10-701 Midterm Exam, Spring 2006
Solutions

1. Write your name and your email address below.
   - Name:
   - Andrew account:
2. There should be 17 numbered pages in this exam (including this cover sheet).
3. You may use any and all books, papers, and notes that you brought to the exam, but not materials brought by nearby students. Calculators are allowed, but no laptops, PDAs, or Internet access.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. Note there are extra-credit sub-questions. The grade curve will be made without considering students' extra credit points. The extra credit will then be used to try to bump your grade up without affecting anyone else's grade.
7. You have 80 minutes.
8. Good luck!

Question	Topic	Max. score	Score
1	Short questions	12 + 0.52 extra	
2	Regression	12	
3	k-NN and Cross Validation	16	
4	Decision trees and pruning	20	
5	Learning theory	20 + 6 extra	
6	SVM and slacks	20 + 6 extra	

1 [12 points] Short questions

The following short questions should be answered with at most two sentences, and/or a picture. For the **(true/false)** questions, answer true or false. If you answer true, provide a short justification, if false explain why or provide a small counterexample.

1. [2 points] Discuss whether MAP estimates are less prone to overfitting than MLE.

   ★ **SOLUTION:** Usually, MAP is less prone to overfitting than MLE. MAP introduces a prior over the parameters. So, given prior knowledge, we can bias the values of the parameters. MLE on the other hand just returns the most likely parameters. Whether MAP is really less prone to overfitting depends on which prior is used – an uninformative (uniform) prior can lead to the same behavior as MLE.

2. [2 points] **true/false** Consider a classification problem with n attributes. The VC dimension of the corresponding (linear) SVM hypothesis space is larger than that of the corresponding logistic regression hypothesis space.

   ★ **SOLUTION:** False. Since they are both linear classifiers, they have same VC dimension.

3. [2 points] Consider a classification problem with two classes and n binary attributes. How many parameters would you need to learn with a Naive Bayes classifier? How many parameters would you need to learn with a Bayes optimal classifier?

   ★ **SOLUTION:** NB has $1 + 2n$ parameters — prior $P(y = T)$ and for every attribute x_i, we have $p(x_i = T|y_i = T)$ and $p(x_i = T|y_i = F)$.

   For optimal Bayes for every configuration of attributes we need to estimate $p(y|x)$. This means we have 2^n parameters.

4. [2 points] For an SVM, if we remove one of the support vectors from the training set, does the size of the maximum margin decrease, stay the same, or increase for that data set?

   ★ **SOLUTION:** The margin will either increase or stay the same, because support vectors are the ones that hold the marging from expanding.

   Here is an example of increasing margin. Suppose we have one feature $x \in \mathbb{R}$ and binary class y. The dataset consists of 3 points:

   $$(x_1, y_1) = (-1, -1), \quad (x_2, y_2) = (1, 1), \quad (x_3, y_3) = (3, 1).$$

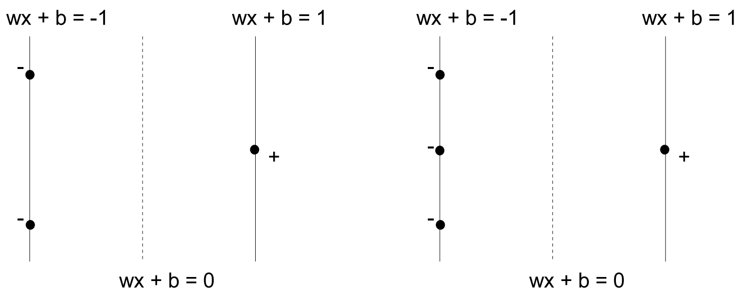


Figure 1: Example of SVM margin remaining the same after one support vector is deleted (for question 1.4).

For standard SVM with slacks the optimal separating hyperplane $wx+b=0$ has parameters

$$w = 1, b = 0$$

corresponding to the margin of $\frac{2}{|w|} = 2$. The support vectors are x_1 and x_2. If we remove (x_2, y_2) from the dataset, new optimal parameters for the separating hyperplane will be

$$w = \frac{1}{2}, b = \frac{-1}{2}$$

for the new margin of $\frac{2}{|w|} = 4$.

If there are redundant support vectors, the margin may stay the same - see Fig. 1

Only mentioning that the margin will increase was worth 1 point. Full score was given for mentioning both possibilities.

5. [2 points] **true/false** In n-fold cross-validation each data point belongs to exactly one test fold, so the test folds are independent. Are the error estimates of the separate folds also independent? So, given that the data in test folds i and j are independent, are e_i and e_j, the error estimates on test folds i and j, also independent?

★ **SOLUTION:** False. Since a data point appears in multiple folds the training sets are dependent and thus test fold error estimates are dependent.

6. [2 points] **true/false** There is an *a priori* good choice of n for n-fold cross-validation.

★ **SOLUTION:** False. We do not know the relation between sample size and the accuracy. High n increases correlation in training set and decreases variance of estimates. How much depends on the data and the learning method.

7. [0.52 extra credit points] Which of following songs are hits played by the B-52s:

★ Love Shack

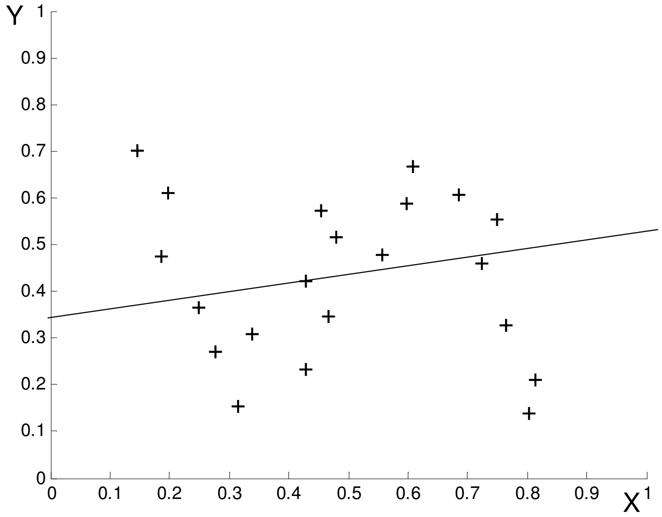
★ Private Idaho

- Symphony No. 5 in C Minor, Op. 67

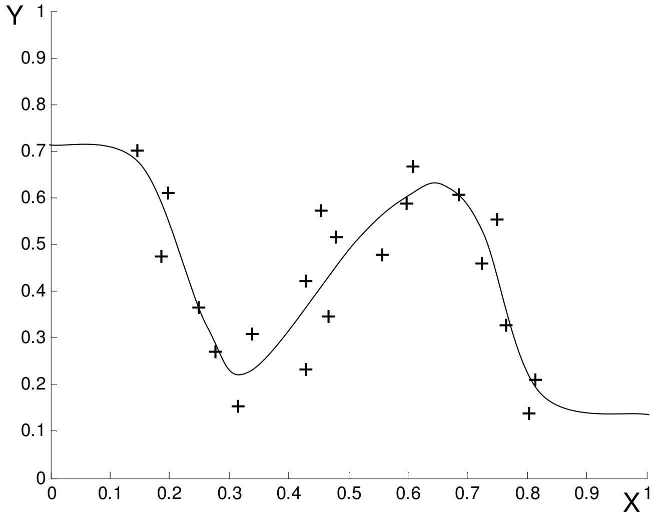
2 [12 points] Regression

For each of the following questions, you are given the same data set. Your task is to fit a smooth function to this data set using several regression techniques. Please answer all questions qualitatively, drawing the functions in the respective figures.

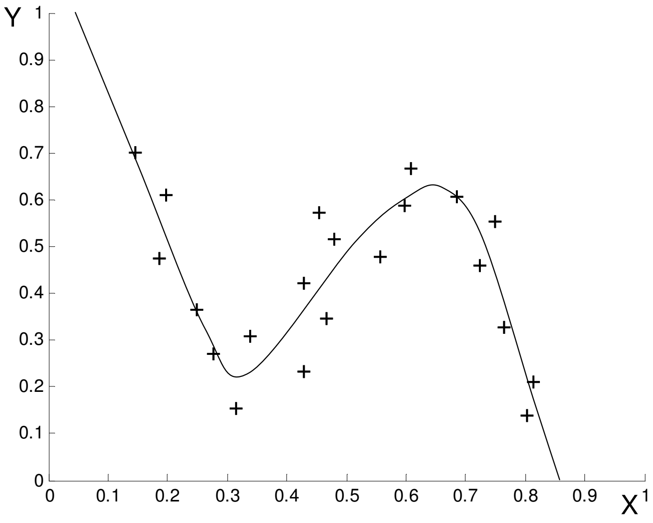
1. [3 points] Show the least squares fit of a linear regression model $Y = aX + b$.



2. [3 points] Show the fit using Kernel regression with Gaussian kernel and an appropriately chosen bandwidth.



3. [3 points] Show the fit using Kernel local linear regression for an appropriately chosen bandwidth.



4. [3 points] Suggest a linear regression model $Y = \sum_i \phi_i(X)$ which fits the data well. Why might you prefer this model to the kernel local linear regression model from part 3)?

★ **SOLUTION:** An appropriate choice would be to fit a polynomial of degree three, i.e. $Y = w_0 + w_1X + w_2X^2 + w_3X^3$. This choice seems to fit the overall trend in the data well. The advantage of this approach over kernel local linear regression is that only four parameters need to be estimated for making predictions. For kernel local linear regression, all the data has to be remembered, leading to high memory and computational requirements.

3 [16 points] k-nearest neighbor and cross-validation

In the following questions you will consider a k-nearest neighbor classifier using Euclidean distance metric on a binary classification task. We assign the class of the test point to be the class of the majority of the k nearest neighbors. Note that a point can be its own neighbor.

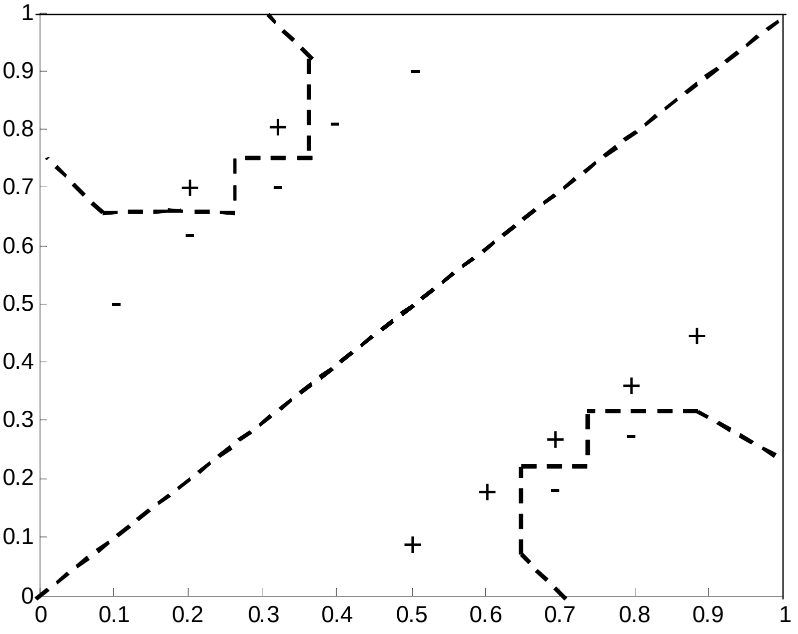


Figure 2: ★ **SOLUTION:** 1-nearest neighbor decision boundary.

1. [3 points] What value of k minimizes the training set error for this dataset? What is the resulting training error?

   ★ **SOLUTION:** Note that a point can be its own neighbor. So, $k = 0$ minimizes the training set error. The error is 0.

2. [3 points] Why might using too large values k be bad in this dataset? Why might too small values of k also be bad?

   ★ **SOLUTION:** Too big k ($k = 13$) misclassifies every datapoint (using leave one out cross validation). Too small k leads to overfitting.

3. [6 points] What value of k minimizes leave-one-out cross-validation error for this dataset? What is the resulting error?

★ **SOLUTION:** $k = 5$ or $k = 7$ minimizes the leave-one-out cross-validation error. The error is $4/14$.

4. [4 points] In Figure 2, sketch the 1-nearest neighbor decision boundary for this dataset.

   ★ **SOLUTION:** See figure 2.

4 [20] Decision trees and pruning

You get the following data set:

V	W	X	Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	0
1	1	1	0

Your task is to build a decision tree for classifying variable Y. (You can think of the data set as replicated many times, i.e. overfitting is not an issue here).

1. [6 points] Compute the information gains $IG(Y|V)$, $IG(Y|W)$ and $IG(Y|X)$. Remember, information gain is defined as

$$IG(A|B) = H(A) - \sum_{b \in B} P(B = b) H(A|B = b)$$

where

$$H(A) = -\sum_{a \in A} P(A = a) \log_2 P(A = a)$$

is the entropy of A and

$$H(A|B = b) = -\sum_{a \in A} P(A = a|B = b) \log_2 P(A = a|B = b)$$

is conditional entropy of A given B.

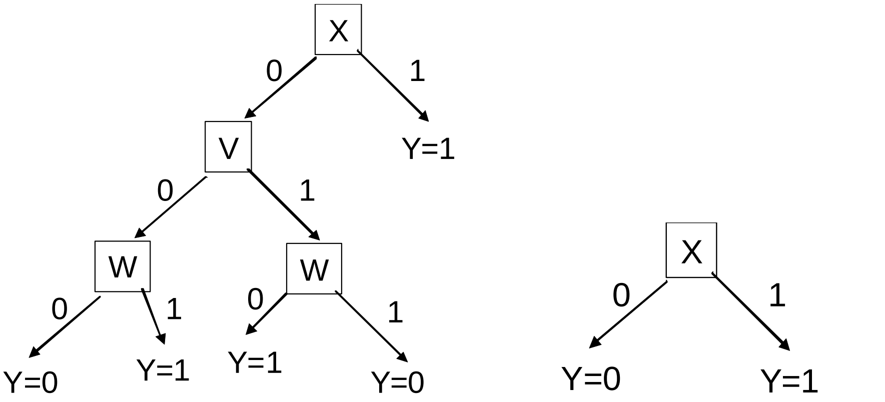
Which attribute would ID3 select first?

★ **SOLUTION:** We calculate:

$$\begin{array}{l} H(Y) = 0.97 \\ H(Y|V) = H(Y|W) = \sum_{b \in B} P(B = b) H(A|B = b) = 0.95 \\ H(Y|X) = 0.8 \end{array}$$

and information gains are: $IG(Y|V) = IG(Y|W) = 0.02$ and $IG(Y|X) = 0.17$. So attribute X is selected first.

2. [3 points] Write down the entire decision tree constructed by ID3, without pruning.



(a) Full tree (b) After top-down pruning

Figure 3: Solutions to questions 4.3, 4.4., 4.6.

★ **SOLUTION:** A full tree is constructed. First we split on X. Given a split on X the information grains for V and W are 0, so we split on either of them (let's say V). Last we split on W (information gain is 1). Figure 3(a) gives the solution.

3. [3 points] One idea for pruning would be to start at the root, and prune splits for which the information gain (or some other criterion) is less than some small ε. This is called top-down pruning. What is the decision tree returned for $\varepsilon = 0.0001$? What is the training set error for this tree?

★ **SOLUTION:** After splitting on X the information gain of both V and W is 0. So we will prune V and set the prediction to either $Y = 0$ or $Y = 1$. In either case we make 2 errors. Figure 3(b) gives the tree.

4. [3 points] Another option would be to start at the leaves, and prune subtrees for which the information gain (or some other criterion) of a split is less than some small ε. In this method, no ancestors of children with high information gain will get pruned. This is called bottom-up pruning. What is the tree returned for $\varepsilon = 0.0001$? What is the training set error for this tree?

★ **SOLUTION:** Note that $Y = V$ xor W. So when splitting on V information gain of V is $IG(Y|V) = 0$, and a step later when splitting on W, information gain of W is $IG(Y|W) = 1$.

So bottom-up pruning won't delete any nodes and the tree remains the same, figure 3(a).

5. [2 points] Discuss when you would want to choose bottom-up pruning over top-down pruning and vice versa. Compare the classification accuracy and computational complexity of both types of pruning.

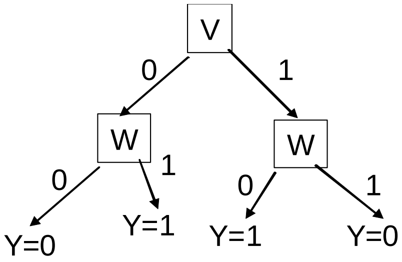


Figure 4: Optimal decision tree for our dataset.

★ **SOLUTION:** Top-down is computationally cheaper – when building the tree we can determine when to stop (no need for real pruning). But as we saw top-down pruning prunes too much. On the other hand bottom-up pruning is more expensive since we have to first build a full tree – which can be exponentially large – and then apply pruning. The other problem with this is that in the lower levels of the tree the number of examples in the subtree gets smaller so information gain might be an inappropriate criterion for pruning. One would usually use a statistical test (the p-values discussed in class) instead.

6. [3 points] What is the height of the tree returned by ID3 with bottom-up pruning? Can you find a tree with smaller height which also perfectly classifies Y on the training set? What conclusions does that imply about the performance of the ID3 algorithm?

★ **SOLUTION:** Bottom-up pruning returns tree of height 3. Smaller tree that has zero error is given in figure 4.

ID3 is a greedy algorithm. It looks only one step ahead and picks the best split. So for instance it can not find optimal tree for the XOR dataset, which is the case in our dataset.

5 [20 + 6 points] Learning theory

5.1 [8 points] Sample complexity

Consider the following hypothesis class: 3-SAT formulas over n attributes with k clauses. A 3-SAT formula is a conjunction (AND, $\wedge$) of clauses, where each clause is a disjunction (OR, $\vee$) or three attributes, the attributes may appear positively or negated ($\neg$) in a clause, and an attribute may appear in many clauses. Here is an example over 10 attributes, with 5 clauses:

$$(X_1 \vee \neg X_2 \vee X_3) \wedge (\neg X_2 \vee X_4 \vee \neg X_7) \wedge (X_3 \vee \neg X_5 \vee \neg X_9) \wedge (\neg X_7 \vee \neg X_6 \vee \neg X_{10}) \wedge (X_5 \vee X_8 \vee X_{10}).$$

You are hired as a consultant for a new company called FreeSAT.com, who wants to learn 3-SAT formulas from data. They tell you: We are trying to learn 3-SAT formulas for secret widget data, all we can tell you us that true hypothesis is a 3-SAT formula in the hypothesis class, and our top-secret learning algorithm always returns a hypothesis consistent with the input data.

Here is your job: we give you an upper bound $\epsilon > 0$ on the amount of true error we are willing to accept. We know that this machine learning stuff can be kind of flaky and the hypothesis you provide may not always be good, but it can only be bad with probability at most $\delta > 0$. We really want to know how much data we need. Please provide a bound on the amount of data required to achieve this goal. Try to make your bound as tight as possible. Justify your answer.

★ **SOLUTION:** As discussed in the lecture, $P(error_{true}(h) > \epsilon) \leq |H| e^{-m\epsilon}$. By rearranging the terms, we obtain that $|H| e^{-m\epsilon} < \delta$ if and only if

$$m > \frac{1}{\epsilon}(\ln |H| + \ln \frac{1}{\delta}). \tag{1}$$

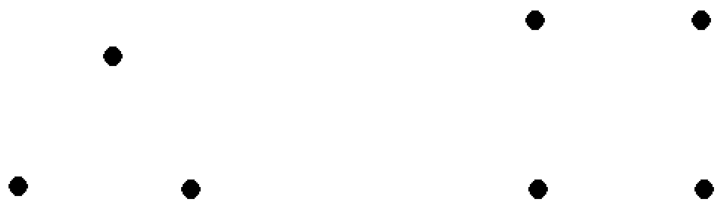
Thus, we only need to determine the number of possible hypotheses $|H|$ the learner can choose from. Each attribute may appear either positively or negated, and may only appear once in a clause. Hence, there are $2n\,2(n-1)\,2(n-2) = 8n(n-1)(n-2)$ possible clauses. Since there are a total of k clauses in each formula, $|H| = (8n(n-1)(n-2))^k$.

Plugging $|H|$ back to Equation 1, we obtain

$$m > \frac{1}{\epsilon}(k \ln 8n(n-1)(n-2) + \ln \frac{1}{\delta}). \tag{2}$$

■ **COMMON MISTAKE 1:** Some people claimed that $|H| = 2^{\binom{n}{3}}$, reasoning that there are $\binom{n}{3}$ clauses, and that each clause can either appear or not appear in the formula. However, in this question, each variable can appear negated, and there are exactly k clauses in the formula.

■ **COMMON MISTAKE 2:** A few people used the bound based on the Chernoff inequality. While this is a valid bound, it is not tight (we took off 1 point).



(a) *three points* (b) *four points*

Figure 5: Figures for Question 5.2.

5.2 [12 points] VC dimension

Consider the hypothesis class of rectangles, where everything inside the rectangle is labeled as positive: A rectangle is defined by the bottom left corner (x_1, y_1) and the top right corner (x_2, y_2), where $x_2 > x_1$ and $y_2 > y_1$. A point (x, y) is labeled as positive if and only if $x_1 \leq x \leq x_2$ and $y_1 \leq y \leq y_2$. In this question, you will determine the VC dimension of this hypothesis class.

1. [3 points] Consider the three points in Figure 5(a). Show that rectangles can shatter these three points.

   ★ **SOLUTION:** We need to verify that for any assignment of labels to the points, there is a rectangle that covers the positively labeled points (and no other). Figure 6 illustrates some possible assignments and corresponding rectangles; the remaining cases follow by symmetry.

2. [3 points] Consider the four points in Figure 5(b). Show that rectangles cannot shatter these four points.

   ★ **SOLUTION:** Assign positive labels to the opposite vertices of the square formed by the 4 points and negative labels to the other two vertices, see Figure 7(a). The rectangle can cover only two points that are vertically or horizontally aligned, but not those along a diagonal.

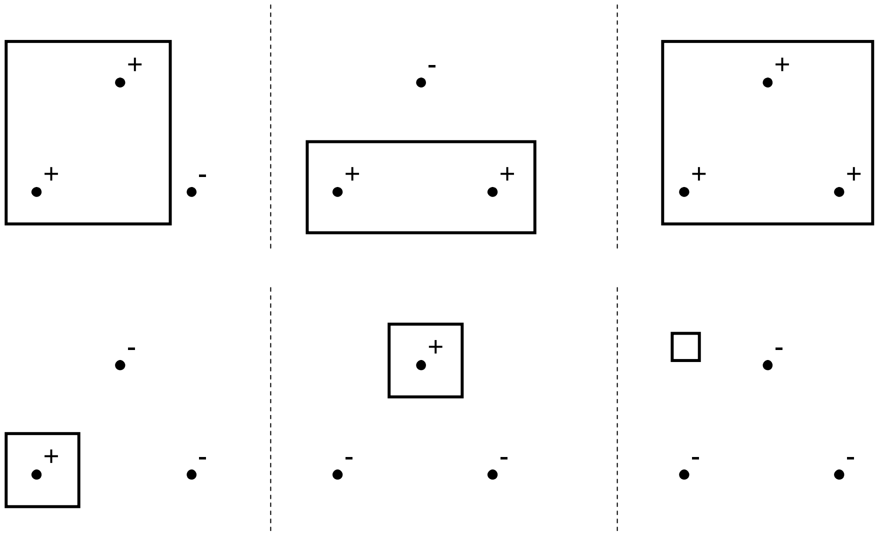
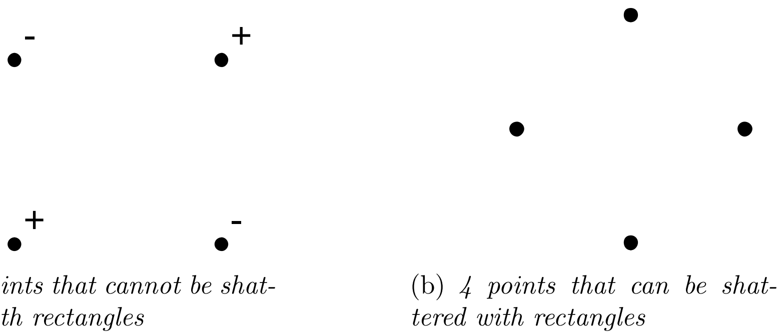


Figure 6: Solution to Question 5.2.1.



(a) *4 points that cannot be shattered with rectangles*

(b) *4 points that can be shattered with rectangles*

Figure 7: Solutions to Questions 5.2.2,5.2.3.

3. [3 points] The VC dimension of a hypothesis space is defined in terms of the largest number of input points that can be shattered, where the "hypothesis" gets to pick the locations, and an opponent gets to pick the labels. Thus, even though you showed in Item 2 that rectangles cannot shatter the four points in Figure 5(b), the VC dimension of rectangles is actually equal to 4. Prove that rectangles have VC dimension of at least 4 by showing the position of four points that can be shattered by rectangles. Justify your answer.

★ **SOLUTION:** The points in Figure 7(b) can always be correctly classified, as illustrated in Figure 8.

■ **COMMON MISTAKE :** Some people merely showed a configuration and one labeling of points that is correctly classified by rectangles. This is incorrect: instead, we need to show a configuration s.t. for *every* labeling of the points, there is a consistent hypothesis (rectangle).

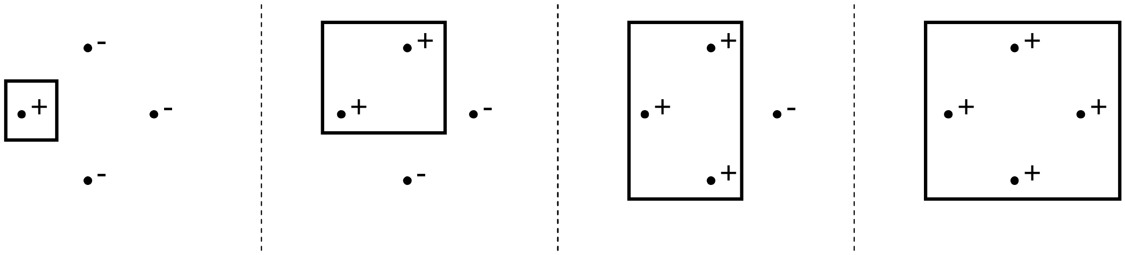
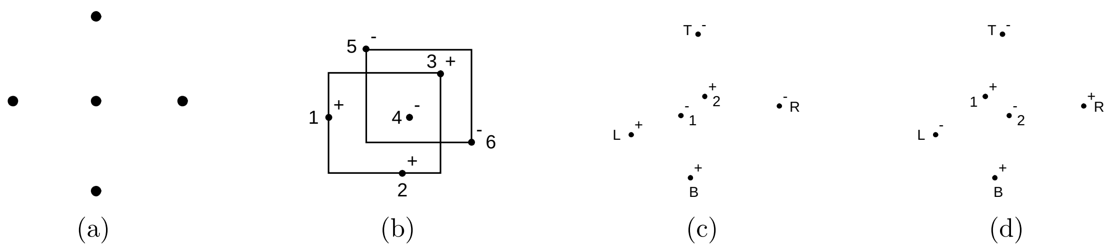


Figure 8: Solution to Question 5.2.3.

4. [3 points] So far, you have proved that the VC dimension of rectangles is at least 4. Prove that the VC dimension is exactly 4 by showing that there is no set of 5 points which can be shattered by rectangles.

★ **SOLUTION:** Given any five points in the plane, the following algorithm assigns a label to each point s.t. no rectangle can classify the points correctly: Find a point with a minimum x coordinate and a point with a maximum x coordinate and assign these points a positive label. Similarly, find a point with a minimum y coordinate and a maximum y coordinate, and assign these points a positive label. Assign the remaining point(s) a negative label. Any rectangle that classifies the positively labeled points correctly must contain all five points, hence would not correctly classify the point(s) with a negative label.

■ **COMMON MISTAKE :** Some people provided a partial proof based on the convex hull of the points, arguing that if there is a point *inside* the hull, we can assign this point a negative label and assign all the other points a positive label. While this statement is true, it still needs to be shown what the labeling ought to be when all five points lie on the boundary of the convex hull.



(a) (b) (c) (d)

Figure 9: Solution to 5.2.5. (a) 5 points that can be correctly classified with *signed* rectangles. (b)-(d) Labeling of any 6 points that cannot be correctly classified with signed rectangles.

5. **Extra credit:** [6 points] Now consider *signed* rectangles, where, in addition to defining the corners, you get to define whether everything inside the rectangle is labeled as positive or as negative. What is the VC dimension of this hypothesis class?

   Prove tight upper and lower bounds: if your answer is k, show that you can shatter k points and also show that $k+1$ points can not be shattered.

★ **SOLUTION:** We will show that $k = 5$.

To prove the lower bound, consider the configuration of points in Figure 9(a). It can be easily verified that this configuration can be classified correctly by signed rectangles for any labeling of the points, hence signed rectangles can shatter 5 points.

The proof of the upper bound is based on the following idea: Consider 6 points in an arbitrary position, as illustrated in Figure 9(b). Suppose that we are able to split these 6 points into two sets V, W, so that the minimal rectangle that covers the set V includes at least one point from rectangle W and vice versa. For example, in Figure 9(b), we can let $V = \{1, 2, 3\}$ and $W = \{4, 5, 6\}$. Then, if we label the points in V as positive and the points in W as negative, no signed rectangle can classify the points correctly.

How do we obtain such a partition? Similarly to the solution to part 4, let

- $x_L \triangleq \min_i x_i$ denote the x coordinate of a leftmost point L,
- $x_R \triangleq \max_i x_i$ denote the x coordinate of a rightmost point R,
- $y_B \triangleq \min_i y_i$ denote the y coordinate of a lowest point B, and
- $y_T \triangleq \max_i y_i$ denote the y coordinate of a topmost point T.

Take any two of the remaining points, say, 1 and 2; let 1 denote the point with the smaller x coordinate, i.e. $x_1 \leq x_2$. If $y_1 \leq y_2$, let $V = \{L, B, 2\}, W = \{R, T, 1\}$, see Figure 9(c). Otherwise, if $y_1 > y_2$, let $V = \{R, B, 1\}, W = \{L, T, 2\}$, see Figure 9(d).

6 [20 + 6 points] SVM and slacks

Consider a simple classification problem: there is one feature x with values in $\mathbb{R}$, and class y can be 1 or -1. You have 2 data points:

$$(x_1, y_1) = (1, 1)$$
$$(x_2, y_2) = (-1, -1).$$

1. [4 points] For this problem write down the QP problem for an SVM with slack variables and Hinge loss. Denote the weight for the slack variables C, and let the equation of the decision boundary be
$$wx + b = 0.$$

★ **SOLUTION:** The general problem formulation is

$$\begin{array}{rl} \min & w^T w + C \sum_i \xi_i \\ \text{subject to} & y_i(wx_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array} \tag{3}$$

In our case w is one-dimensional, so $w^T w = w^2$. Plugging in the values for x_i and y_i we get

$$\begin{array}{rl} \min & w^2 + C(\xi_1 + \xi_2) \\ \text{subject to} & 1(1w + b) \geq 1 - \xi_1 \\ & -1(-1w + b) \geq 1 - \xi_2 \\ & \xi_1 \geq 0, \quad \xi_2 \geq 0 \end{array} \quad \Rightarrow \quad \begin{array}{rl} \min & w^2 + C(\xi_1 + \xi_2) \\ \text{subject to} & w + b \geq 1 - \xi_1 \\ & w - b \geq 1 - \xi_2 \\ & \xi_1 \geq 0, \quad \xi_2 \geq 0 \end{array} \tag{4}$$

■ **COMMON MISTAKE 1:** Many people stopped after writing down a general SVM formulation (Equation 3) or its dual. We gave 2 points out of 4 for that.

2. [6 points] It turns out that optimal w is
$$w^* = \min(C, 1).$$
Find the optimal b as a function of C. *Hint: for some values of C there will be an interval of optimal b's that are equally good.*

★ **SOLUTION:** First let us write down the constraints on b from Equation 4:

$$\begin{array}{l} w + b \geq 1 - \xi_1 \\ w - b \geq 1 - \xi_2 \end{array} \quad \Rightarrow \quad 1 - \xi_1 - w \leq b \leq -1 + \xi_2 + w \tag{5}$$

We know $w^* = \min(C, 1)$, so we only need conditions on ξ_i.

$$\begin{array}{l} w^* + b \geq 1 - \xi_1 \\ w^* - b \geq 1 - \xi_2 \end{array} \quad \Rightarrow \quad \xi_1 + \xi_2 \geq 2 - 2w^*$$

We also know that $\xi_i \geq 0$ and that we are minimizing

$$w^2 + C(\xi_1 + \xi_2), \quad C > 0$$

$w = w^*$ is fixed, so we need to minimize $C(\xi_1 + \xi_2)$. Therefore

$$\xi_1 + \xi_2 = \max\{2 - 2w, 0\} = \max\{2 - 2\min\{C, 1\}, 0\}$$

Suppose $C \geq 1$. Then $w^* = 1$ and

$$\xi_1 + \xi_2 = \max\{2 - 2w^*, 0\} = \max\{0, 0\} = 0 \quad \Rightarrow \quad \xi_1 = \xi_2 = 0$$

and Equation 5 gives us

$$\begin{aligned} 1 - \xi_1 - w^* \leq b \leq -1 + \xi_2 + w^* \quad &\Rightarrow \quad 1 - w \leq b \leq -1 + w \\ &\Rightarrow \quad 0 \leq b \leq 0 \\ &\Rightarrow \quad b = 0 \end{aligned}$$

Now suppose $C < 1$. Then $w^* = C$ and

$$2 - 2w^* = 2(1 - C) > 0$$

so

$$\xi_1 + \xi_2 = \max\{2 - 2w^*, 0\} = 2 - 2C \quad \Rightarrow \quad \xi_2 = 2 - 2C - \xi_1 \tag{6}$$

and Equation 5 gives us

$$\begin{aligned} 1 - \xi_1 - w^* \leq b \leq -1 + \xi_2 + w^* \quad &\Rightarrow \quad 1 - \xi_1 - C \leq b \leq -1 + \xi_2 + C \\ &\Rightarrow \quad 1 - \xi_1 - C \leq b \leq -1 + 2 - 2C - \xi_1 + C \\ &\Rightarrow \quad 1 - \xi_1 - C \leq b \leq 1 - \xi_1 - C \\ &\Rightarrow \quad b = 1 - C - \xi_1 \end{aligned} \tag{7}$$

So, where are the intervals of equally good b's that we were promised in the beginning? Equation 6 along with

$$\xi_i \geq 0$$

give us flexibility in choosing ξ_1: all

$$\xi_1 \in [0; 2 - 2C]$$

are equally good. Therefore

$$b \in [1 - C - (2 - 2C), 1 - C] = [-1 + C, 1 - C]$$

To conclude, we have shown that optimal b is

$$b = \begin{cases} 0, & C \geq 1 \\ [-1 + C, 1 - C], & 0 < C \leq 1 \end{cases}$$

3. [4 points] Suppose that $C < 1$ and you have chosen a hyperplane

$$xw^* + b^* = 0, \text{such that } b^* = 0$$

as a solution. Now a third point, $(x_3, 1)$, is added to your dataset. Show that if $x_3 > \frac{1}{C}$, then the old parameters (w^*, b^*) achieve the same value of the objective function

$$w^2 + \sum_i C\xi_i$$

for the 3-point dataset as they did for a 2-point dataset.

★ **SOLUTION:** Because the new criterion is

$$new\_criterion = w^2 + C(\xi_1 + \xi_2 + \xi_3) = old\_criterion + C\xi_3,$$

we only need to show that the corresponding constraint

$$y_3(w^* x_3 + b^*) \geq 1 - \xi_3 \tag{8}$$

is inactive, i.e. $\xi_3 = 0$. Plugging in $w^* = C, b = 0$ we get

$$\frac{x_3}{C} \geq 1 - \xi_3$$

But

$$x_3 > \frac{1}{C} \Rightarrow \frac{x_3}{C} > 1$$

so the constraint (8) is satisfied for $\xi_3 = 0$, qed.

4. [6 points] Now in the same situation as in part 3., assume $x_3 \in [1, \frac{1}{C}]$. Show that there exists a b_3^* such that (w^*, b_3^*) achieve the same value of the objective function for the 3-point dataset as (w^*, b^*) achieve for the 2-point dataset. *Hint: Consider b_3^* such that the positive canonical hyperplane contains x_3.*

★ **SOLUTION:** We cannot anymore show that the new constraint

$$y_3(w^* x_3 + b^*) \geq 1 - \xi_3$$

is inactive given the old value of b^*, so let us use the hint and consider b_3^* such that the positive canonical hyperplane contains x_3:

$$y_3(w^* x_3 + b_3^*) = 1 \Rightarrow b_3^* = 1 - Cx_3$$

Using Equation 7 we get

$$\begin{aligned} b_3^* = 1 - C - \xi_1 \quad &\Rightarrow \quad \xi_1 = 1 - C - (1 - Cx_3) \\ &\Rightarrow \quad \xi_1 = C(x_3 - 1) \\ &\Rightarrow \quad \xi_2 = 2 - 2C - \xi_1 = 2 - C(1 + x_3) \end{aligned}$$

and we can check that $\xi_i \geq 0$ holds:

$$\begin{aligned} x_3 > 1 \quad &\Rightarrow \quad \xi_1 = C(x_3 - 1) > 0 \\ x_3 < \frac{1}{C} \quad &\Rightarrow \quad \xi_2 = 2 - C(1 + x_3) > 1 - C > 0 \end{aligned}$$

and the value of the new objective function is

$$w^{*2} + C(\xi_1 + \xi_2 + \xi_3) = C^2 + C(C(x_3 - 1) + 2 - C(1 + x_3) + 0) = C^2 + 2C(1 - C)$$

whereas the old objective function value was

$$w^{*2} + C(\xi_1 + \xi_2) = \{\text{using (6) and (7)}\} = C^2 + C(1 - C + 1 - C) = C^2 + 2C(1 - C)$$

qed.

5. **Extra credit:** [6 points] Solve the QP problem that you wrote in part 1 for the optimal w. Show that the optimal w is

$$w^* = \min(C, 1).$$

Hint: Pay attention to which constraints will be tight. It is useful to temporarily denote $\xi_1 + \xi_2$ with t. Solve the constraints for t and plug into the objective. Do a case analysis of when the constraint for t in terms of C will be tight.

★ **SOLUTION:** From Equation 4 we have

$$\begin{aligned} w + b \geq 1 - \xi_1 \\ w - b \geq 1 - \xi_2 \end{aligned} \quad \Rightarrow \quad w \geq 1 - \frac{\xi_1 + \xi_2}{2}$$

Let us show that this is in fact a tight constraint. We need to minimize w^2. If $\frac{\xi_1 + \xi_2}{2} \leq 1$, then

$$1 - \frac{\xi_1 + \xi_2}{2} \geq 0$$

so the minimal w^2 is achieved when $w = 1 - \frac{\xi_1 + \xi_2}{2}$.

Suppose now that $\frac{\xi_1 + \xi_2}{2} > 1$. Then optimal value of w is 0. The conditions on ξ_i become

$$\begin{aligned} b &\geq 1 - \xi_1 \\ -b &\geq 1 - \xi_2 \end{aligned}$$

We can then set $b = 0, \xi_i^{new} = 1$ to achieve the optimization criterion value of

$$w^2 + C(\xi_1^{new} + \xi_2^{new}) = 0 + 2C = 2C$$

instead of the old value of

$$w^2 + C(\xi_1 + \xi_2) = 0 + C(\xi_1 + \xi_2) > 2C$$

so $\frac{\xi_1+\xi_2}{2} > 1$ cannot be the optimal solution to the problem.

Now we know that $w = 1 - \frac{\xi_1+\xi_2}{2}$. Denote

$$t \equiv \xi_1 + \xi_2.$$

The optimization criterion becomes

$$\min(w^2 + C(\xi_1 + \xi_2)) = \min((1 - \frac{t}{2})^2 + Ct) \tag{9}$$

Take a derivative w.r.t. t and set it to 0:

$$\begin{aligned} 2(1 - \frac{t}{2})\frac{-1}{2} + C &= 0 \\ \frac{t}{2} &= 1 - C \end{aligned}$$

so the lowest point of (9) is achieved by

$$t = 2(1 - C)$$

Now we need the last observation: because $\xi_i \geq 0$, we have a constraint on t: $t \geq 0$. Therefore

$$t = \max\{2(1 - C), 0\}$$

so

$$w = 1 - \frac{t}{2} = 1 - \max\{1 - C, 0\} = 1 + \min\{C - 1, 0\} = \min\{C, 1\}$$

qed. This was a really hard question, especially since you did not have a lot of time on the midterm, but then it was an extra credit part.