

CS273A Midterm Exam
Introduction to Machine Learning: Fall 2023
Monday November 6th, 2023

Your name:

Peter Anteater

Row/Seat Number:

Your ID #(e.g., 123456789)

314 159265

UCINETID (e.g. ucinetid@uci.edu)

panteat@uci.edu

- Please put your name and ID **on every page**.
- Total time is 80 minutes. READ THE EXAM FIRST and organize your time; don't spend too long on any one problem.
- Please **write clearly** and **show all your work**.
- If you need clarification on a problem, please raise your hand and wait for the instructor or TA to come over.
- You may use **one** sheet containing handwritten notes for reference, and a (basic) calculator.
- Turn in your notes and any scratch paper with your exam.

Problems

1	Cross-Validation, <i>(15 points.)</i>	3
2	True/False, <i>(10 points.)</i>	5
3	Support Vector Machines, <i>(12 points.)</i>	7
4	Naïve Bayes Classifiers, <i>(12 points.)</i>	9
5	Separability, <i>(16 points.)</i>	11
6	Gradient Descent, <i>(10 points.)</i>	13

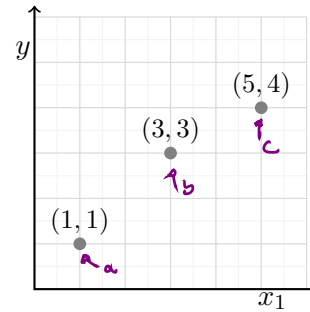
Total, *(75 points.)*

This page is intentionally blank, use as you wish.

Problem 1 Cross-Validation, (15 points.)

For a regression problem to predict real-valued y given a single real-valued feature x_1 , we observe training data (pictured at right):

x_1	y
1.0	1.0
3.0	3.0
5.0	4.0



- (1) Compute the **leave-one-out** cross-validation MSE of a 1-nearest neighbor predictor. (In case of ties, prefer to use the data listed earlier in the table.) (3 points.)

<u>leave out</u>	<u>predict</u>	<u>err</u>
a	3	2
b	1	2
c	3	1

$$MSE = \frac{1}{2} (2^2 + 2^2 + 1^2) = 3.$$

- (2) Compute the **leave-one-out** cross-validation MSE of a 2-nearest neighbor predictor. (In case of ties, prefer to use the data listed earlier in the table.) (3 points.)

<u>out</u>	<u>pred</u>	<u>err</u>
a	3.5	2.5
b	2.5	0.5
c	2	2

$$MSE = \frac{1}{3} \left((5/2)^2 + (1/2)^2 + (4/2)^2 \right) = \frac{42}{12} = 3.5$$

- (3) Compute the **leave-one-out** cross-validation MSE of a constant predictor, $f(x) = \theta_0$. (3 points.)

<u>out</u>	<u>pred</u>	<u>err</u>
a	3.5	2.5
b	2.5	0.5
c	2	2

$$MSE = 3.5$$

(This happens to be the same as the 2-NN model, since the avg of the other 2 points is the same as the mean of all the other (2) points.)

- (4) Compute the **leave-one-out** cross-validation MSE of a linear predictor, $f(x) = \theta_1 x_1 + \theta_0$. (3 points.)

<u>out</u>	<u>pred</u>	<u>err</u>
a	2	1
b	2.5	0.5
c	5	1

$$MSE = \frac{1}{3} \left(1^2 + \left(\frac{1}{2}\right)^2 + 1^2 \right) = 9/12 = 3/4.$$

- (5) If choosing among these models, which would you select (based on cross-validation)? (3 points.)

We should choose the linear model (#4), since it has the smallest LOOXV MSE.

This page is intentionally blank, use as you wish.

Problem 2 True/False, (10 points.)

Here, assume that we have m data points $y^{(i)}, x^{(i)}, i = 1 \dots m$, each with n features, $x^{(i)} = [x_1^{(i)} \dots x_n^{(i)}]$. For each of the scenarios below, circle one of “true” or “false” to indicate whether you agree with the statement.

True or **false**: Using “batch” gradient descent (all m data per step) is less prone to getting stuck in local optima than stochastic gradient descent.

True or **false**: Optimizing a linear classifier using the hinge loss and L_2 regularization will always converge to a global optimum of the loss.

True or **false**: Increasing the number of features available to a perceptron model will increase its bias.

True or **false**: The perceptron algorithm is always guaranteed to converge.

True or **false**: With sufficiently many data, an SVM with polynomial kernel $K(x, x') = (1 + x \cdot x')^2$ can approximate any decision function.

True or **false**: The SVM optimization problem is an example of a linear program (a linear objective with linear constraints).

True or **false**: Using a soft-margin SVM in place of a hard-margin SVM will typically reduce the training error.

True or **false**: A 1-nearest neighbor model is an example of a learner with no inductive bias.

True or **false**: Feature selection (electing to ignore certain features of the data in our model) can be used to reduce model variance.

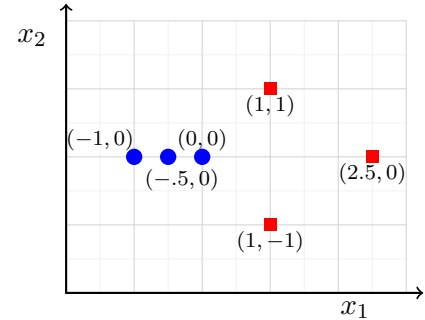
True or **false**: Given sufficiently many layers, a neural network with one hidden node per layer can approximate any function.

This page is intentionally blank, use as you wish.

Problem 3 Support Vector Machines, (12 points.)

For a classification problem to predict binary y given two real-valued features x_1, x_2 , we observe training data (pictured at right):

x_1	x_2	y
0.0	0.0	-1
-0.5	0.0	-1
-1.0	0.0	-1
1.0	-1.0	1
1.0	1.0	1
2.5	0.0	1

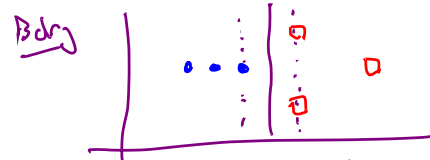


Our linear classifier takes the form,

$$f(x; w_1, w_2, b) = \text{sign}(w_1 x_1 + w_2 x_2 + b).$$

- (1) Consider the optimal linear SVM classifier for the data, i.e., the one that separates the data and has the largest margin. **Sketch** its decision boundary in the above figure, and **list** the support vectors here. (2 points.)

$$\text{SVC: } (0, 0), (1, 1), (1, -1)$$



- (2) Derive the parameter values w_1, w_2, b of this $f(x)$ using these support vectors. (4 points.)

$$\begin{aligned} w_1 \cdot 0 + w_2 \cdot 0 + b &= -1 & b &= -1 \\ w_1 \cdot 1 + w_2 \cdot 1 + b &= +1 & \Rightarrow w_2 &= 0 \\ w_1 \cdot 1 + w_2 \cdot (-1) + b &= +1 & w_1 &= 2 \end{aligned}$$

- (3) What is the *training error rate* of a linear SVM on these data? (3 points.)

$$0$$

- (4) What is the *leave-one-out cross validation error rate* for a linear SVM trained on these data? (3 points.)

Non-SVCs - all predict correctly ✓

(0, 0) - ok: new boundary is at $x_1 = 1/2$

(1, 1) - wrong:

(1, -1) - wrong:

$$\Rightarrow \text{err} = 2/6 = 1/3$$



This page is intentionally blank, use as you wish.

Problem 4 Naïve Bayes Classifiers, (12 points.)

Consider the table of measured data given at right. We will use the two observed features x_1, x_2 to predict the class y . Feature x_1 can take on one of three values, $x_1 \in \{a, b, c\}$; feature x_2 is binary, $x_2 \in \{0, 1\}$. In the case of a tie, we will prefer to predict class $y = 0$.

x_1	x_2	y
b	0	0
c	1	0
a	0	0
a	1	0
c	0	1
c	1	1
c	1	1
a	0	1
c	0	1
c	1	1

- (1) Write down the probabilities learned by a naïve Bayes classifier: (4 points.)

$$p(y = 0) : \quad 4/10$$

$$p(y = 1) : \quad 6/10$$

$$p(x_1 = a | y = 0) : \quad 2/4$$

$$p(x_1 = a | y = 1) : \quad 1/6$$

$$p(x_1 = b | y = 0) : \quad 1/4$$

$$p(x_1 = b | y = 1) : \quad \emptyset$$

$$p(x_1 = c | y = 0) : \quad 1/4$$

$$p(x_1 = c | y = 1) : \quad 5/6$$

$$p(x_2 = 0 | y = 0) : \quad 1/2$$

$$p(x_2 = 0 | y = 1) : \quad 1/2$$

$$p(x_2 = 1 | y = 0) : \quad 1/2$$

$$p(x_2 = 1 | y = 1) : \quad 1/2$$

- (2) Using your naïve Bayes model, compute the probability $p(y = 1 | x_1 = c, x_2 = 0)$: (4 points.)

$$P_0 = p(y=0, x_1=c, x_2=0) = 4/10 \cdot 1/4 \cdot 1/2 = 1/20$$

$$P_1 = p(y=1, x_1=c, x_2=0) = 6/10 \cdot 5/6 \cdot 1/2 = 5/20$$

$$\Rightarrow p(y=1 | x_1=c, x_2=0) = \frac{5}{5+1} = 5/6$$

- (3) Using your naïve Bayes model, compute the probability $p(y = 1 | x_1 = b, x_2 = 1)$: (4 points.)

$$P_0 = p(y=0, x_1=b, x_2=1) = 4/10 \cdot 1/4 \cdot 1/2 = 1/20$$

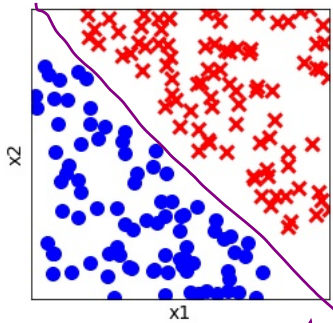
$$P_1 = p(y=1, x_1=b, x_2=1) = 6/10 \cdot \emptyset \cdot 1/2 = \emptyset$$

$$\Rightarrow p(y=1 | x_1=b, x_2=1) = \emptyset.$$

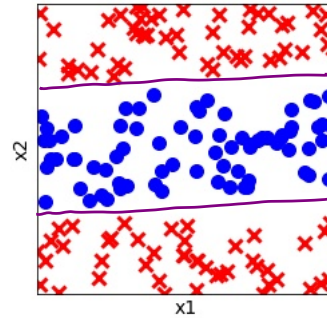
This page is intentionally blank, use as you wish.

Problem 5 Separability, (16 points.)

Consider each of the following pictured data sets. Select **all** sets of features that could be used by a perceptron (linear classifier) to separate the data. (Note: check each feature list carefully; they may differ by subproblem.)

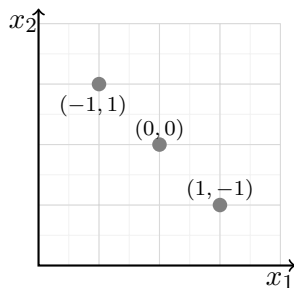


- ☐ $[1, x_1]$
- ☐ $[1, x_2]$
- ☒ $[1, x_1, x_2, (x_1)^2]$
- ☒ $[1, x_1, x_2, (x_2)^2]$
- ☒ $[1, x_1, x_2, (x_1)^2, (x_2)^2, (x_1 x_2)]$
- needs x_1 & x_2
anything else is extra.*

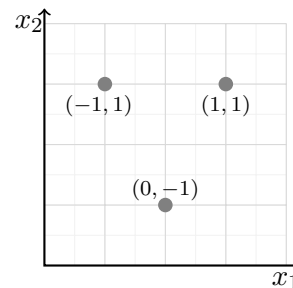


- ☐ $[1, x_1]$
- ☐ $[1, x_2]$
- ☐ $[1, x_1, x_2, (x_1)^2]$
- ☒ $[1, x_1, x_2, (x_2)^2]$
- ☒ $[1, x_1, x_2, (x_1)^2, (x_2)^2, (x_1 x_2)]$
- needs decision boundary
at two different values
of x_2 , but doesn't need
to depend on x_1*

Now consider the following sets of feature vectors, with no target labels. Select **all** the sets of features that could be used by a perceptron (linear classifier) to **shatter** the data, i.e., correctly separate any setting of the target values $y^{(i)}$:



- ☐ $[1, x_1]$
- ☐ $[1, x_2]$
- ☐ $[1, x_1, x_2]$
- ☒ $[1, x_1, (x_1)^2]$
- ☒ $[1, x_2, (x_2)^2]$
- colinear placement
⇒ can't shatter with $[1, x_1, x_2]$,
but quad. function of
either x_1 or x_2 can do it.*



- ☐ $[1, x_1]$
- ☐ $[1, x_2]$
- ☒ $[1, x_1, x_2]$
- ☒ $[1, x_1, (x_1)^2]$
- ☐ $[1, x_2, (x_2)^2]$
- placement ok for shattering with
 $[1, x_1, x_2]$, but two points with
identical x_2 values means we
can't do it without x_1 .*

This page is intentionally blank, use as you wish.

Problem 6 Gradient Descent, (10 points.)

Suppose that we have training data $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, where $x^{(i)}$ is a scalar feature and $y^{(i)} \in \{-1, +1\}$, and we wish to train a linear classifier, $\hat{y} = \text{sign}[a + bx]$, with two parameters a, b . In order to train the model, we use stochastic gradient descent on the *exponential loss* surrogate, whose loss for data point i is given by:

$$J^{(i)}(X, Y) = \exp[-y^{(i)}(a + bx^{(i)})].$$

- (1) Write down the gradient of the single-data surrogate loss $J^{(i)}$ (4 points.):

$$\frac{\partial J^i}{\partial a} = \exp[-y^i(a + bx^i)] \cdot (-y^i)$$

$$\frac{\partial J^i}{\partial b} = \exp[-y^i(a + bx^i)] \cdot (-y^i)(x^i)$$

$$\nabla J^i = \left[\frac{\partial J^i}{\partial a} \quad \frac{\partial J^i}{\partial b} \right]$$

- (2) Give one advantage of stochastic gradient descent over batch gradient (2 points.):

SGD is much faster at "curly" optimization, since it performs many more parameter updates per epoch.

- (3) Give pseudocode for a stochastic gradient descent function `theta = train(X, Y)`, including all necessary elements for it to work. (4 points.)

Init $\theta = \emptyset$, or at random

Set step size $\alpha = .001$ (or something)

for epoch in $1 \dots \text{max}$:

for each data point i , in some order (eg. random)

$\theta \leftarrow \theta - \alpha \nabla J^i$

- we can use fancier stopping criteria, or alter the step size as we go, but this is enough to function.

This page is intentionally blank, use as you wish.

Name:

ID#:

This page is intentionally blank, use as you wish.

This page is intentionally blank, use as you wish.

Name:

ID#:

This page is intentionally blank, use as you wish.

This page is intentionally blank, use as you wish.