Problem Set – Computation Theory – Ensemble Models – Reinforcement learning _KEY

1. Consider training a two-input perceptron. Give an upper bound on the number of training examples sufficient to assure with 90% confidence that the learned perceptron will have true error of at most 5%?
   VC dimension is 3

$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon))$$

   that makes

$$m \geq \frac{1}{0.05}\left(4 * \log_2\left(\frac{2}{0.1}\right) + 8 * 3 * \log_2\left(\frac{13}{0.05}\right)\right)$$

$$m \geq 20 * (4 * 4.32193 + 8 * 3 * 8.0224) = 4196.496$$

$$m \geq 4197$$

2. The VC dimension is always less than size of the hypothesis space. True/False?

   True VC dimension of a hypothesis is at most the log of the hypothesis space.

3. Consider the space of instances X corresponding to all points in the x, y plane. Give the VC dimension of the following hypothesis spaces:
   a. $H_r =$ the set of all rectangles in the x, y plane. That is $H = \{((a < x < b) \wedge (c < y < d)) \mid a, b, c, d \in \mathbb{R}\}$.

VC-dim = 4. For instance the set of points {(1, 0), (0.1), (−1, 0), (0,−1)} can be shattered, but if you draw the smallest enclosing box around 5 points it is not possible to label the point inside the box − and the remaining points on the edges +.

   b. $H_c =$ circles in the x , y plane. points inside the circle are classified as positive examples.

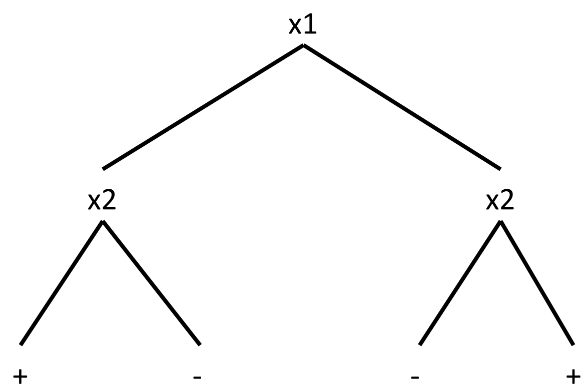
VC-dim = 3. It is easy to see the VC-dimension is at least 3 since any 3 points that make up a non-degenerate triangle can be shattered. It is a bit trickier to prove that the VC-dimension is less than 4. Given 4 points, the easy case is when one is inside the convex hull of the others. In

that case, because circles are convex, it is not possible to label the inside point − and the outside points +. Otherwise, call the points a, b, c, d in clockwise order. the claim is that it is not possible for one circle c1 to achieve labeling +,−,+,− and for some other c2 to achieve labeling −,+,−,+,−. If such c1, c2 existed, then their symmetric difference would consist of 4 disjoint regions, which is impossible for circles.

c. $H_t =$ triangles in the x, y plane. Points inside the triangle are classified as positive examples.

VC-dim = 7. Given 7 points on a circle, they can be labeled in any desired way because in any labeling, the negative examples form at most 3 contiguous blocks. Therefore one edge of the triangle can be used to cut off each block. However, no set of 8 points can be shattered. If one of the points is inside the convex hull of the rest, then it is not possible to label that point negative and the rest positive. Otherwise, it is not possible to label them in alternating +,−,+,−,+,−,+,− order.

4. (PAC Learning) Consider the hypothesis class Hrd2 of "regular, depth-2 decision trees" over n Boolean variables. A "regular, depth-2 decision tree" is a depth-2 decision tree (a tree with four leaves, all distance 2 from the root) in which the left and right child of the root are required to contain the same variable. For instance, the following tree is in Hrd2. As a function of n, how many syntactically distinct trees are there in Hrd2? By "syntactically distinct", we mean trees that look different but may still represent the same function.



A Hrd2 tree is constructed by filling blanks in the following structure. Two trees are syntactically different if they are different in any of the seven fields. In that case, the number of trees is 2^{4} * n * (n – 1)

5. Computational Learning Theory

(a) Consider the class C of concepts of the form: $(a \le x_1 \le b) \wedge (c \le x_2 \le d)$. Note that each concept in this class corresponds to a rectangle in 2-dimensions. Let a, b be integers in the range

[0, 199] and c, d be integers in the range [0, 99]. Give an upper bound on the number of training examples sufficient to assure that for any target concept c ∈ C, any consistent learner using H = C will, with probability 0.99, output a hypothesis with error at most 0.05.

Since, the learner is consistent, we will use the formula
m >= (1/ε) * (ln (1/d) + ln |H|) to get a bound on m

where, d = 0.01 and ε = 0.05

|H| is the number of rectangles = [(n1 * (n1 – 1))/2] * [(n2 * (n2 – 1))/2] where n1 is 200 and n2 is 100.

|H| = (200*199*100*99) / 4 = 98505000

Therefore, m >= (1/0.05) * (ln(1/0.01) + ln (98505000))

m>=460.216

Number of training examples sufficient to satisfy the required conditions is 461.

(b) Consider the class C of concepts of the form: $(a \leq x_1 \leq b) \wedge (c \leq x_2 \leq d) \wedge (e \leq x_3 \leq f)$. Note that each concept in this class corresponds to a hyper-rectangle in 3-d. Now suppose that a, b, c, d, e, f take on real values instead of integers. Give an upper bound on the number of training examples sufficient to assure that for any target concept c ∈ C, a learner will, with probability 0.95, output a hypothesis with error at most 0.01.

We will use the formula
m >= (1/ε) * [4 * $\log_2$(2/d) + 8*VC(H)*$\log_2$(13/ε)]

where, d = 0.05, ε = 0.01

VC(H) = 2 * Number of dimensions = 2 * 3 = 6

m>= (1/0.01) * [4 * $\log_2$(2/0.05) + 8*6*$\log_2$(13/0.01)]

m>= (1/0.01) * [4 * $\log_2$(40) + 48*$\log_2$(1300)]

m>=100*(4*5.3219 + 48*10.3443)

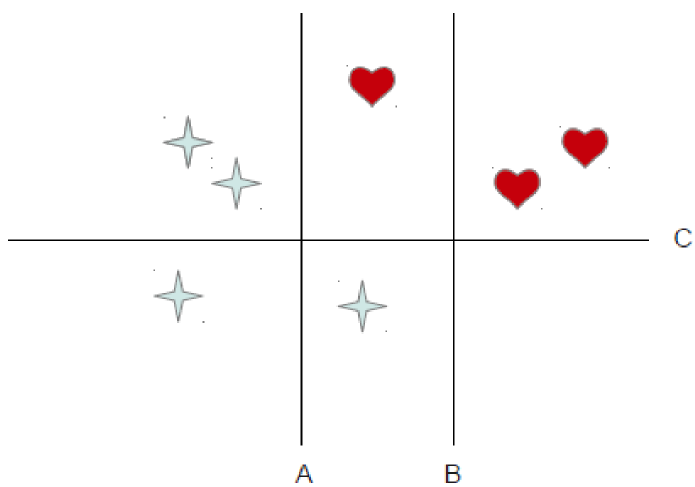
m>=51781.4

Number of training examples sufficient to satisfy the required conditions is 51782.

6. Consult the AdaBoost algorithm given in Bishop Chapter 14. Suppose there are two weak leaners h_1 and h_2, and a set of 17 points.
    a) Let say h_1 makes one mistake and h_2 makes four mistakes on the dataset. Which leaner will AdaBoost choose in the first iteration (namely m=1)?
    Will choose h_1.
    b) What is α_1?
    $\epsilon_m = 1/17$. $\alpha_m = \ln\{(1 - 1/17)/1/17\} = \ln(16)$
    c) Calculate the data weighting co-efficient w_2 for the following two cases (1) the points on which chosen leaner made a mistake and (2) the points on which the chosen leaner did not make a mistake.

Case 1: Error made
$w_2 = 1/17 \times 16 = 16/17$
Case 2: No Error
$w_2 = 1/17$.

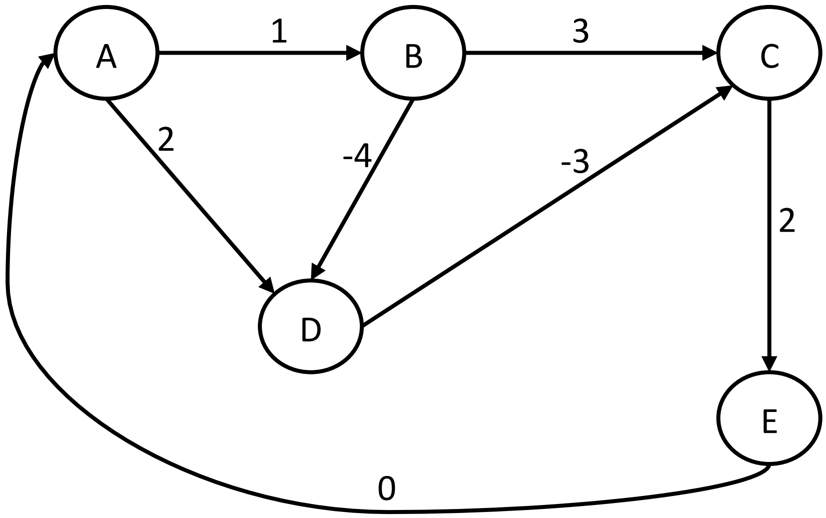
7. The diagram shows training data for a binary concept where positive examples are denoted by a heart. Also shown are three decision stumps (A, B and C) each of which consists of a linear decision boundary. Suppose that AdaBoost chooses A as the first stump in an ensemble and it has to decide between B and C as the nest stump. Which will it choose? Explain. What will be the ϵ and α values for the first iteration?



It will choose B because the only example mis-classified by A is correctly classified by B (the misclassified examples are assigned higher weight in the next iteration). B also makes another one error C makes 2 errors.
In the first iteration $\epsilon = 1/7$
and $\alpha = ln\left\{\frac{1-\epsilon}{\epsilon}\right\} = \ln(6)$

8. Reinforcement Learning

Consider a deterministic reinforcement environment drawn below (let ϒ =0.1). The numbers on the arcs indicate the immediate rewards. Assume we learn a Q-table. Also assume all the initial values in your Q table are 5.



(a) Suppose the learner follows the path A-> B -> C -> E -> A. Using one-step, standard Q learning show the calculations that produce the new Q table entries and report the final Q table of the graph.
(b) After the learning in part (a), what will be the next two states, in order, visited by the agent if it performs no exploration steps? Show the results of one-step, standard Q learning after the agent takes this path.

a) Learner follows the path A-> B -> C -> E -> A

We are using one step , standard Q learning

Gamma = 0.1

- **Initial Values of Q table are 5.**

$$Q = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

- **Applying the transition rule in the projected path we get the following in the Q table**
  i. **A -> B = 1 + 5(0.1) = 1 + 0.5 = 1.5**
  ii. **B -> C = 3 + 5(0.1) = 3 + 0.5 = 3.5**
  iii. **C -> E = 2 + 5(0.1) = 2 + 0.5 = 2.5**
  iv. **E -> A = 0 + 5(0.1) = 0 + 0.5 = 0.5**

b) The next two nodes in the order, visited by the agent if it performs no exploration steps are

- **A -> D -> C**
- **Thereby, applying the transition rule in this path will lead to the Q table**
    1. **A -> D = 2 + 5(0.1) = 2.5**
    2. **D -> C = -3 + 2.5(0.1) = -2.75**