CS178, Machine Learning
Winter 2010

Midterm Exam

Closed book, closed notes

Time: 50 min

Choose 3 of the 4 problems. If you work on all 4, clearly indicate which 3 you would like graded.

Name:

EXAM SOLUTIONS

Problem 1 ____________

Problem 2 ____________

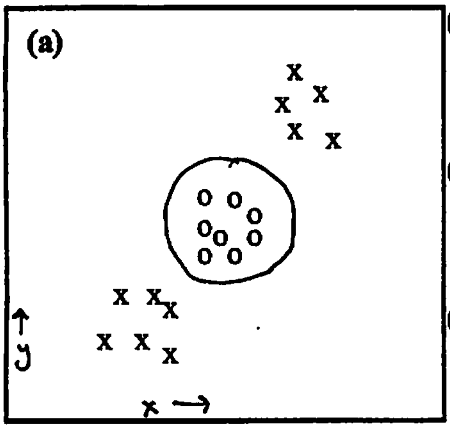
Problem 3 ____________

Problem 4 ____________

Total: ____________

Problem 1

For each of the following examples of training data, **(1)** state whether the two classes can be exactly separated (zero error on the training set) using *some* Gaussian model based classifier with equal covariance matrices (same covariance for both classes). Justify your answer with a sentence or two. **(2)** Can they be separated using two Gaussian models with unrestricted covariances? Again, justify with a sentence or two. **(3)** Write a parametric form for the decision boundary that would exactly separate the training data and the required entries of the feature matrix X, or justify why none exists.



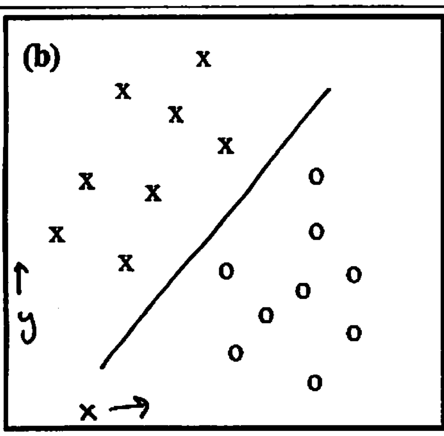
(1) No - the classes are not linearly separable

(2) Yes; a quadratic decision boundary will separate them - two circular covariances, for example

(3) The circle, eg. $(x-a)^2+(y-b)^2=c$

or $ax^2+bx+cy^2+dy+e=\phi$.

$$\underline{X} = \begin{bmatrix} 1 & x^{(i)} & (x^{(i)})^2 & y^{(i)} & (y^{(i)})^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$ will work.

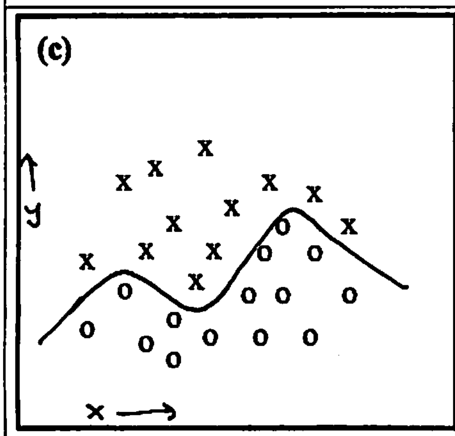


(1) Yes - these data are linearly separable

(2) Yes - this is more general than part (1)

(3) A line, $ax+by+c=0$

$$\underline{X} = \begin{bmatrix} 1 & x^{(i)} & y^{(i)} \\ \vdots & \vdots & \vdots \end{bmatrix}$$ will work.



(1) No - not linearly separable

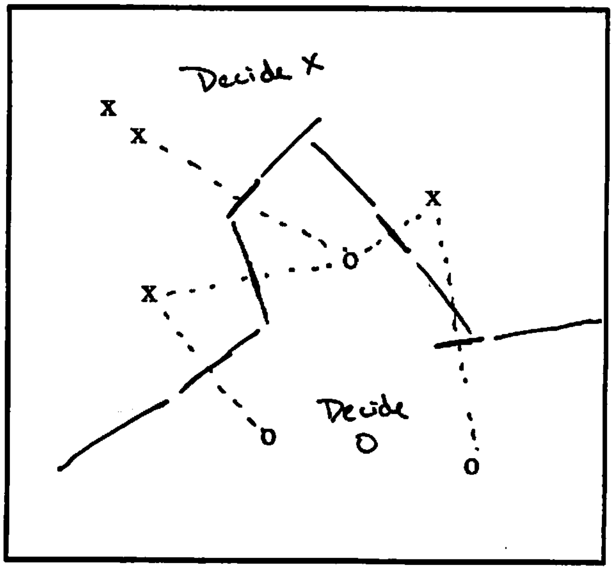
(2) No - not quadratically separable, either

(3) The decision boundary sketched is of the form $y = ax^4+bx^3+cx^2+dx+e$ (4th power poly)

$$\Rightarrow \underline{X} = \begin{bmatrix} 1 & x^{(i)} & (x^{(i)})^2 & (x^{(i)})^3 & (x^{(i)})^4 & y^{(i)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$ will work.

Problem 2

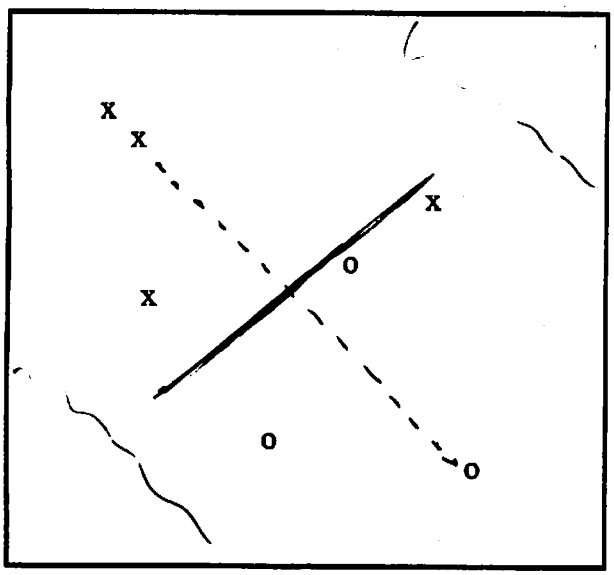
Consider the following set of training data for a k-nearest neighbors classifier.



(1) Draw the decision boundary for k=1. Show your work and justify your answer in a few sentences (2-3).

Dashed lines connect training examples
The midpoints form potential decision boundary parts
(equidistant from two examples of different classes)

Extend these lines until they intersect one another

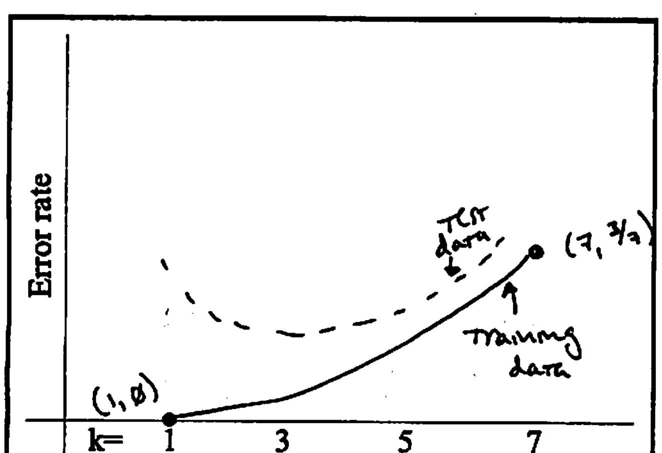


(2) Sketch the decision boundary for k=5 in the most relevant part of feature space. Again, show your work and justify your answer in a few sentences.

Near training data.

In the middle of the data, for "o" to be decided, we must be closer to all 3 "o"s (and thus, closer to the bottom right "o") than the 3rd closest "x" (one of the two upper-left x's).

Again, this is the line segment bisecting their connectors.



(3) Sketch the basic shape you would expect to see for the error rate on both training and test data, as a function of increasing k=1...7. For the training error rate, indicate the values (error rates) of the endpoints (k=1 and k=7).

On training data, with k=1 we have memorized the data ⇒ zero training error.
with k = 7, we always pick "x" ⇒ 3/7 error.

Test data we would expect a curve w/ under & over-fitting, (u-shaped), although this won't always be true in practice.

Problem 3

Suppose that we have training data $\{(x^1,y^1) \ldots (x^m, y^m)\}$, and wish to predict y using a quadratic function of x: $\hat{y}(x) = a x^2 + bx + c$

(a) Write the mean squared error cost function for our predictor

$$MSE(a,b,c) := \frac{1}{m}\sum_{i=1}^{m} \left(y^{(i)} - \hat{y}(x^{(i)})\right)^2 = \frac{1}{m}\sum_{i=1}^{m} \left(y^{(i)} - a(x^{(i)})^2 - b(x^{(i)}) - c\right)^2$$

(b) Compute its gradient with respect to a, b, and c. Interpret your equation(s).

$$\frac{\partial MSE}{\partial a} = \frac{\partial}{\partial a}\left[\frac{1}{m}\sum \left(y^{(i)} - a x^{i\,2} - bx^i - c\right)^2\right] = \frac{1}{m}\sum \frac{\partial}{\partial a}\left(y^i - a x^{i\,2} - bx^i - c\right)^2$$

$$= \frac{1}{m}\sum 2\left(y^i - a(x^i)^2 - bx^i - c\right) \cdot \frac{\partial}{\partial a}\left[y^i - a(x^i)^2 - bx^i - c\right]$$

$$= \frac{1}{m}\sum 2\underbrace{\left(y^i - a(x^i)^2 - bx^i - c\right)}_{(1)} \cdot \underbrace{(x^i)^2}_{(2)}$$

Similarly,

$$\frac{\partial MSE}{\partial b} = \frac{2}{m}\sum \left(y^i - a(x^i)^2 - bx^i - c\right) \cdot x^i$$

$$\frac{\partial MSE}{\partial c} = \frac{2}{m}\sum \left(y^i - a(x^i)^2 - bx^i - c\right) \cdot 1.$$

↳ this is the error in our prediction (under or over estimated & by how much)

(2) - this is the sensitivity of our prediction to changes in a.

(c) Write pseudocode to find the optimal values of a, b, and c using gradient descent. Be sure to specify initialization, the update itself (with sufficient detail to enable someone to write code for it), and the stopping condition (again, with sufficient detail to enable it to be coded easily).

Initialize $[a,b,c]$, for example $a := b := c := 0$. Set. stepsize $\alpha = .1$; stopping tolerance $\gamma = .0001$

Do {

For each data point i, compute $\hat{y}^i = a(x^i)^2 + bx^i + c$, our prediction.

Compute the gradient $[\partial a, \partial b\ \partial c] = \nabla$ as

$$\partial a = \frac{1}{m}\sum (y^i - \hat{y}^i) \cdot (x^i)^2$$

$$\partial b = \frac{1}{m}\sum (y^i - \hat{y}^i) \cdot (x^i)$$

$$\partial c = \frac{1}{m}\sum (y^i - \hat{y}^i) \cdot 1.$$

Take a step:

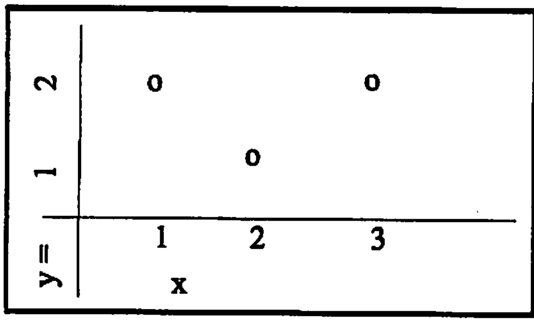
$a := a - \alpha \cdot \partial a$

$b := b - \alpha \cdot \partial b$

$c := c - \alpha\ \partial c$.

} while (~~###~~ $\|\alpha \nabla\| > \gamma$) ← this is ~~###~~ $\alpha\left(\sqrt{\partial a^2 + \partial b^2 + \partial c^2}\right) > \gamma$

Problem 4



Suppose that we observe the three data points shown (all with integer (x,y) values). We wish to use leave-one-out cross validation to decide how complex of a model we should use, comparing the constant model

(Constant) $\hat{y}(x) = b$

to a linear model

(Linear) $\hat{y}(x) = a\,x + b$

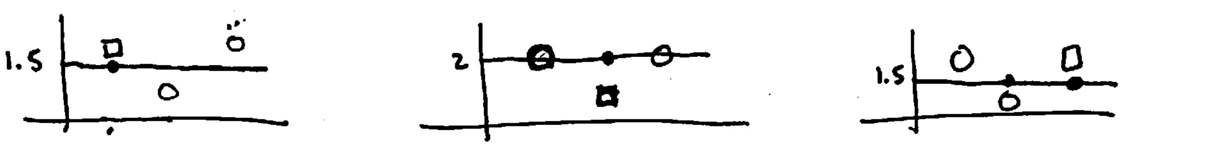
For each data point, create a training data set by leaving that point out, and a test set including only that point. Find the best predictor of each type, and compute the test accuracy. The leave-one-out cross validation accuracy is the average accuracy across each of these runs.

(a) For each cross-validation set, sketch the best predictor and compute the mean squared error. Then, compute the leave-one-out cross validation accuracy.

we'll regress 3 times & check the squared error of the left-out point.

(Constant predictor)

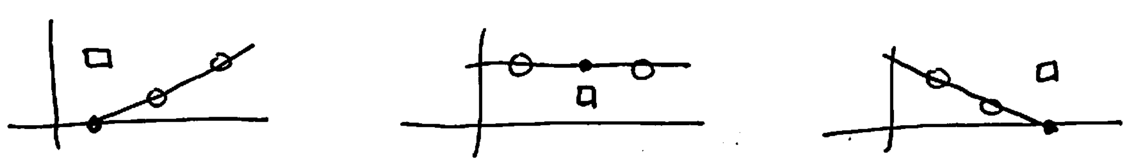
Leave out #1:



Err = .5 ⇒ MSE: $(.5)^2$ Err = 1 ⇒ MSE = $(1)^2$ Err = .5 ⇒ MSE = $(.5)^2$

MSE: $\frac{1}{3}(.25 + 1 + .25) = \frac{1}{3}(1.5) = .5$.

(Linear predictor)



Err = 2 ⇒ MSE = 4 Err = 1 ⇒ MSE = 1 Err = 2 ⇒ MSE = 4

MSE: $\frac{1}{3}(4+1+4) = \frac{1}{3}(9) = 3$.

(b) What, if anything, do their relative values suggest? Should we try to predict future data using a linear or constant predictor trained from our current (full set of) training data?

Constant predictor's MSE is much better (lower) than the linear predictor.
This suggests that the linear predictor is overfitting, ie it is too complex a model for the amount of data we have measured for training.