54/54

EXAM: SPRING 2019
CIS 472/572
INSTRUCTOR: THIEN HUU NGUYEN

May 16, 2019

The exam is closed book and open notes (1 page, handwritten except with prior permission). You will have 1 hour and 20 minutes to do this exam. Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page. Undergraduates only: There are 9 problems in this exam. You may skip one of these problems (either Problem 1, 2, 3, 4, 5, 6, 7, 8 or 9). Please write down on the front of your test which problem you are choosing to skip. You will receive full credit on the skipped question.

NAME Zayd Hammoudeh

Zayd Hammoudeh

PROBLEM 1: PERCEPTRONS

Suppose you learn a perceptron on a linearly separable dataset by running the perceptron learning algorithm until convergence. Which of the following could be affected by changing the order of the training examples? For full points, **you must briefly explain each answer** (one sentence is enough).

1. (2 points) The convergence rate (number of iterations before convergence) could change. [✓ True] or False. **Explain.**

True. If positive and negative examples are not well mixed, weights could thrash more taking longer to converge.

✓ −0

2. (2 points) The final perceptron accuracy on the training data could change. True or [✓ False]. **Explain.**

False. There is universal convergance of a perceptron on linearly separable data.

✓

3. (2 points) The final perceptron accuracy on a separate, unseen test set could change. [✓ True] or False. **Explain.**

True. The hyperplane may be "barely separating." This means it may have limited generalizability near the decision boundary and may have different accuracy on the test set in turn.

✓

Zayd Hammoud

PROBLEM 2: PERCEPTRON UPDATES

You are training a classifier to distinguish between bubonic plague (+1) and the flu (-1).

Patient	chills	fever	cramps	seizures	gangrene	Disease
	X_1	X_2	X_3	X_4	X_5	Y
1.	1	1	0	1	1	1
2.	1	1	0	0	0	-1
3.	1	1	1	1	0	1
4.	1	0	1	0	0	-1

1. (6 points) Show the weights and bias (w and b) obtained by running perceptron algorithm on this dataset for one iteration. (Here, "one iteration" means going over all of the examples once, in the order shown above.)

$\omega_0 = \vec{0}$, $b = \vec{0}$

First sample $\vec{\omega}_0 \cdot \vec{P}_1 + b_0 = 0$ (wrong) need to update

$\vec{\omega}_1 = \vec{\omega}_0 + y \cdot P_1 = \boxed{\langle 1, 1, 0, 1, 1 \rangle}$ ✓

$b_1 = b_0 + y = \boxed{1}$

−0

Second Sample $\vec{\omega}_1 \cdot \vec{P}_2 + b_1 = 2 + 1 = 3$ sign(3) = 1 (wrong) need to update

$$\begin{array}{cc} \omega_1 & P_2 \\ \begin{bmatrix}1\\1\\0\\1\\1\end{bmatrix} \cdot & \begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix} \end{array} = 1+1+0 \\ = 2$$

$\vec{\omega}_2 = \vec{\omega}_1 + y \cdot P_2 = \boxed{\langle 0, 0, 0, 1, 1 \rangle}$ ✓

$b_2 = b_1 + y = 1 - 1 = \boxed{0}$

Third Sample $\vec{\omega}_2 \cdot \vec{P}_3 + b_2 = 1 + 0 = 1$ sign(1) = 1 (right) no update

$$\begin{array}{cc} \omega_2 & P_3 \\ \begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix} \cdot & \begin{bmatrix}1\\1\\1\\1\\0\end{bmatrix} \end{array} = 0+0+0+1+0 \\ = 1$$

$\vec{\omega}_3 = \vec{\omega}_2 = \boxed{\langle 0, 0, 0, 1, 1 \rangle}$ ✓

$b_3 = b_2 = \boxed{0}$

Fourth Sample $\vec{\omega}_3 \cdot \vec{P}_4 + b_3 = 0 + 0 = 0$ (wrong) need to update

$$\begin{array}{cc} \omega_3 & P_4 \\ \begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix} \cdot & \begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix} \end{array} = 0+0+0+0+0 = 0$$

$\vec{\omega}_4 = \vec{\omega}_3 + y \cdot P_4 = \boxed{\langle -1, 0, -1, 1, 1 \rangle}$

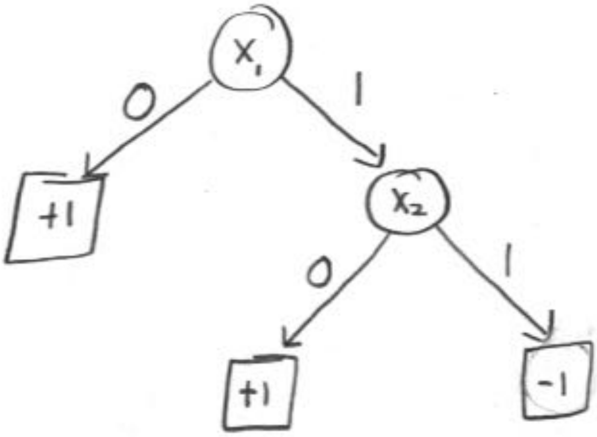
$b_4 = b_3 + y = 0 - 1 = \boxed{-1}$ ✓

Zayd Hammoudeh

PROBLEM 3: REPRESENTATION

Let x_1 and x_2 be two binary-valued attributes, which can take on values of 0 or 1. Consider the NAND function, $y = \neg(x_1 \wedge x_2)$, which is -1 if both $x_1 = 1$ and $x_2 = 1$ and +1 otherwise.

1. (3 points) Draw a decision tree that represents this function.



$y = \neg x_1 \vee \neg x_2$

✓

-0

2. (3 points) Specify parameters for a linear classifier that represents this function. (You do not need to find a maximum margin separator – any separator will do. Do not create any additional attributes.)

$$\vec{\omega} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$b = 1.5$$

$$y = \text{sign}(\vec{\omega} \cdot \vec{x} + b)$$

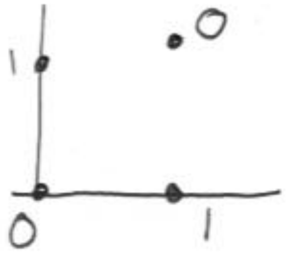
✓

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	-1

PROBLEM 4: LEARNING POWER

Which of the following classifiers will have zero training error (under 0/1 loss) on the following dataset? Circle YES for classifiers that will have zero error and NO for those that will not. Give a short explanation (one sentence) for each answer.

X1	X2	Category
1	1	0
1	0	1
0	1	1
0	0	0



1. (2 points) 3-nearest neighbor: YES or [✓ NO?] Explain.

No (Assume X_1 & X_2 binary features). If you are on any coordinate, e.g. (1,1), the two other closest points have ✓ opposite category and will cause you to be wrong 100% of the time since those two are always different so with equal weighting it will always choose two closest causing wrong label.

2. (2 points) Logistic regression: YES or [✓ NO?] Explain.

No. This is simply the XOR function which is not linearly separable Logistic regression cannot properly classify this ✓

−0

3. (2 points) Neural network with a single hidden layer, and only one node in the hidden layer: YES or [✓ NO?] Explain.

No. Two hidden nodes required to create classifier for XOR. One hidden node even with nonlinearity is same as one output with a potentially different nonlinearity. Still linear decision functions connot classify xor.

✓

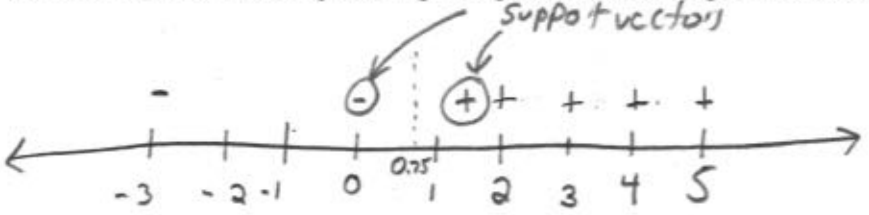
Zayd Hammoudeh

PROBLEM 5: LINEAR SUPPORT VECTOR MACHINES

Consider the following 1-dimensional data:

x	-3	0	1.5	2	3	4	5
Class	−	−	+	+	+	+	+

1. (3 points) Draw the decision boundary of a (hard-margin) linear support vector machine on this data and identify the support vectors. (The boundary should be a single point on the number line, separating the positive and negative classes.)



Occurs at : $\frac{1.5+0}{2} = \boxed{0.75}$

Support vectors : 0 & 1.5

✓

2. (3 points) Calculate the leave one out cross validation error for this SVM on the data set. (That is, how many individual points would be predicted incorrectly if they were removed from the training data?)

Leave 3 : Decision boundary unchanged and correct

Leave 0: Decision boundary $\frac{-3+1.5}{2} = -.75$ (wrong)

Leave 1.5 : Decision boundary $\frac{0+2}{2} = 1$ (right).

Leave 2,3,4,5: Decision boundary unchanged and all correct.

✓

Only 0 misclassified so accuracy is $\boxed{\frac{6}{7}}$

Loyd Hammoudeh

PROBLEM 6: KERNELS

Consider a kernelized SVM with $b = 0$ and the following instances and weights:

instance ($x^{(i)}$)	label ($y^{(i)}$)	weight (α_i)
(0,0,0)	+1	1.0
(1,1,1)	-1	1.0

What is the predicted label for the instance $x = (-1, -1, -1)$ under each kernel? (NOTE: Be careful with positive and negative signs! I recommend showing your work in order to have a chance at partial credit.)

b zero so dropping it in the work

1. (2 points) Linear kernel, $K(x, x') = x \cdot x'$.

$h(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i \; y_i \left(K(x_i, x) + b \right)\right)$

$= \text{Sign}\left(\alpha_0 \cdot y_0 K(x_0 \cdot x) + \alpha_1 \cdot y_1 \cdot K(x_1, x)\right)$

$= \text{Sign}(1 \cdot 1 \cdot 0 + 1 \cdot -1 \cdot -3)$ ✓

$= \text{Sign}(3) = \boxed{1}$

$K_1(x_0, x) = 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 = 0 = x_0 \cdot x$

$K_1(x_1, x) = 1 \cdot -1 + 1 \cdot -1 + 1 \cdot -1 = -3 = x_1 \cdot x$

-0

2. (2 points) Quadratic kernel, $K(x, x') = (1 + x \cdot x')^2$

$h(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i \; y_i \left(K_2(x_i, x) + b \right)\right)$

$= \text{sign}(1 \cdot 1 \cdot 1 + 1 \cdot -1 \cdot 4)$

$= \text{Sign}(1 - 4) = \text{Sign}(-3) = \boxed{-1}$ ✓

$\left.\begin{aligned} K_2(x_0, x) &= (1 + 0)^2 = 1 \\ K_2(x_1, x) &= (1 + -3)^2 = 4 \end{aligned}\right\}$ dot product above

$= (-2)^2 = 4$

3. (2 points) Cubic kernel, $K(x, x') = (1 + x \cdot x')^3$

$h(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i \; y_i (K_3(x_i, x) + b)\right)$

$= \text{sign}(1 \cdot 1 \cdot 1 + 1 \cdot -1 \cdot -8)$

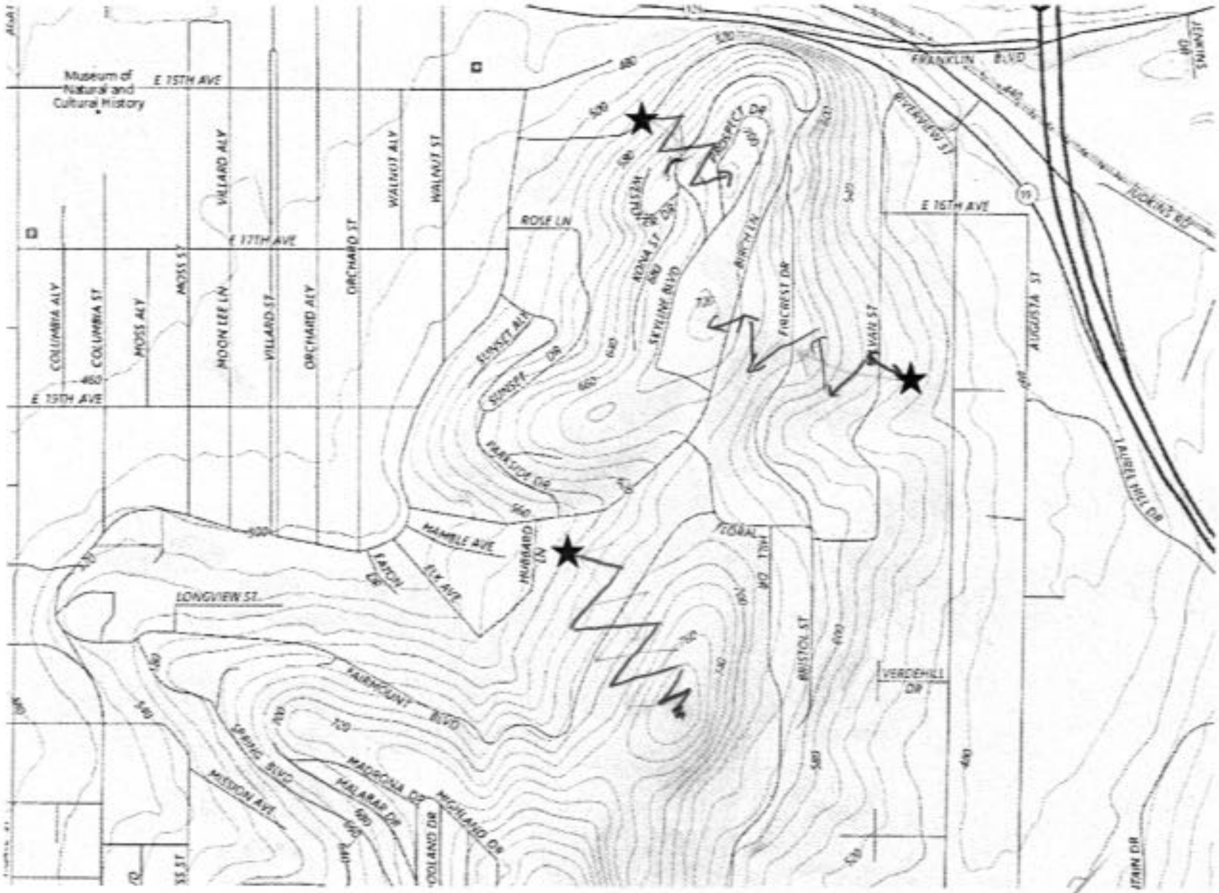
$= \text{sign}(9) = \boxed{1}$ ✓

$\left.\begin{aligned} K_3(x_0, x) &= (1 + 0)^3 = 1 \\ K_3(x_1, x) &= (1 + -3) = -2^3 = -8 \end{aligned}\right\}$ dot product above

Zayd Hammoudeh

PROBLEM 7: GRADIENT DESCENT

1. (6 points) Starting at each location marked with a star, draw the path that gradient ascent (that is, following the gradient uphill) would take until it converges to a local optimum. Assume that the step size is small enough for a smooth path. Your path does not need to be perfect.



—0

✓

Zoyd Hammock

PROBLEM 8: TRUE/FALSE QUESTIONS

1. (2 points) Classifier A has 90% accuracy on the training set and 75% accuracy on the test set. Classifier B has 78% accuracy on both the training and test sets. Therefore, we can conclude that classifier A is better than classifier B (because it has better mean accuracy). True or [✓ False.] **Explain.**

False. Classifier B generalizes better (i.e. has higher accuracy on the test test.) Classifier A has most likely overfit the training set.

✓

2. (2 points) Training a kernelized SVM model is equivalent to training a linear model with an expanded set of features. [✓ True] or False. **Explain.**

true. Kernel is equivalent to a dot product after feature mapping via ϕ so still a linear model just with $K(x_1, x_2) = \phi(x_1)\phi(x_2)$ with ϕ implicit.

✓ —0

3. (2 points) Consider two logistic regression models, trained on the same dataset with gradient descent until convergence, but initialized with different initial random weights. Given a sufficiently small learning rate and convergence threshold, both models will have the same accuracy on the test set. [✓ True] or False. **Explain.**

True since loss is convex. Assuming appropriate learning rate, always converge to global minimum (assuming logistic loss) on training set. Therefore essentially same model meaning essentially same test set accuracy (for appropriate convergence threshold).

✓

Zayd Hammoudeh

PROBLEM 9: PERCEPTRON AND SVM

1. (3 points) In the learning algorithm of perceptron, we run over the training dataset multiple times and make an appropriate update for each training example we encounter along the way. This is very similar to the way we apply stochastic gradient descent to optimize the loss functions we studied in class. Based on this similarity, please suggest the loss function that perceptron is trying to optimize. Your loss function should be a function of the the i-th training example (i.e., with the input $x_i \in \mathbb{R}^d$ and the output $y_i \in \{-1, +1\}$), and the model parameters $w \in \mathbb{R}^d$ and the bias $b \in \mathbb{R}$: $L(x_i, y_i, w, w_0)$.

$$\mathcal{L}(x_i, y_i, \omega, \omega_0) = \max\{0, -y_i(\vec{\omega}x_i + \omega_0)\} \quad ✓$$

−0

2. (3 points) Once you write down the loss function, please justify it by showing that you can derive the update rules for perceptron based on stochastic gradient descent.

Hint: Remember in class, we mentioned the connection between linear SVM and perceptron and that the loss function of SVM is the hinge loss.

If y_i and $(\omega x_i + \omega_0)$ (which are both real scalars), their product is positive. Sine the product is negated, the value is less than zero making the loss 0 and no gradient So no update like perceptron.

If y_i and $(\omega x_i + \omega_0)$ are different signs, product is positive. ✓
The update is the gradient meaning

$$\nabla_\omega \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \omega} = \frac{\partial}{\partial \omega}\left[-y_i(\omega x_i + \omega_0)\right] = \boxed{-y_i x_i}. \text{ Therefore } \omega' = \omega - \nabla_\omega \mathcal{L} = \omega - (-y_i x_i) \boxed{= \omega + y_i x_i}$$

$$\nabla_{\omega_0} \mathcal{L} = \frac{\partial}{\partial \omega_0} = \left[-y_i[\omega x_i + \omega_0]\right] = -y \cdot 1 = \boxed{-y}. \text{ Therefore } \omega_0' = \omega_0 - \nabla_{\omega_0}\mathcal{L} = \omega - (-y) = \boxed{\omega + y}$$

Using only a single sample for each update since that is definition of stochastic gradient descent