CS 178 Midterm Exam

Machine Learning and Data Mining: Fall 2019

Monday November 4th, 2019

Your name:	**Row/Seat Number:**
Solutions	Stage
Your ID #(e.g., 123456789)	**UCINetID (e.g.ucinetid@uci.edu)**
314159265	panteater@uci.edu

- Please put your name and ID **on every page**.
- Total time is 50 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

Total, *(52 points.)*

Name: ID#:

Bayes Classifiers, *(10 points.)*

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

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. Each feature can take on one of three values, $x_i \in \{a, b, c\}$.
In the case of a tie, we will prefer to predict class $y = 0$.

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

$p(y = 0)$: $1/3$ $p(y = 1)$: $2/3$

$p(x_1 = a \,|\, y = 0)$: ϕ $p(x_1 = a \,|\, y = 1)$: $2/3$

$p(x_1 = b \,|\, y = 0)$: $2/3$ $p(x_1 = b \,|\, y = 1)$: $1/6$

$p(x_1 = c \,|\, y = 0)$: $1/3$ $p(x_1 = c \,|\, y = 1)$: $1/6$

$p(x_2 = a \,|\, y = 0)$: ϕ $p(x_2 = a \,|\, y = 1)$: $1/3$

$p(x_2 = b \,|\, y = 0)$: $2/3$ $p(x_2 = b \,|\, y = 1)$: $1/3$

$p(x_2 = c \,|\, y = 0)$: $1/3$ $p(x_2 = c \,|\, y = 1)$: $1/3$

(2) Using your naïve Bayes model, what value of y would you predict given $(x_1 = a, x_2 = b)$?: *(3 points.)* Since $p(x_1 = a \,|\, y = 0) = 0$, we would predict $y = 1$.

(3) Using your naïve Bayes model, compute the probabilities: *(3 points.)*
$p(y = 0|x_1 = b, x_2 = c)$: $2/3$ $p(y = 1|x_1 = b, x_2 = c)$: $1/3$

$$p(y=1|x) = \frac{p(y=1)\,p(x_1=b|y=1)\,p(x_2=c|y=1)}{(\ldots) + p(y=0)\,p(b|0)\,p(c|0)} = \frac{2/3 \cdot 1/6 \cdot 1/3}{2/3 \cdot 1/6 \cdot 1/3 + 1/3 \cdot 2/3 \cdot 1/3}$$

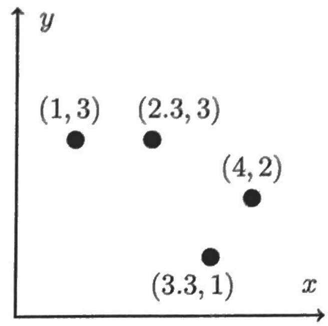
$$= \frac{4 \cdot 1 \cdot 2}{4 \cdot 1 \cdot 2 + 2 \cdot 4 \cdot 2} = \frac{8}{8+16} = 1/3.$$

Name: ID#:

Nearest Neighbor Regression, *(12 points.)*

For a regression problem to predict y given a scalar feature x, we observe training data (pictured at right):

	x	y
(a)	1	3
(b)	2.3	3
(c)	3.3	1
(d)	4	2



(1) Compute **training** MSE of a 1-nearest neighbor predictor. *(3 points.)*

ϕ

(2) Compute the **leave-one-out** cross-validation error (MSE) of a 1-nearest neighbor predictor. *(3 points.)*

Leave out	Predict	Fold MSE
(a)	3	ϕ
(b)	1	2^2
(c)	2	1^2
(d)	1	1^2

$\Rightarrow \frac{1}{4}(4+1+1) = 1\frac{1}{2}$.

(3) Compute the **training** MSE of a 2-nearest neighbor predictor. *(3 points.)*

Data Point	Predict	Err
(a)	3	0
(b)	2	1
(c)	1.5	$1/2$
(d)	1.5	$1/2$

$\Rightarrow \frac{1}{4}(1^2 + \frac{1}{2}^2 + \frac{1}{2}^2) = 3/8$

(4) Compute the **leave-one-out** cross-validation error (MSE) of a 2-nearest neighbor predictor. *(3 points.)*

Leave out	Predict	MSE
(a)	2	1^2
(b)	2	1^2
(c)	2.5	$(1\frac{1}{2})^2$
(d)	2	ϕ

$\Rightarrow \frac{1}{4}(1 + 1 + 9/4) = 1\,1/16$

Name: ID#:

True/False, *(10 points.)*

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

True or **false**: In a soft-margin SVM (i.e., loss $\sum_j w_j^2 + R \sum_i \epsilon^{(i)}$), increasing the value of R will make the model more likely to overfit.

True or **false**: A soft-margin SVM model is harder to optimize than a hard-margin SVM, since it is not a quadratic program.

True or **false**: A kernel SVM will be more efficient than a linear SVM when the number of training data, m, is large.

True or **false**: Applying "early stopping" by increasing the convergence tolerance in SGD increases the bias of the learner to reduce overfitting.

True or **false**: When training a perceptron using the logistic negative log-likelihood loss, gradient descent can never become stuck in a local optimum.

True or **false**: Given sufficently many data m, the 1-nearest neighbor classifier error rate approaches the Bayes optimal error rate.

True or **false**: Stochastic gradient descent is often preferred over batch when the number of data points m is very large.

True or **false**: For a perceptron, increasing the regularization penalty of a linear regression model will decrease the resulting model's variance.

True or **false**: For a perceptron, doubling the number of training data available will decrease the resulting model's bias.

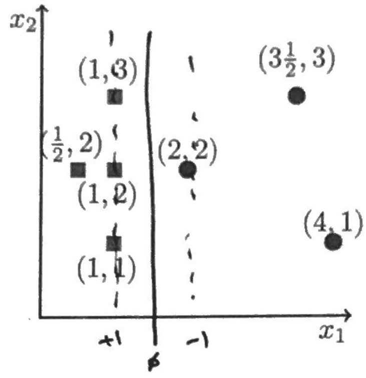
True or **false**: For a perceptron, using $2 \times n$ features per data point by adding n random values to each will increase the resulting model's variance.

Name: ID#:

Support Vector Machines, *(10 points.)*

Suppose we are learning a linear support vector machine with two real-valued features x_1, x_2 and binary target $y \in \{-1, +1\}$. We observe training data (pictured at right):

x_1	x_2	y
0.5	2	+1
1	1	+1
1	2	+1
1	3	+1
2	2	-1
3.5	3	-1
4	1	-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.)* Sketch above

SVs: (1,3) (1,2) (1,1) (2,2)

(2) Derive the parameter values w_1, w_2, b of this $f(x)$ using these support vectors. What is the length of the margin? *(3 points.)*

$$\begin{aligned} w_1 \cdot 1 + w_2 \cdot 2 + b &= +1 \\ w_1 \cdot 1 + w_2 \cdot 1 + b &= +1 \\ w_1 \cdot 2 + w_2 \cdot 2 + b &= -1 \end{aligned} \quad \Rightarrow \quad \begin{aligned} w_1 &= -2 \\ w_2 &= \phi \\ b &= +1 \end{aligned}$$

Margin width $= \frac{2}{\sqrt{w_1^2 + w_2^2}} = 1$ (or same, by inspection).

(3) What is the *training error* of a linear SVM on these data? *(2 points.)*

ϕ

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

All leave-one-out folds result in the same boundary, except (2,2)

For (2,2), it shifts right, to (at least) 2.25 ⇒ incorrect prediction

⇒ 1/7.

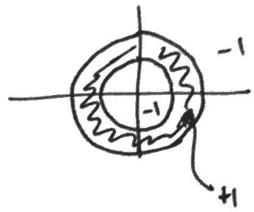
Name: ID#:

VC-Dimensionality, *(10 points.)*

Consider the VC dimension of two classifiers defined using two features x_1, x_2.

(1) First, consider a simple classifier f_A that predicts class $+1$ within a ring with inner radius r and a width of w:

$$f_A(x) = \begin{cases} +1 & (r < (x_1^2 + x_2^2)^{1/2} < r + w) \\ -1 & \text{otherwise} \end{cases}$$

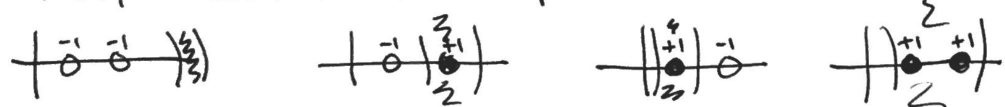


Show that this classifier has VC dimension 2. *(5 points.)*

It only matters how far x is from the origin, so let $x_2 = 0$.

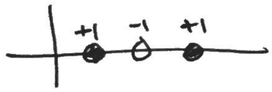
Take any two points; say $x_1^{(1)} = 1$ and $x_1^{(2)} = 1.75$.

Then, we can shatter these two points:



Ex: $r > 1.75$, $w = 1$ | $r = 1.25$, $w = 1$ | $r = .25$, $w = 1$ | $r = .9$, $w = 1$

But, we can never predict the three-point pattern



no matter what the precise x_1 values - if both +1's inside the ring, the middle point must also be.

(2) Now, suppose that we fix $w = 1$, i.e., it is no longer a parameter of the model:

$$f_B(x) = \begin{cases} +1 & (r < (x_1^2 + x_2^2)^{1/2} < r + 1) \\ -1 & \text{otherwise} \end{cases}$$

What is the VC dimension of f_B? Justify your answer. *(5 points.)*

Still 2 - the proof for part (1) used $w=1$, so w is not necessary to shattering two points.

Note: if you can change w, your classifier is a bit more flexible, and you can be less careful with the exact values of $x^{(1)}$ and $x^{(2)}$, but it doesn't change how <u>many</u> points can be shattered.