CS178 Final Exam

Machine Learning & Data Mining: Winter 2012

Thursday March 22nd, 2012

Your name: SOLUTIONS

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

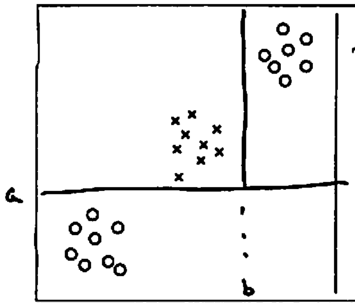
Name of the person to your right (if any):

- Total time is 1:50. 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 to come over.
- Turn in any scratch paper with your exam.

(This page intentionally left blank)

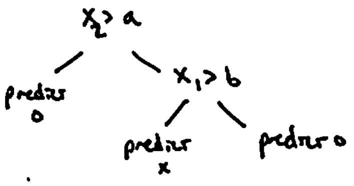
Separability & classification boundaries

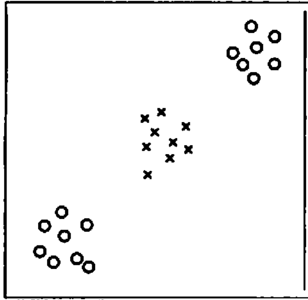
For each of the following examples of training data and classifiers, state whether there exists a set of parameters that can separate the data and justify your answer briefly (~1 sentence). If yes, also sketch the decision boundary.



Two-level decision tree:

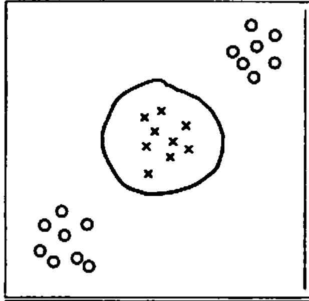
Yes - for example,





Gaussian Bayes with equal covariances

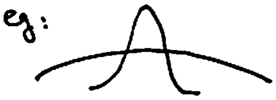
No ⇒ linear decision boundary, but not linearly separable.



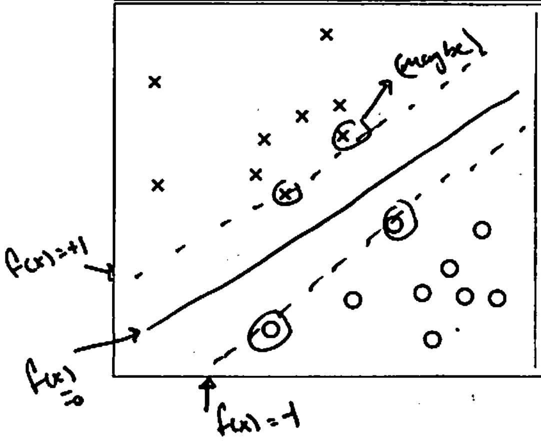
Gaussian Bayes with diagonal covariances

Yes - "o" class has larger variance than x

eg:



in one dimension.



Sketch the decision boundary that would be learned by a linear support vector machine, and identify the support vectors.

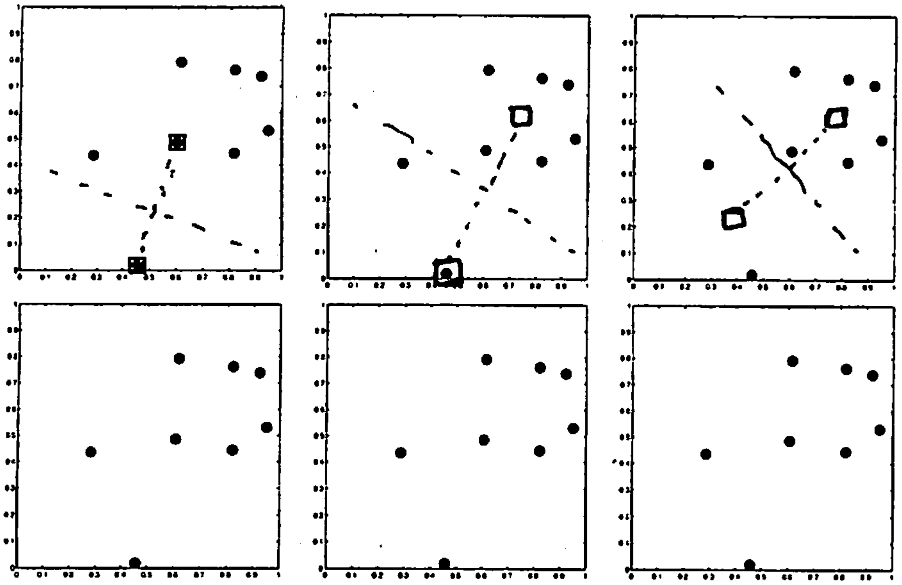
Support vectors (data on the margin) circled.

Clustering

Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data using two different algorithms, where each panel is used to show an iteration or step of the algorithm.

k-means

(a) Starting from the two cluster centers indicated by squares, perform k-means clustering on the data points. In each panel, indicate (somehow) the data assignment, and in the next panel show the new cluster centers. Stop when converged, or after 6 steps, whichever is first. It may be helpful to recall from our nearest-neighbor classifier that the set of points nearer to A than B is separated by a line.



Stop.

(b) In your own words and at a high level, explain how the expectation-maximization algorithm for Gaussian mixture models differs from the k-means algorithm.

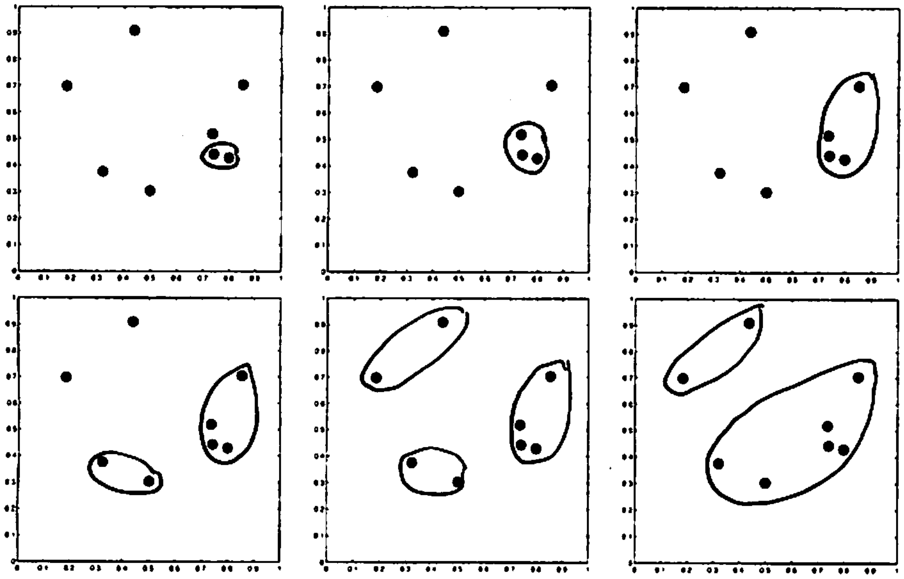
Two main differences:

(1) In GMMs, each cluster has a mean (like kmeans), but also a covariance (shape) Σ, and magnitude π

(2) EM computes "soft" membership probabilities for its updates, while k-means makes a hard decision, assigning each point to one cluster before updating the parameters.

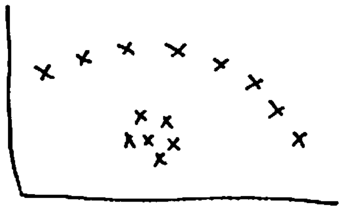
Linkage

(a) Now execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "single linkage" (minimum distance) for the cluster scores. Stop when converged, or after 6 steps, whichever is first. Show each step separately in a panel.



(judgement call this one or next one)

(b) Draw an example data set in which "single linkage" (minimum distance) would produce a more reasonable clustering than "complete linkage" (maximum distance), and explain briefly why this is the case.



Single linkage will (eventually) group the long chain & cluster into 2 groups, while maximum (complete linkage) will never find that grouping.

Complexity

Suppose that we have two features x_1, x_2. For each pair of learners, circle whether the one on the left has greater complexity, about equal complexity, or lower complexity than the one on the right.

A perceptron is [✓ more] equal less complex than a decision stump

3 bagged perceptrons is more equal [✓ less] complex than a 2-layer neural network with 3 hidden nodes

A linear support vector machine is more [✓ equal less] complex than a perceptron

depends on regularization of the SVM

Circle one answer for each:

Early stopping with increase not change [✓ decrease] the complexity of our learner.

Regularization will increase not change [✓ decrease] the complexity of our learner.

Boosting will [✓ increase] not change decrease the complexity of our learner.

Cross-validation and Linear Regression

Consider the data shown to the right. Suppose that we wish to learn a linear regression model with polynomial features of order p, for $p = 0$ (constant predictor) or $p = 1$ (linear).

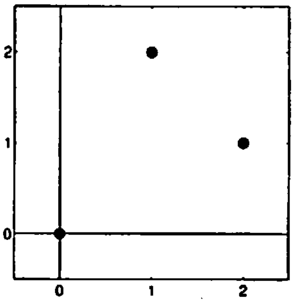
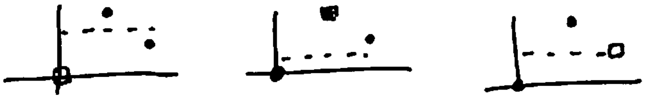


Figure 1: Data for cross-validation.

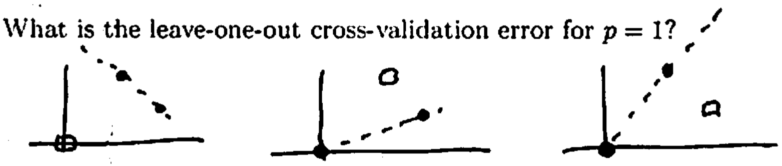
(a) What is the leave-one-out cross-validation error for $p = 0$?



$\frac{1}{3}\left[(1.5)^2 + (1.5)^2 + \emptyset^2 \right]$

$= 1.5$

(b) What is the leave-one-out cross-validation error for $p = 1$?

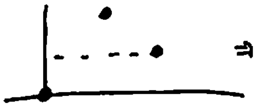


$\frac{1}{3}\left[3^2 + (1.5)^2 + 3^2 \right]$

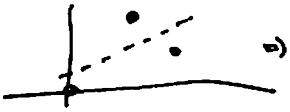
$= 6\,3/4$

(c) Are these values larger or smaller than the errors that would be found by training on all data? Sketch the leave-one-out cross-validation error along with the training error as a function of p.

Larger :

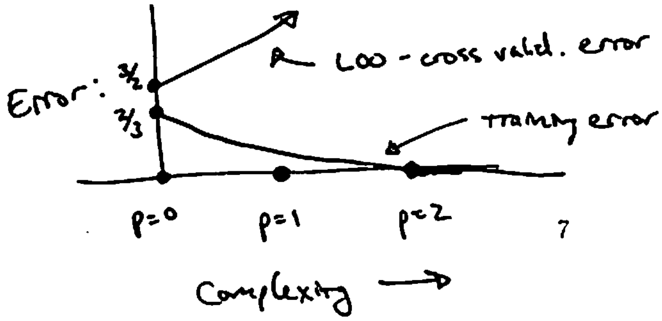


$\Rightarrow \frac{1}{3}[1^2 + 1^2 + 0^2]$ training error $= 2/3$



$\Rightarrow$ something smaller than (b)

(for $p = 2$, it is zero!)



Decision Trees

We plan to use a decision tree to predict an outcome y using four features, $x_1, \ldots, x_3$. We observe seven training patterns, each of which we represent as $[x_1, x_2, x_3]$ (so, "010" means $x_1 = 0$, $x_2 = 1$, $x_3 = 0$). We observe the training data,

$y = 0$: [000], [000], [110]

$y = 1$: [001], [010], [111], [111]

You may find the following values useful (although you may also leave logs unexpanded):

$\log_2(1) = 0$ $\log_2(2) = 1$ $\log_2(3) = 1.59$ $\log_2(4) = 2$

$\log_2(5) = 2.32$ $\log_2(6) = 2.59$ $\log_2(7) = 2.81$ $\log_2(8) = 3$

(a) What is the entropy of y?

$p(y=1) = 4/7 \quad \Rightarrow \quad H = 4/7 \log(7/4) + 3/7 \log(7/3)$

(b) Which variable would you split first? Justify your answer.

Consider the distribution of the class y under each split:

	x_1 = 0	x_1 = 1
y=0	000 000	100
y=1	001 010	111 111

	x_2 = 0	x_2 = 1
y=0	000 000	100
y=1	001	010 111 111

	x_3 = 0	x_3 = 1
y=0	000 000 110	
y=1	010	001 111 111

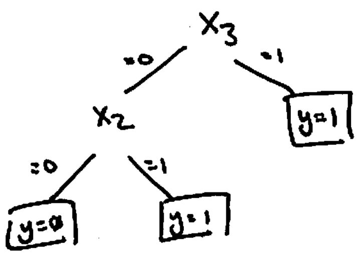
The entropy given x_3 is lowest (by inspection or by calculating: $\frac{4}{7} H(1/4) + 3/7\, H(0)$)

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

$$H(4/7) - \frac{4}{7} H(1/4) + \underbrace{3/7\; H(0)}_{\emptyset} = 4/7 \log 7/4 + 3/7 \log 7/3 - \frac{4}{7}\frac{1}{4}\log(4) - \frac{4}{7}\frac{3}{4}\log 4/3$$

~~H(4/7)~~ .

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



Latent space models and gradient descent

Suppose that we have n users, each of whom have rated a subset of d items, and would like to learn to predict the unrated items. We decide on a simple model that includes a "base" rating per user, $a^{(i)}$, for each user i; a "base" rating per item, b_j, for each item j, and a single "latent" dimension v and coefficient for each user, so that our model for the rating of user i and item j is

$$x_j^{(i)} \approx a^{(i)} + b_j + u^{(i)} v_j$$

where all quantities are scalar, real-valued numbers.

(a) Write down the mean-squared error cost function for this prediction problem.

$$C(a,b,u,v) = \frac{1}{N} \sum_{\text{rated}} \left(x_j^i - a^i - b_j - u^i v_j\right)^2$$

(b) Find the gradient of the MSE with respect to a, b, u, and v.

$$\left.\begin{aligned} \frac{\partial C}{\partial a_i} &= \frac{1}{N} \sum \left(x^i - a^i - b_j - u^i v_j\right)(-1) \\ \frac{\partial C}{\partial b_j} &= \frac{1}{N} \sum \left(x^i - a^i . b_j - u^i v_j\right)(-1) \end{aligned}\right\} \text{ Same / Symmetric}$$

$$\left.\begin{aligned} \frac{\partial C}{\partial u^i} &= \frac{1}{N} \sum \left(x^i - a^i - b_j - u^i v_j\right)(-v_j) \\ \frac{\partial C}{\partial v_j} &= \frac{1}{N} \sum \left(x^i - a^i - b_j - u^i v_j\right)(-u^i) \end{aligned}\right\} \text{ Same / symmetric}$$

(c) Give basic pseudocode for gradient descent ([✓ batch] or online) for this problem. (Specify which version you have chosen.)

```
Initialize $a^i, b_j, u^i, v_j$ for all $i,j$.
Choose step size $\alpha$
while (! done) {
    For all $i,j$, predict $\hat{x}_j^i = a^i + b_j + u^i v_j$ ;
    Calculate gradients in (b)
    $a^i \leftarrow a^i - \alpha \frac{\partial C}{\partial a^i}$ , $b_j \leftarrow b_j - \alpha \frac{\partial C}{\partial b_j}$ , etc.
    done = test gradient magnitude, or change in MSE, or change in parameters.
}.            $\|\nabla C\| < \epsilon$            $-MSE(i) + MSE(i-1) < \epsilon$
```