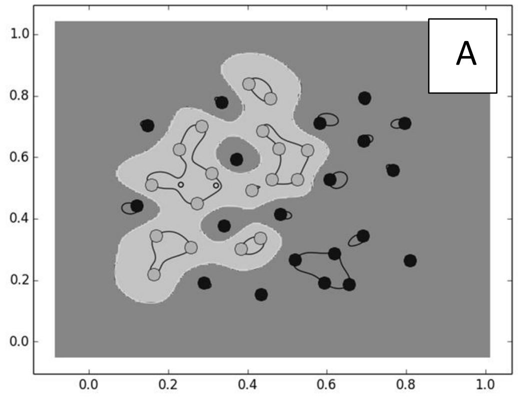


Question 6 (15 points):

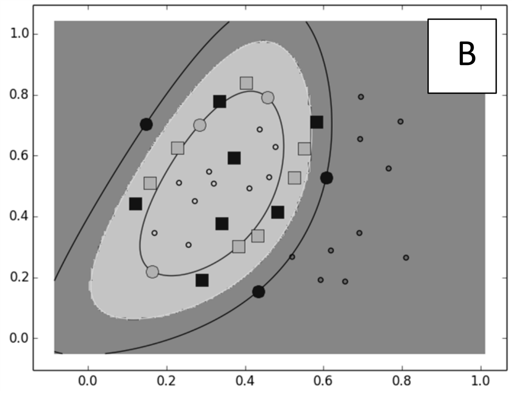
We trained a SVM on the same training set S of labeled points $(x, y), x \in \mathbb{R}^2$ using different kernels by minimizing:

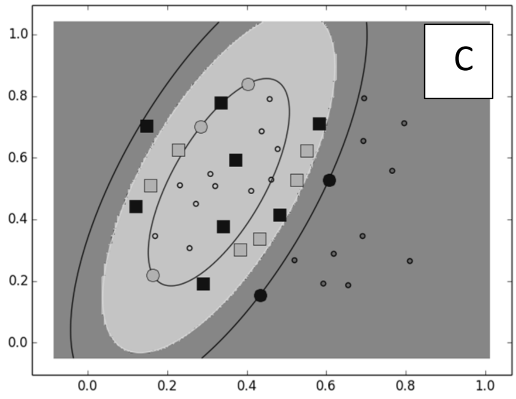
$$\|w\|^2 \ + C \cdot L_S^{hinge}(w)$$

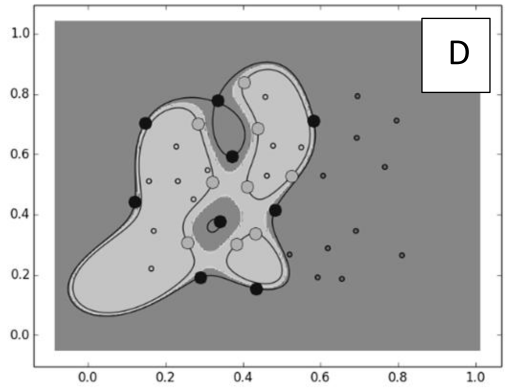
Match the following plots to their corresponding kernels and regularization parameters:

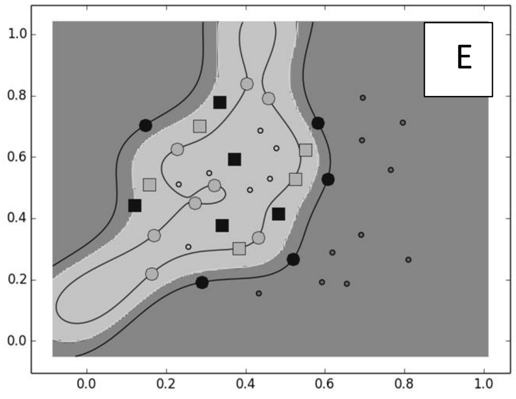


$C = 10000 , K(x,x') = e^{-10\|x-x'\|^2}$	D
$C = 100 , K(x,x') = e^{-10\|x-x'\|^2}$	E
$C = 100 , K(x,x') = e^{-200\|x-x'\|^2}$	A
$C = 1000 , K(x,x') = (1 + \langle x,x' \rangle)^2$	C
$C = 1000 , K(x,x') = (1 + \langle x,x' \rangle)^3$	B









All plots contain the points in S, the decision areas and lines representing the margins. The bright area is labeled -1 and the dark $+1$. Squares represent margin violations and large circles represent support vectors which are not margin violations

Good Luck!

Reminder page

The AdaBoost algorithm:

Input: A set of labeled examples $\mathrm{S} = \{(x_1, y_1), \dots, (x_m, y_m)\}$ and a class of weak classifiers $\mathcal{H}$.

Output: A classifier.

```
Initialize $D^{(1)} = \left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}\right)$
For t=1, …, T:
    $h_t = \arg\min_{h \in \mathcal{H}} L^{01}_{D^{(t)}}(h) = \arg\min_{h \in \mathcal{H}} \sum_i D_i^{(t)} \cdot [[h_t(x_i) \neq y_i]]$
    $\epsilon_t = L^{01}_{D^{(t)}}(h_t) = \sum_i D_i^{(t)} \cdot [[h_t(x_i) \neq y_i]]$
    $w_t = \frac{1}{2} \log\left(\frac{1}{\epsilon_t} - 1\right)$
    $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(x_i))}{\sum_j D_j^{(t)} \exp(-w_t y_j h_t(x_j))}$
Output: $h_s(x) = sign(\sum_{t=1}^{T} w_t h_t(x))$
```

Draft Page

Draft Page