CS178 Intro to Machine Learning
Winter 2010
Final Exam

Instructions:

(1) Write the names of your immediate-front and immediate-right neighbors (if any):

Person in front:

Person to the right:

(2) READ THE EXAM FIRST and organize your time.

(3) Write your name on each sheet of paper

(4) Attach any scratch paper you used during the exam, and your two pages of handwritten notes if any.

Scores

P1 _____

P2 _____

P3 _____

P4 _____

P5 _____

Total: _____

P6 _____

(page intentionally left blank)

Problem 1: Short answer (18 points)

(a) Consider a data set with N examples ($x^{(i)}, y^{(i)}$), both dimensions real-valued. We create a model to predict y: ~~y^{(i)} = w_0 + w_1 x^{(i)} + e_i~~, where e_i is random noise. We train our model to minimize MSE, so that $y^{(i)} = w_0 + w_1 x^{(i)} + e^{(i)}$

$$(\hat{w}_0, \hat{w}_1) = \arg\min \sum_{i=1}^{N} (y^{(i)} - w_0 - w_1 x^{(i)})^2$$

(1) (True) or False (circle one) : We can optimize w0 and w1 in closed form, using matrix inversion. 2

(2) Which one of the following will be true after training our linear regression model? (Circle one)

1. $\sum_{i=1}^{N} (y^{(i)} - w_0 - w_1 x^{(i)}) y^{(i)} = 0$
2. $\sum_{i=1}^{N} (y^{(i)} - w_0 - w_1 x^{(i)}) (x^{(i)})^2 = 0$
3. (3.) $\sum_{i=1}^{N} (y^{(i)} - w_0 - w_1 x^{(i)}) (x^{(i)}) = 0$
4. $\sum_{i=1}^{N} (y^{(i)} - w_0 - w_1 x^{(i)})^2$

3

(b) Give an advantage online gradient descent has over its batch counterpart, and an advantage that batch has over online descent. Many possible:

2 + 2

online
- often faster

batch
- less randomness
- easier to gauge convergence
- strictly decreasing cost fn (for sufficiently small step size)

(c) Suppose we currently believe our model to be overfitting. (For concreteness, suppose it is the linear regression model we used in HW 2, trained to minimize the mean squared error of the training data.) We decide to increase the number of training data used by our learner. Choose one answer for each part.

(1) Training error will most likely [(increase) decrease stay the same] 1

(2) Test error will most likely [increase (decrease) stay the same] 1

(3) The VC dimension of our learner will most likely [increase decrease (stay the same)] 1

Now suppose we instead believe our model to be underfitting. We again increase the number of training data used by our learner. Choose one answer for each part.

(4) Training error will most likely [increase decrease (stay the same)] 1

(5) Test error will most likely [increase decrease (stay the same)] 1

(6) The VC dimension of our learner will most likely [increase decrease (stay the same)] 1

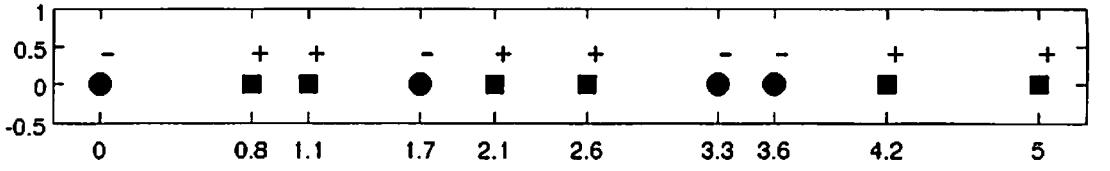
Problem 1, continued:

(d) Recall the Viola-Jones face detection model from class – we extracted ~180,000 pixel-based features from the data, then used boosting to learn a sequence of decision stumps. After hearing the lecture, Abe gets an idea. "A decision stump's just an axis-aligned linear classifier – I bet I could do a little bit better if I used logistic regression within the boosting so I could give a different slope to each linear classifier." He codes it up, but it does miserably. Why?

3

There are too many features – logistic regression will use 180K parameters, one "slope" for each feature, and will badly overfit any moderate data set. The "single feature" assumption makes the learner weak enough to underfit, then boosting is used to make it more complex.

Problem 2: Cross-validation and Nearest Neighbor methods (14 points)



Using the above data with one feature "x" (values marked below data point), and a class variable "y" (positive or negative, pictured above the data point and indicated by a square or circle, respectively), answer the following:

Part 1: Compute the leave-one-out cross-validation error of a 1-Nearest-Neighbor classifier. (Break any ties using the left-to-right ordering.)

4/10

5

Part 2: Compute the leave-one-out cross-validation error of a 3-NN classifier.

× × ✓ × ✓ × × × × ×

8/10

5

Part 3: Compute the leave-one-out cross-validation error of a 9-NN classifier.

4/10

4

Problem 3: K-Means Clustering (18 points)

(a) Execute the k-means algorithm (as accurately as possible) on the following data. Dots are data points, the two squares show the initial cluster centers. In your plots, draw the cluster centers as squares and also show the decision boundary that defines each cluster. (If a cluster has no data associated with it, assume its mean is unchanged at the next iteration.) Use as many pictures as you need for convergence. Also, explain your process and placements during the algorithm.

Explanation: Your sketches & progress may differ slightly - it's the method that matters.

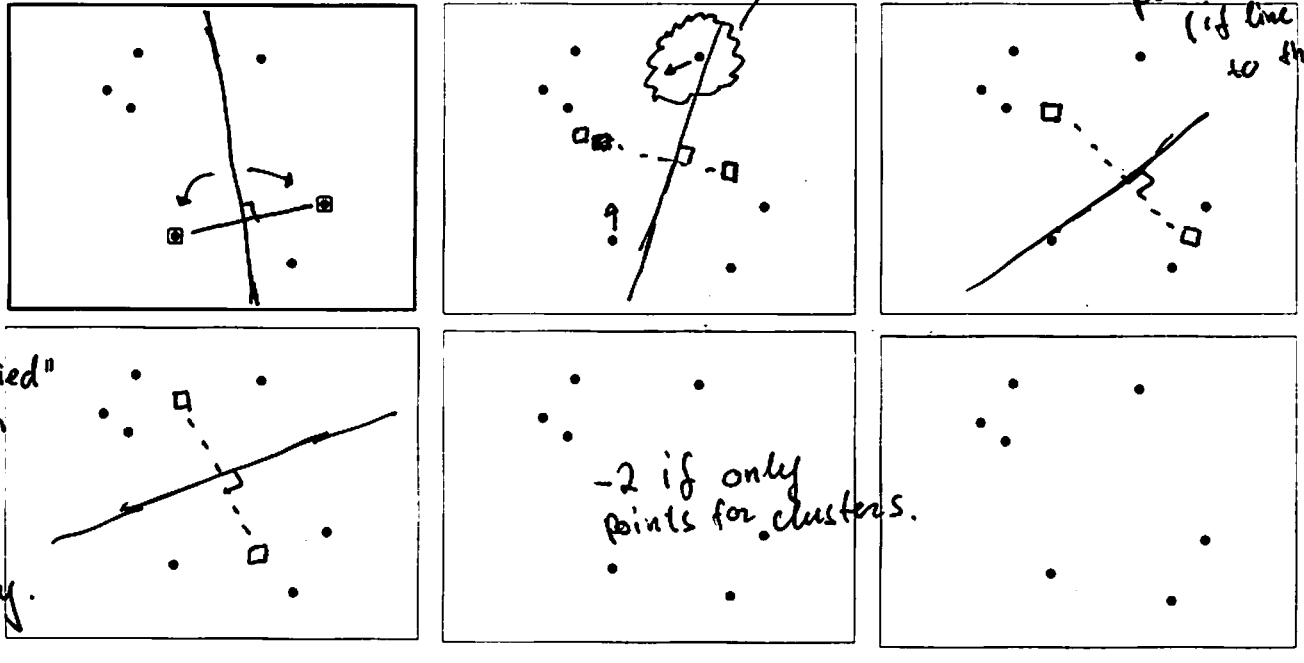
(1) Assign data to nearest cluster. The "decision boundary" between two clusters is a straight line, bisecting the line joining them.

(2) Move to mean of assigned data - should look like the center of mass of the associated points.

8

Repeat until no assignment changes.

location of this point looks questionable to me ↓ wasn't taking any points off for this (if line passes to the left of it).



-2 if forgot (1) or (2) but pics looks OK

-1 for "unclassified" points in ovals

-1 now decision boundary.

(b) Write (a formula for) the cost function optimized by k-means. Explain your notation.

4

$$\frac{1}{2}\sum_{i=1}^{N}\left(x^{(i)}-\mu^{(z_i)}\right)^2$$

z_i = membership id of datum i $(\in 1..k)$

$x^{(i)}$ = feature values of datum i

$\mu^{(c)}$ = cluster center of cluster c.

(c) Give an advantage of hierarchical agglomerative clustering over k-means, and an advantage of k-means over hierarchical clustering. Many possible:

3 k-means:

+ often faster

3 clear objective function

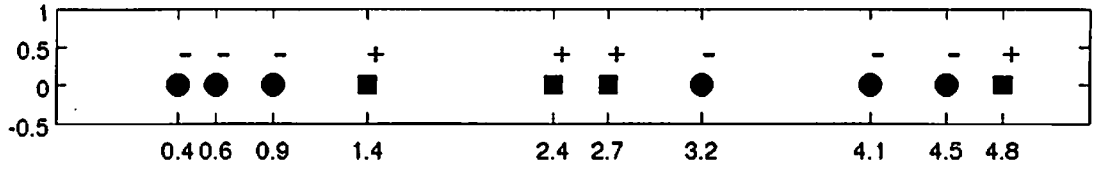
interpretable as mixture model

hierarchical:

can learn "long skinny" clusters (for some choices of costs)

produces many clusterings (no need to specify k first)

Problem 4: Classification in 1D (20 points)



Using the above 1D data to answer the following questions. Express error rates as number of data incorrectly classified.

(a) What is the best training error rate we can get on these data from a Gaussian class-conditional model classifier with equal variances? (Explain briefly, how it is achieved and why it is the best.)

5

3/10, many ways, ex: (-1) ← | → (+1)
1.0

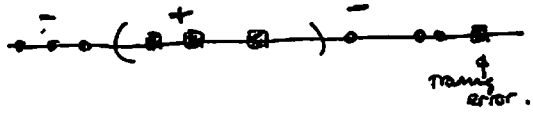
Equal covariance Gaussians will produce a linear decision boundary, eg. "-" if $x < 1.0$
"+" if $x > 1.0$

(b) What is the best training error rate we can get from such a model with unequal variances? (Again, explain briefly how it is achieved and why it is the best.)

5

Unequal variances ⇒ quadratic decision boundary

Can achieve 1/10 error:



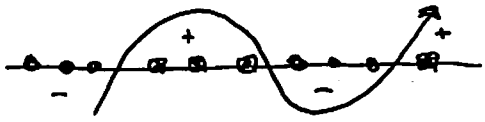
-1 wrong answer
-2 wrong explan.

(c) What is the best training error rate we can get from a perceptron classifier using polynomial features? (Again, explain briefly how it is achieved and why it is the best.)

5

We can attain 0/10 error with cubic features. Let $T(x) = \begin{cases} +1 & x > 0 \\ - & x < 0 \end{cases}$

and choose $T[(x-1)(x-3)(x-4.7)]$

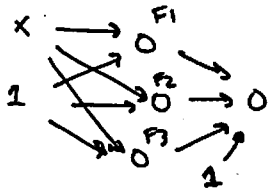


(d) What is the best training error rate we can get from a neural network classifier using features "x" and "1"? (Again, explain briefly how it is achieved and why it is the best.)

5

With at least 3 hidden features we can attain 0/10 error.

-3 right answer no explanation.



For example:

$F_1 = \sigma(100 \cdot (x-1))$

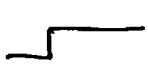
$F_2 = \sigma(100 \cdot (x-3))$

$F_3 = \sigma(100 \cdot (x-4.7))$

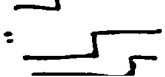
Then output

$\sigma(100 \cdot (F_1 - F_2 + F_3) - 250)$ =

→ F_1 =



F_2 =



Problem 5: Decision Trees and Entropy (20 points)

Consider learning a decision tree to predict an outcome "Y" using four features, "X1 ... X4". We observe eight training patterns, each of which we express as [x1 x2 x3 x4] (so, "0110" means, "X1=0, X2=1; X3=1; X4=0"). The data values are as follows:

Y=0 : [0110], [1010], [0011], [1111]
Y=1 : [1011], [0000], [0100], [1110]

$H(x) = \sum_i p(x=i) \log_2\left(\frac{1}{p(x=i)}\right)$

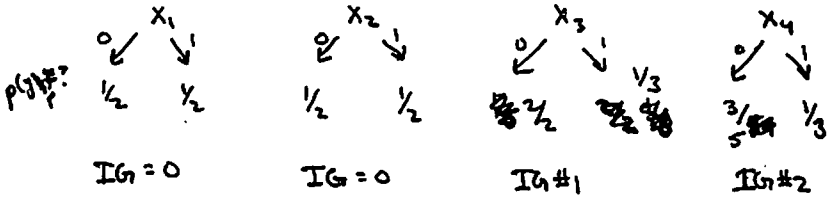
Note: you may find it useful to know the following log values:
log2(1)=0; log2(2)=1; log2(3)=1.59; log2(4)=2; log2(5)=2.32; log2(6)=2.59; log2(7)=2.81; log2(8)=3

(a) What is the entropy of Y?

$p(y=1) = 4/8 \Rightarrow \quad H(y) = \frac{1}{2}\log_2(2) + \frac{1}{2}\log_2(2) = 1/2 + 1/2 = 1$

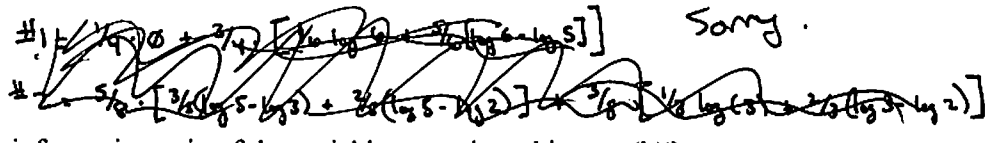
3

(b) Which variable would be split first? *Fully justify* your answer.



By inspection of the probabilities, X_3 has higher info gain than X_4

5



Sorry.

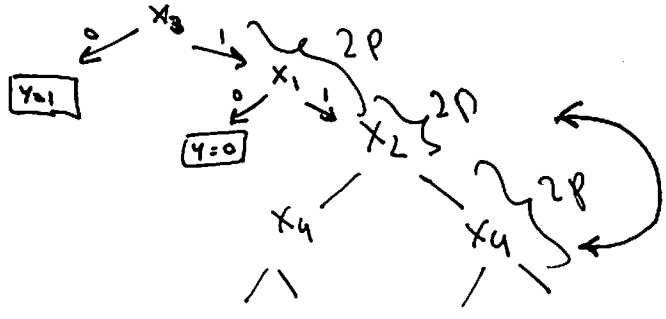
(c) What is the information gain of the variable you selected in part (b)?

$1 - \left[\frac{2}{8}(\log 1) + \frac{6}{8}\cdot\left[\frac{1}{3}\log 3 + \frac{2}{3}\log 3/2\right]\right]$

3

$= 1 - \frac{6}{8}\left[\frac{1}{3}\log 3 + \frac{2}{3}\log 3 - \frac{2}{3}\log 2\right] = 1 - \frac{6}{8}\left[\log 3 - 2/3\right]$

(d) Draw the rest of the decision tree learned on these data:



these splits can be interchanged

6

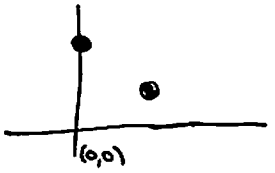
(e) What is pruning and why is it used? Give an example of why pruning is generally done *after* learning the full decision tree, rather than pre-emptively.

Important points
- pruning simplifies the decision tree

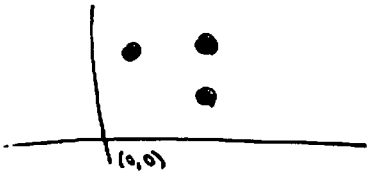
4 3

Problem 6: Shattering (10 points)

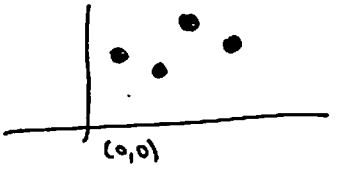
Which of the following three examples can be shattered by each learner? (There are two real-valued features, X1 and X2.) List all that can be shattered below each classifier. No explanation required.



(#1)



(#2)



(#3)

(1 pt eight diag)

Note: T[x] denotes the threshold function: +1 if x>0, -1 if x<0

3

(a) $T[a+bx_1]$ - A linear decision boundary in x_1 (x_2 unused)

⇒ #1 only

(b) $T[\frac{1}{2}((x_1-a)^2+(x_2-b)^2)+c]$ - circles centered @ (a,b) of radius $\sqrt{c}$

⇒ #1, #2

3

(c) $T[ax_1+bx_2+c]$ - a general linear decision on the plane

⇒ #1, #2

3

+1