CS178 Midterm Exam

Machine Learning & Data Mining: Winter 2011

Thursday February 3rd, 2011

Your name: Alex Ihler

SOLUTIONS

Name of the person in front of you (if any):

Name of the person to your right (if any):

- Total time is 1:15. 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 myself or the TA to come over.
- Turn in any scratch paper with your exam.

Scores:

P1: ________

P2: ________

P2: ________

P4: ________

P5: ________

P6: ________

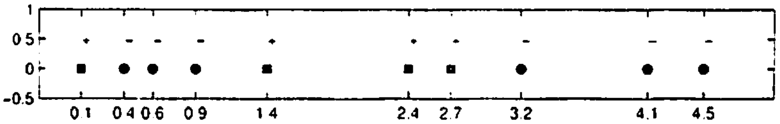
P7: ________

Total: ________________

(This page intentionally left blank)

Problem 1: (16 points) Classification in One Dimension

We observe a collection of training data with one feature. "x" and a class label $c \in \{-, +\}$, shown here: class + is indicated by squares and − by circles, and also labelled with text. Answer each of the following questions. Express error rates as the fraction of data points incorrectly classified.



(a) What is the best training error rate we can achieve on these data from a Gaussian model Bayes classifier with *equal* variances for the two classes? Explain briefly (sketch + 1-2 sentences): how is it achieved and why is it the best?

Equal variances ⇒ decision boundary is linear (a point in 1D)

Data □ ○○○ □ □□ | ○ ○○ ⇒ error rate 3/10

Predict: + −

(b) What is the best training error we can achieve from a Gaussian Bayes classifier with *arbitrary* variances for the two classes? Explain briefly how it's achieved and why it's the best.

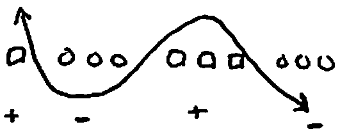
Non-equal variances ⇒ quadratic decision boundary ⇒ 2 points in 1D.

Data: □ ○○○ (□ □ □) ○ ○○ ⇒ error rate 1/10

Predict: − + −

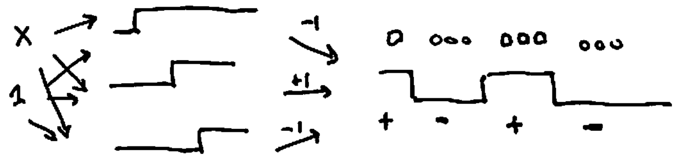
(c) What is the best training error we can achieve from a perceptron classifier with polynomial features? Explain briefly how it's achieved and why it's the best.

With a cubic polynomial, we can get 0/10 training error:



(d) What is the best training error we can achieve from a two-layer neural network (multi-layer perceptron) with input features "x" and "1"? Explain briefly how it's achieved (e.g., # of hidden nodes & what they look like) and why it's the best.

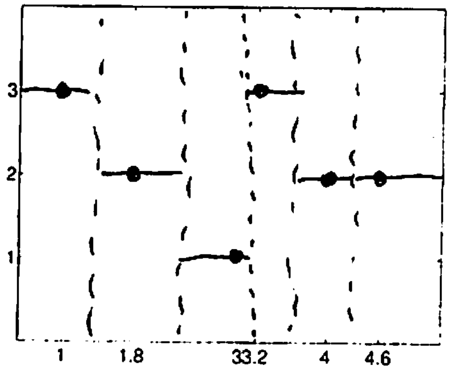
With 3 hidden nodes we can get 0/10 training error



⇒ 0/10 errors.

Problem 2: (12 points) kNN Regression

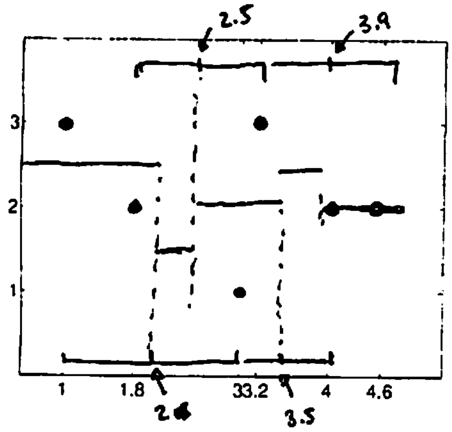
Consider a regression problem for predicting the data shown at left using your k-nearest neighbor regression algorithm from the homework. Under each of the following scenarios, (a) sketch the regression function when trained on all the data; (b) compute its resulting training error (MSE). (If you like you may leave an arithmetic expression, i.e., leave values as for example "$(.6)^2$")



(a) $k = 1$

Draw lines midway between each point
Within these bins our prediction is constant.

Training error 0/10. (data are memorized)

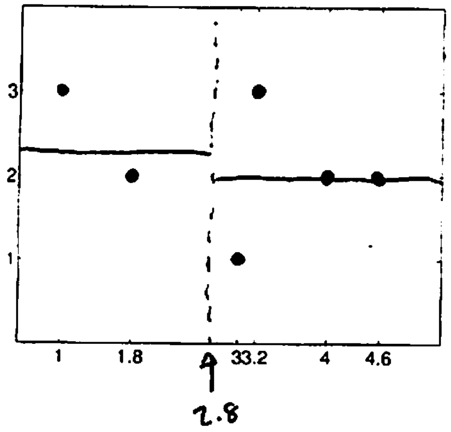


(b) $k = 2$

Find mid point of data pairs

& Training error =

$$\frac{1}{10}\left[(.5)^2 + (.5)^2 + (1)^2 + (1)^2 + (0)^2 + (0)^2\right]$$



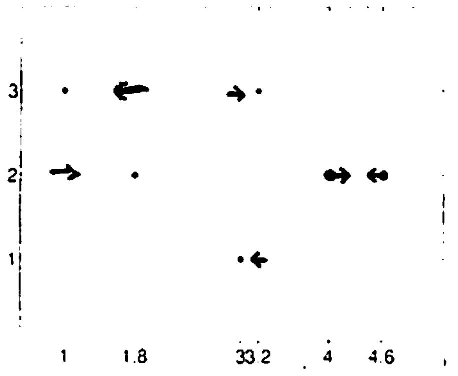
(c) $k = 5$

Find mid point between points 1 & 6
1st bin is avg of 1st 5 points
2nd bin is avg of last 5 points.

$$\frac{1}{10}\left[(4/5)^2 + (1/5)^2 + (1)^2 + (1)^2 + 0^2 + 0^2\right]$$

Problem 3: (10 points) Cross-validation Errors

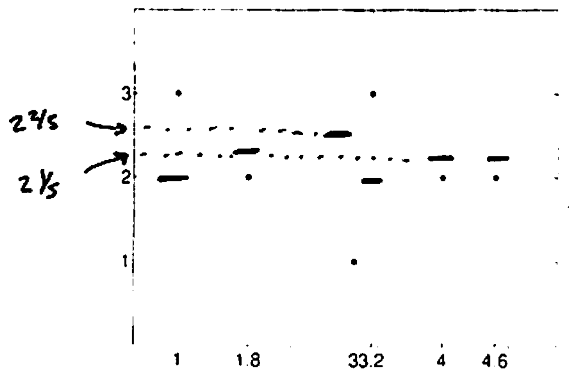
Using the same data, compute the *leave-one-out cross-validation* MSE error rate (i.e., the 6-fold cross-validation error rate). Use the figures to sketch the predicted value at each training point *when it is left out*. Again, you may leave un-evaluated arithmetic expressions.



(a) $k = 1$

When each data point is left out, our prediction at that point is the nearest of its neighbors. (See sketch)

LOO XV error $=\frac{1}{6}\left[(1)^2+(1)^2+(2)^2+(2)^2+(0)^2+(0)^2\right]$



(b) $k = 5$

When each data point is left out, there are only 5 points remaining ⇒ prediction is the average of those data.

(See sketch)

LOOXV error $=\frac{1}{6}\left[(1)^2+(.2)^2+(1.4)^2+(1)^2+(.2)^2+(.2)^2\right]$

Problem 4: (16 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 the following training data, with three binary features $x_1, \ldots, x_3$ and a binary class y:

x_1	x_2	x_3	y
1	0	0	0
0	1	1	0
0	0	1	0
1	0	0	1
0	0	1	1
0	1	0	1
1	1	0	1

$p(y=0) = 3/7$

$p(x_1=0 | y=0) = 2/3$ $\qquad$ $p(x_1=0 | y=1) = 1/2$

$p(x_2=1 | y=0) = 1/3$ $\qquad$ $p(x_2=1 | y=1) = 1/2$

$p(x_3=0 | y=0) = 1/3$ $\qquad$ $p(x_3=0 | y=1) = 3/4$

Learn to predict y using a naïve Bayes classifier; **show your work.**

(a) After learning the model, what is the predicted probability $p(y = 0|x_1 = 0, x_2 = 1, x_3 = 0)$?

$$p(y=0|\vec{x}) = \frac{p(y=0)\prod p(x_i|y=0)}{\sum_y p(y)\prod p(x_i|y)} = \frac{3/7 \cdot 2/3 \cdot 1/3 \cdot 1/3}{3/7 \cdot 2/3 \cdot 1/3 \cdot 1/3 + 4/7 \cdot 1/2 \cdot 1/2 \cdot 3/4}$$

(b) Suppose that we *only* observe $x_1 = 0$. What is the predicted probability $p(y = 0|x_1 = 0)$?

$$p(y=0|x_1) = \frac{p(y=0)\,p(x_1|y=0)}{\sum_y p(y)\,p(x_1|y)} = \frac{3/7 \cdot 2/3}{3/7 \cdot 2/3 + 4/7 \cdot 1/2} = \frac{2}{2+2} = 1/2 .$$

For the next two parts, learn a *joint* Bayes classifier.

(a) What is the predicted probability $p(y = 0|x_1 = 0, x_2 = 1, x_3 = 0)$?

This can be read almost directly from the table:

$$\frac{\#\, y=0 \;\&\; \vec{x}=\ldots}{\#\, \vec{x}=\ldots} = \frac{\emptyset}{\emptyset+1} = \emptyset .$$

(b) What is $p(y = 0|x_1 = 0)$?

Same argument:

$$\frac{\#\, y=0 \;\&\; x_1=0}{\#\, x_1=0} = \frac{2}{2+2} = 1/2 .$$

(Same as the naïve Bayes, since there is only one feature)

Problem 5: (15 points) Regression

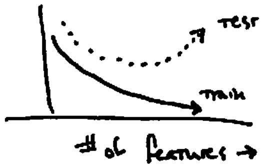
We have a dataset consisting of M examples, each with one feature x and a real-valued target y.

(a) Suppose we use linear regression to fit the data, using D polynomial features. From our full database, we choose M_1 data points (at random) to be a training data set, and M_2 of the remaining data to be a validation (test) set.
Suppose we begin to increase the set of available features by, for example, taking polynomials of x. What do you expect to happen to the mean squared error (1) on the training set and (2) on the test set? Sketch and explain in 1-2 sentences.

Increasing the # of features increases the complexity of our learner.

On the training set, error will go down

test set, error may go down & will eventually increase again



(b) Now suppose that we train a *non-linear* regression model, where our prediction is

$$\hat{y}(x) = \exp(\theta x)$$

with a single, scalar parameter θ.

(1) Write down the formula for the mean squared error on the training data.

$$C = \frac{1}{M_1} \sum_i \left(y^{(i)} - \exp(\theta x^{(i)})\right)^2$$

(2) Suppose that we train our model to minimize the (training) MSE. Which of the following will be true of θ? (Circle one)

i. $\sum_i y^{(i)} = \sum_i x^{(i)} \exp(\theta x^{(i)})$

ii. $\sum_i y^{(i)} \exp(\theta x^{(i)}) = \sum_i x^{(i)} \exp(\theta x^{(i)})$

iii. $\sum_i y^{(i)} x^{(i)} \exp(\theta x^{(i)}) = \sum_i x^{(i)} \exp(\theta x^{(i)})$

[✓ iv. $\sum_i y^{(i)} x^{(i)} \exp(\theta x^{(i)}) = \sum_i x^{(i)} \exp(2\theta x^{(i)})$]

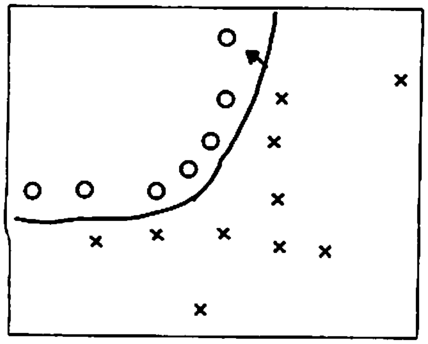
Take the derivative; it will equal zero at the optimal value of θ.

$$\frac{\partial}{\partial \theta} C = \sum (y^i - \exp(\theta x^i)) * \exp(\theta x^i) \cdot x^i$$
$$= 0$$

$$\Rightarrow \sum y^i x^i \exp(\cdot) = \sum x^i \exp(\cdot) \cdot \exp(\cdot)$$
$$= \sum x^i \exp(2\theta x^i)$$

Problem 6: (18 points) Separability

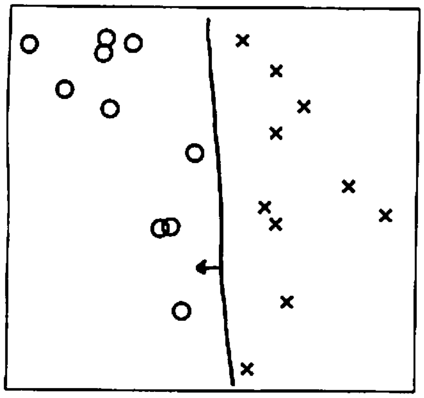
For each of the following examples of training data, sketch a classification boundary that separates the data. Using the original features (x_1, x_2) and a constant feature, state (a) whether the two classes can be exactly separated using *some* Gaussian model Bayes classifier; (b) whether the two classes can be exactly separated by a perceptron classifier, and (c) whether they can be separated using a multi-layer perceptron.



(a) Yes - sketch shows a quadratic boundary, achievable by Gaussians w/ different Σ.

(b) No - perceptron has a linear boundary

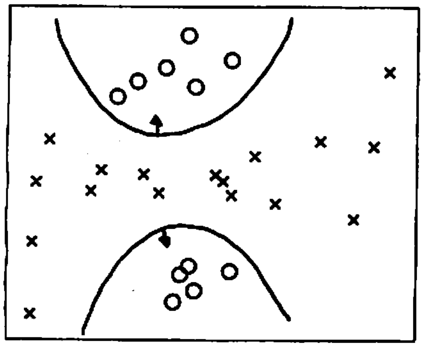
(c) Yes - given enough hidden nodes



(a) Yes - linear boundary

(b) Yes - linear boundary

(c) Yes - even a single layer perceptron works.



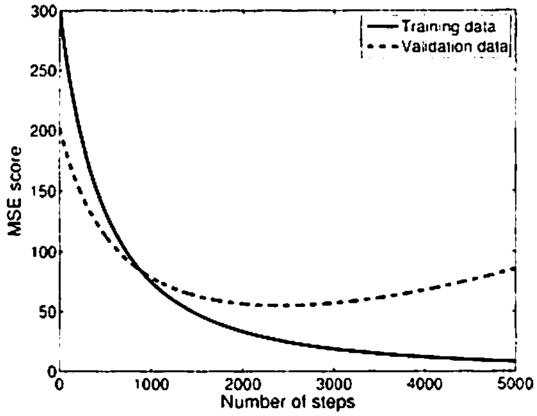
(a) Yes - quadratic boundary shown

(b) No - not linearly separable

(c) Yes - again just add enough hidden units.

Problem 7: (12 points) Training and Test Error

Consider the following plot, which shows the training set error and the validation test set error for a neural network model as it is trained, i.e., the x axis indicates the number of iterations of training (gradient steps). Note that the training error decreases monotonically, while the test error does not.



(a) Explain what is happening and why: suggest a possible solution.

The increase is due to over fitting. We could

(1) use early stopping to reduce ~~&~~ overfitting

(2) regularize

(3) reduce the # of hidden units

⋮

Now suppose that we were to re-train the model with 10 times as much data, while keeping all other aspects (initialization, etc.) the same.

(b) Would you expect the training curve to be different? If so, sketch how it might change.

Not significantly in shape; it might increase a little but will have the same profile.

(c) Would you expect the validation (test) curve to be different? If so sketch how it might change.

It will most likely look flatter to the right (not overfit, or at least not so much).

More data means we will learn a better model, which is more likely to generalize well.