CS273A Midterm Exam

Introduction to Machine Learning: Winter 2019

Tuesday February 12th, 2019

Your name:	**Row/Seat Number:**
Solutions	Lecture
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 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 Bayes Classifiers, *(10 points.)* **3**

2 Linear and Nearest Neighbor Regression, *(12 points.)* **5**

3 Multiple Choice, *(12 points.)* **7**

4 Support Vector Machines, *(10 points.)* **9**

5 Gradient Descent, *(10 points.)* **11**

6 VC-Dimensionality, *(10 points.)* **13**

Total, *(64 points.)*

Name: ______ ID#: ______

Bayes Classifiers, *(10 points.)*

x_1	x_2	y
a	b	0
b	c	0
b	c	0
c	c	0
a	c	1
a	b	1
b	a	1
b	b	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.)*

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

$p(x_1 = a \,|\, y=0)$: 1/4 $p(x_1 = a \,|\, y=1)$: 1/2

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

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

$p(x_2 = a \,|\, y=0)$: ϕ $p(x_2 = a \,|\, y=1)$: 1/4

$p(x_2 = b \,|\, y=0)$: 1/4 $p(x_2 = b \,|\, y=1)$: 1/2

$p(x_2 = c \,|\, y=0)$: 3/4 $p(x_2 = c \,|\, y=1)$: 1/4.

(2) Using your naïve Bayes model, compute: *(3 points.)*
$p(y = 0|x_1 = a, x_2 = c)$: 3/4 $p(y = 1|x_1 = a, x_2 = c)$: 1/4

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

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

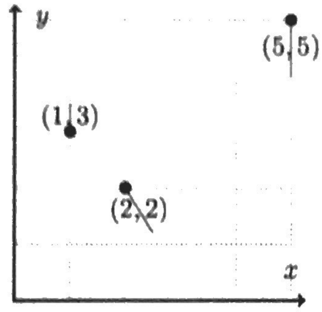
(3) Compute the probabilities $p(y = 0|x_1 = a, x_2 = c)$ and $p(y = 1|x_1 = a, x_2 = c)$ for a **joint** (not naïve) Bayes model trained on the same data. *(3 points.)*

$p(y=0|a,c) = \phi$ $p(y=1|a,c) = 1.$

Name: ID#:

Linear and Nearest Neighbor Regression, *(12 points.)*

Consider the data points shown at right, for a regression problem to predict y given a scalar feature x.



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

$\emptyset$.

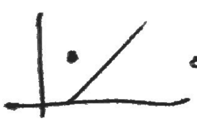
(2) Compute the **leave-one-out** cross-validation error (MSE) of a 1-nearest neighbor predictor. *(3 points.)*

$\frac{1}{3}\left[1^2 + 1^2 + 3^2\right] = 11/3.$

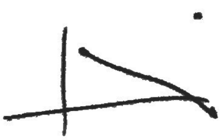
(3) Compute the **leave-one-out** cross-validation error (MSE) of a 2-nearest neighbor predictor. *(3 points.)*

$1/3 \left[1/2^2 + 2^2 + 2\frac{1}{2}^2 \right] = 7/2.$

(4) Compute the **leave-one-out** cross-validation error (MSE) of a linear regressor, e.g., a model of the form $f(x) = \theta_0 + \theta_1 x$. *(3 points.)*

 $\to 2^2$ $\to 1/3\left[2^2 + 1\frac{1}{2}^2 + 6^2\right] = \frac{457}{12} = 38\ 1/12.$

 $\to (1\frac{1}{2})^2$

 $\to 6^2$

Name: ID#:

Multiple Choice, *(12 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 choices below, will it likely increase, decrease, or have no effect on overfitting (circle your choice)? If you think it is equally likely to go either way, pick *No Effect.*

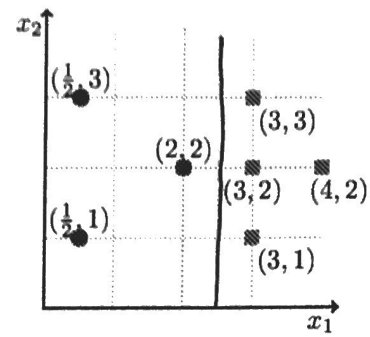
1	Gathering more labeled training data	**Reduce**	**Increase**	**No Effect**
2	For a linear regressor, use $2 \times m$ training data by adding m all-zero (x and y) data points.	**Reduce**	**Increase**	**No Effect**
3	For a linear regressor, use $2 \times n$ features per data point by adding n random values to each.	**Reduce**	**Increase**	**No Effect**
4	For a linear regressor, use $2 \times n$ features per data point by adding n all-zero features to each.	**Reduce**	**Increase**	**No Effect**
5	For a linear regressor, increasing the $L2$ regularization penalty	**Reduce**	**Increase**	**No Effect**
6	For a 3-nearest neighbor classifier, use $2 \times m$ training data by copying (duplicating) each data point.	**Reduce**	**Increase**	**No Effect**
7	For a 3-nearest neighbor classifier, use $2 \times n$ features per data point by copying (duplicating) the features.	**Reduce**	**Increase**	**No Effect**
8	For a k-nearest neighbor classifier, rescaling the data to zero mean, unit variance	**Reduce**	**Increase**	**No Effect**
9	For a neural network model, increasing the number of hidden nodes in the first layer	**Reduce**	**Increase**	**No Effect**
10	For a neural network, changing the activation function of the hidden nodes from logistic (sigmoid) to rectified linear (ReLU).	**Reduce**	**Increase**	**No Effect**
11	Switching from linear to polynomial Kernel SVMs	**Reduce**	**Increase**	**No Effect**
12	Using gradient descent to optimize our SVM model, rather than a QP (quadratic program) solver.	**Reduce**	**Increase**	**No Effect**

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	1	-1
2	2	-1
0.5	3	-1
3	2	+1
3	1	+1
3	3	+1
4	2	+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.)* (2,2) ; (3,3), (3,2), (3,1).

(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.)*

$w_1 \cdot 2\frac{1}{2} + w_2 \cdot (x_2) + b = \emptyset$. ⇒ $w_2 = \emptyset$.

$w_1 \cdot 2 + b = -1$

$w_1 \cdot 3 + b = +1$ ⇒ $w_1 = 2$

$b = -3$.

$M = \frac{2}{\sqrt{w_1^2 + w_2^2}} = 1$

(or by inspection).

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

$\emptyset$.

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

1/7

point (2,2) left out ⇒ boundary shifts & mispredicts.

others - boundary is stable.

Name: ID#:

Gradient Descent, *(10 points.)*

Suppose that we have training data $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$, 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 decide to use gradient descent on a smooth surrogate loss called the *exponential loss*:

$$J(X, Y) = \frac{1}{m} \sum_{i=1}^{m} \exp\big(-y^{(i)}(a + bx^{(i)})\big) \qquad (*)$$

(1) Write down the gradient of our surrogate loss function.

$$\frac{\partial J}{\partial a} = \frac{1}{m} \sum_i \exp(-y^i(a+bx^i)) \cdot (-y^i)$$

$$(**)$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_i \exp(-y^i(a+bx^i)) \cdot (-y^i x^i)$$

$$\nabla J = \left[\frac{\partial J}{\partial a} \;\; \frac{\partial J}{\partial b} \right].$$

(2) Give one advantage of batch gradient descent over stochastic gradient

Easier to monitor convergence

Monotonic descent

(Also easier to set step size schedule, etc) & more...

(3) Give one advantage of stochastic gradient descent over batch gradient

Faster for large data sets (m), particularly early in optimization

Often avoids shallow local minima

⋮

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

Initialize θ to something (zero, random, etc).

Set step size α, stopping tolerance ϵ

Init $J^{old} = \infty$, $J = \infty$.

while $(|J - J^{old}| > \epsilon)$ { // or some other stopping criterion

$\theta \leftarrow \theta - \alpha \nabla J.$ // ∇J in $(**)$.

$J^{old} = J$

$J = \frac{1}{m}\sum \ldots$ // $(*)$ def of J.

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:

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

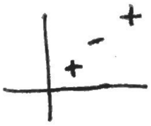
Show that this classifier has VC dimension 2. *(5 points.)*

It only matters what each point's radius, $r^i = (x_1^i)^2 + (x_2^i)^2$ is; our two (or 3) points will be located at $(x_1^1)^2 + (x_2^1)^2 = r^1 < r^2 = (x_1^2)^2 + (x_2^2)^2 < r^3 = (x_1^3)^2 + (x_2^3)^2$.

2 points:

$r<0$, $r^1<w<r^2$ $\quad$ $r^1<r<r^2<r+w$.

3 points: the learner cannot shatter b/c:



$r < r^1 < r+w < r^2$

but then point at r^3 is mis predicted.

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

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

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

It turns out, this does not change the VC dimension (still 2).

Place the points so that $r^2 - r^1 < 1$. Then:

(not to scale) :)