CS273a Midterm Exam

Introduction to Machine Learning: Fall 2016

Tuesday November 1st, 2016

Your name: SOLUTIONS

Your ID # and UCINetID (e.g., 123456789, myname@uci.edu):

Your seat (row and number):

- 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.
- Turn in any scratch paper with your exam

(This page intentionally left blank)

Problem 1: (10 points) Bayes Classifiers

In this problem you will use Bayes Rule: $p(y|x) = p(x|y)p(y)/p(x)$ to perform classification. Suppose we observe some training data with two binary features x_1, x_2 and a binary class y. After learning the model, you are also given some validation data.

Table 1: Training Data

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

Table 2: Validation Data

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

In the case of any ties, we will prefer to predict class 0.

(a) Give the predictions of a joint Bayes classifier on the validation data. What is the validation error rate?

$\hat{y}$:

00 ⇒ 0 ✗

01 ⇒ 1 ✗

10 ⇒ 1 ✓

11 ⇒ 0 ✓

⇒ validation error rate = 1/2.

(b) Give the predictions of a naïve Bayes classifier on the validation data. What is the validation error rate?

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

$p(x_1=1 \mid y=0) = 1/2$ $\quad p(x_1=1 \mid y=1) = 1/2$

$p(x_2=1 \mid y=0) = 3/4$ $\quad p(x_2=1 \mid y=1) = 1/2$

⇒ $\hat{y}$:

00 ⇒ 1

01 ⇒ 0

10 ⇒ 1

11 ⇒ 0

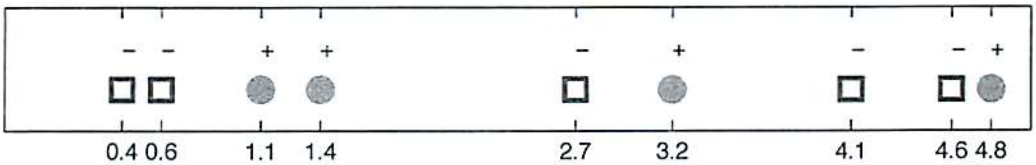
⇒ validation error rate = $\emptyset$.

(c) **True** or **False**: In a naïve Bayes model, the features x_i are independent, i.e., $p(x_1, x_2) = p(x_1)p(x_2)$.

The features are conditionally independent, $p(x_1 x_2 | y) = p(x_1|y)\, p(x_2|y)$

but not independent.

Problem 2: (9 points) Nearest Neighbor Classification



Given the above data with one scalar feature x (whose values are given below each data point) and a class variable $y \in \{-1, +1\}$, with filled circles indicating $y = +1$ and squares $y = -1$ (the sign is also shown above each data point for redundancy), we use a k-nearest neighbor classifier to perform prediction; in the case of ties, we prefer to predict class -1. Answer the following:

(a) Compute the training error rate of a 1-Nearest-Neighbor classifier trained on these data.

ϕ

(b) Compute the leave-one-out cross-validation error rate of a 1-Nearest-Neighbor classifier on these data.

check nearest non-self point.

✓✓ ✓✓ × × ✓ × ×

$\Rightarrow 4/9$

(c) Compute the training error for a 3-Nearest-Neighbor classifier on these data.

check nearest 3 points, including self.

✓✓ ✓✓ × × ✓✓ ×

$\Rightarrow 3/9$.

Problem 3: (10 points) Gradient Descent

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 use gradient descent on a smooth surrogate loss called the *exponential loss*:

$$J(X, Y) = \frac{1}{m} \sum_i \exp\left(y^{(i)}(a + bx^{(i)})\right)$$

Note - this loss has a typo; it should be $\exp(-y^i(a+bx^i))$

we'll take the gradient of the loss as given here.

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

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

$$\frac{\partial J}{\partial a} = \frac{1}{m} \sum_i \exp\left[y^{(i)} (a + b x^{(i)}) \right] \cdot y^{(i)}$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_i \exp\left[y^{(i)} (a + b x^{(i)}) \right] y^{(i)} x^{(i)} .$$

(b) Give one advantage of batch gradient descent over stochastic gradient.

Easier to test for convergence

Easy to make monotonic

⋮

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

Init $\theta = [a, b]$ (random, zero, etc).

Select step size α

while ($\neg$done) {

  Compute ∇J as in part (a)

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

  check if done, eg: $\|\nabla J\| < \epsilon$; # of iterations $> T$; etc.

}

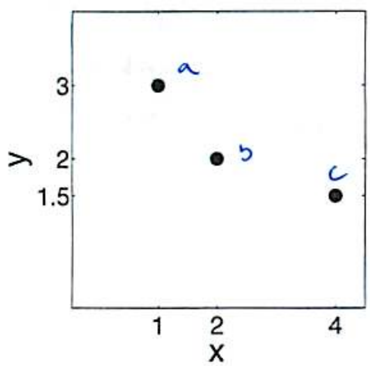
return θ.

Problem 4: (10 points) Linear Regression, Cross-validation

Consider the following data points, copied in each part. We wish to perform linear regression to minimize the mean squared error (MSE) of our predictions.

(a) Compute the **leave-one-out** cross-validation error of a zero-order (constant) predictor,

$$\hat{y}(x) = \theta_0$$



Best constant predictor: mean of the training data.

Leave out:

a ⇒ $\theta_0 = 1.75$

b $= 2.25$ ⇒

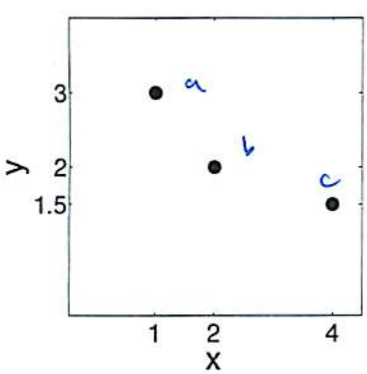
c $= 2.5$

MSE $= (1.25)^2 + (.25)^2 + (1)^2$

⇒ Xval MSE $= \frac{1}{3}\left[(5/4)^2 + (1/4)^2 + (1)^2\right] = \frac{1}{3}\left(\frac{42}{16}\right) = 7/8.$

(b) Compute the **leave-one-out** cross-validation error of a first-order (linear) predictor,

$$\hat{y}(x) = \theta_0 + \theta_1 x$$



Leave one out ⇒ best line for remaining two points:

a ⇒ $\hat{y} = 2.25$

b ⇒ $= 2.5$ ⇒

c ⇒ $= \emptyset$

MSE $= (.75)^2$, $(.5)^2$, $(1.5)^2$

Xval MSE $= \frac{1}{3}\left((3/4)^2 + (2/4)^2 + (6/4)^2\right)$

$= 49/48.$

Problem 5: (20 points) Multiple Choice

For the following questions, assume that we have m data points $y^{(i)}$, $x^{(i)}$, $i = 1 \ldots m$, each with n features, $x^{(i)} = [x_1^{(i)} \; \ldots \; x_n^{(i)}]$.

Circle one answer for each:

Suppose that we are training a linear classifier (perceptron). Before training, we decide to remove (throw away) 10% of our features (selected at random). This is most likely to make it **more** **equally** **less** likely to overfit the data.

When training a k-nearest neighbor model, we decide to increase the value of k. This will most likely make our model **more** **equally** **less** likely to overfit the data.

Again, training a k-nearest neighbor model, we double the amount of data available to the model. We then re-train the model, including re-optimizing k.
This is likely to **increase** **not change** **decrease** the bias.

Suppose that, when training a linear regressor, we double the amount of data available for training. This is most likely to decrease the **bias** **variance** **both** **neither** of our learned model.

Still training a linear regressor, instead of providing more real data, we instead include m additional points of "fake" data, $(x^{(i)}, y^{(i)}) = (0, 0)$.
This will most likely **increase** **not change** **decrease** the bias.
It will most likely **increase** **not change** **decrease** the variance.

True or **false**: if the VC dimension of a model is H, then the model can shatter any set of H training points.

True or **false**: Linear regression can be solved using either matrix algebra or gradient descent.

True or **false**: Increasing the regularization of a linear regression model will decrease the variance.

Before training a linear classifier, we transform one of our features by taking its logarithm, i.e., `X[:,1] = np.log(X[:,1]);`. This is likely to **increase** **not change** **decrease** the model's VC dimension.

We train a Gaussian Bayes classifier, but then decide to re-train it, forcing the two classes' covariance matrices to be equal, i.e., $\Sigma_{(y=+1)} = \Sigma_{(y=-1)}$. This is likely to **increase** **not change** **decrease** the variance of our model.

Problem 6: (9 points) Short answer

Consider the two possible decision boundaries (indicated by Line 1 and Line 2) for the binary classification problem shown in Figure 1. For each algorithm below, will it possibly produce boundary 1, boundary 2, or both? Please give a concise explanation of your choice.

Perceptron Algorithm :

1 or 2 - both lines separate the data & so the perceptron algorithm will not update them.

Logistic Regression :

1 only - assigns higher likelihood to the data than L2; objective is convex (no local optima), so if reasonably optimized it will not produce L2.

Support Vector Machine (hard-margin) :

1 only - maximum margin.

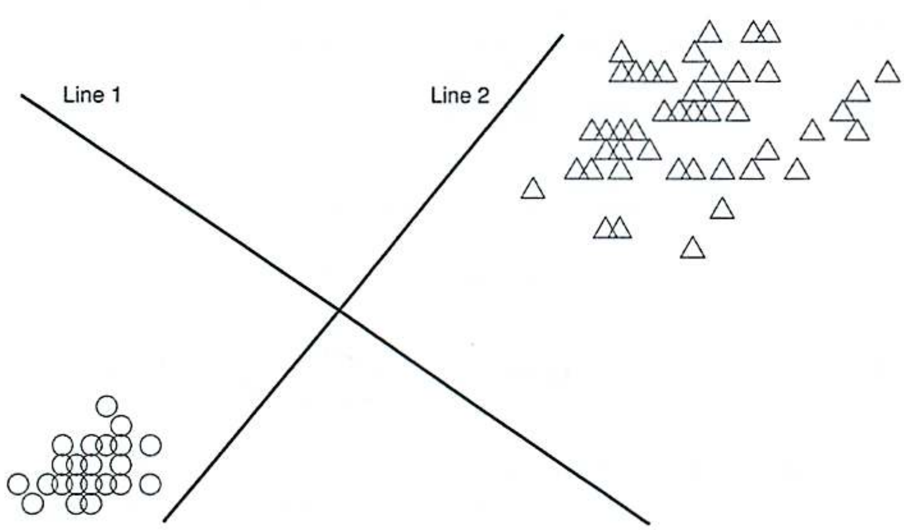


Figure 1: Possible linear decision boundaries.

Problem 7: (10 points) Support Vector Machines

Suppose we are learning a linear support vector machine with a single scalar feature x and binary target $y \in \{-1, +1\}$. We observe training data:

$$D = \{(x^{(i)}, y^{(i)})\} = \{(0, +1), (-3, +1), (1, -1)\}$$

Our linear classifier takes the form $f(x; a, b) = \text{sign}(ax + b)$.

(a) Write down the primal optimization problem for a support vector machine on these data.

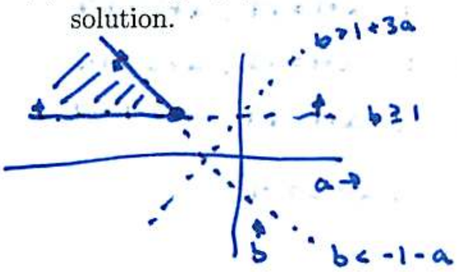
$\min a^2$

ST $\quad a \cdot \cancel{0} + b \geq +1$

$a \cdot -3 + b \geq +1$

$a \cdot 1 + b \leq -1$

(b) Sketch (graph) the constraint set on the parameters a, b, and give the values of a, b at the solution.



Solution is $b = 1$

$a + b = -1 \Rightarrow a = -2$

(c) Identify the support vectors.

SVs: (0,+1) and
(1,-1)

(d) Give **two** possible advantages of the *dual* form of the SVM over the primal.

Easy to initialize to a valid (feasible) point

Can use kernel similarity f'n, ~~compared~~ equivalent to large or infinite # of features

May be more efficient if m small and n is large
(# data) (# features)

Problem 8: (10 points) VC Dimension

Consider the following classifier, parameterized by a single scalar parameter a and operating on a scalar feature x:

$$f(x\ ;\ a) = \begin{cases} +1 & x \le a \text{ or } a+1 < x \le a+2 \\ -1 & \text{otherwise} \end{cases}$$

In this problem, we will show the VC dimension of $f(x\ ;\ a)$ is 3.

(a) Show by example that $f(x\ ;\ a)$ can shatter three points. Hint: place your points at $x^{(1)} = 0$, $x^{(2)} = 0.75$, $x^{(3)} = 1.5$.

Check each pattern:

$+ + + \Rightarrow a > 1.5$

$+ + - \Rightarrow a = 1$

$+ - + \Rightarrow a = 0.25$

$+ - - \Rightarrow a = 0.7$

$- + + \Rightarrow a = -0.3 \quad : a+1 = .7,\ a+2 = 1.7$

$- + - \Rightarrow a = -0.7 \quad : a+1 = .3,\ a+2 = 1.3$

$- - + \Rightarrow a = -0.1 \quad : a+1 = .9 \quad a+2 = 1.9$

$- - - \Rightarrow a = -3 \quad : a+2 = -1$

(b) Argue that $f(x\ ;\ a)$ cannot shatter four points. (Which target pattern cannot be reproduced?)

It cannot ~~shatter~~ produce the pattern $- + - +$ $\Rightarrow$ cannot shatter 4 points.

Order the points by their x value.

1st point, $y^{(1)} = -1 \Rightarrow x^{(1)} > a$

2nd $y^{(2)} = +1 \Rightarrow x^{(2)} > a+1$

$y^{(3)} = -1 \Rightarrow x^{(3)} > a+2$

But then $y^{(4)} = +1$ cannot be predicted, because $x^{(4)} > x^{(3)} > a+2$

$\Rightarrow \hat{y}(x^{(4)}) = -1.$