CS 480 Fall 18 Final solutions 9

1. (i) S S A D S A B G

(ii) S D E G

(iii) S A D B G

(iv) S D A B E C G

(v) S D B C E G

(vi) yes , yes.

2. (a) After the first iteration: there are 6 points in cluster A and 2 points in cluster B.

A : all points except B : $\{(3,3), (4,3)\}$

Centers : C_A : $(1.5, 2.1)$ C_B : $(3.5, 3)$

After the second iteration:

$A^{(2)}$: points $\{(0,2), (0.5, 2.5), (1, 1.5), (1,2), (2.5, 2.5)\}$

$B^{(2)}$: $\{(3,2), (3,3), (3,4)\}$

Alternative solution:

A contains all points except $(4,3)$. $B = \{(4,3)\}$.

C_A : $\left(\frac{11}{7}, \frac{15.5}{7}\right)$ C_B : $(4,3)$

But after the second iteration, we get the same cluster $(A^{(2)}, B^{(2)})$.

(b) $$D(x,y) = \sum_{i=1}^{n} (x - x_i)^2 + \sum_{i=1}^{n} (y - y_i)^2$$

$$\frac{\partial D}{\partial x} = 2 \sum_{i=1}^{n} (x - x_i) = 0 \qquad \frac{\partial D}{\partial y} = 2 \sum_{i=1}^{n} (y - y_i) = 0$$

Solving : $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$

$$\frac{\partial^2 D}{\partial x^2} = 2n , \quad \frac{\partial^2 D}{\partial y^2} = 2n . \quad \frac{\partial^2 D}{\partial x \partial y} = 0$$

$$\left(\frac{\partial^2 D}{\partial x^2}\right)\left(\frac{\partial^2 D}{\partial y^2}\right) - \left(\frac{\partial^2 D}{\partial x \partial y}\right)^2 = 4n^2 > 0$$

Hence $D(\bar{x}, \bar{y})$ is a minimum.

3. What we need is a data structure that can support the following operations efficiently; maintain a finite set S of keys so that:

- insert (S, x) : insert key x into set S.
- delete (S, x) : delete x from set S
- nearest (S, x) : return the node with key y such that $|y - x| = \min_{z \in S} |z - x|$

Also each node has auxiliary key that is the class label.
The solution is to use an AVL tree. It is known that the operations insert, delete and nearest can be performed in $O(\log n)$ time on an AVL-tree.

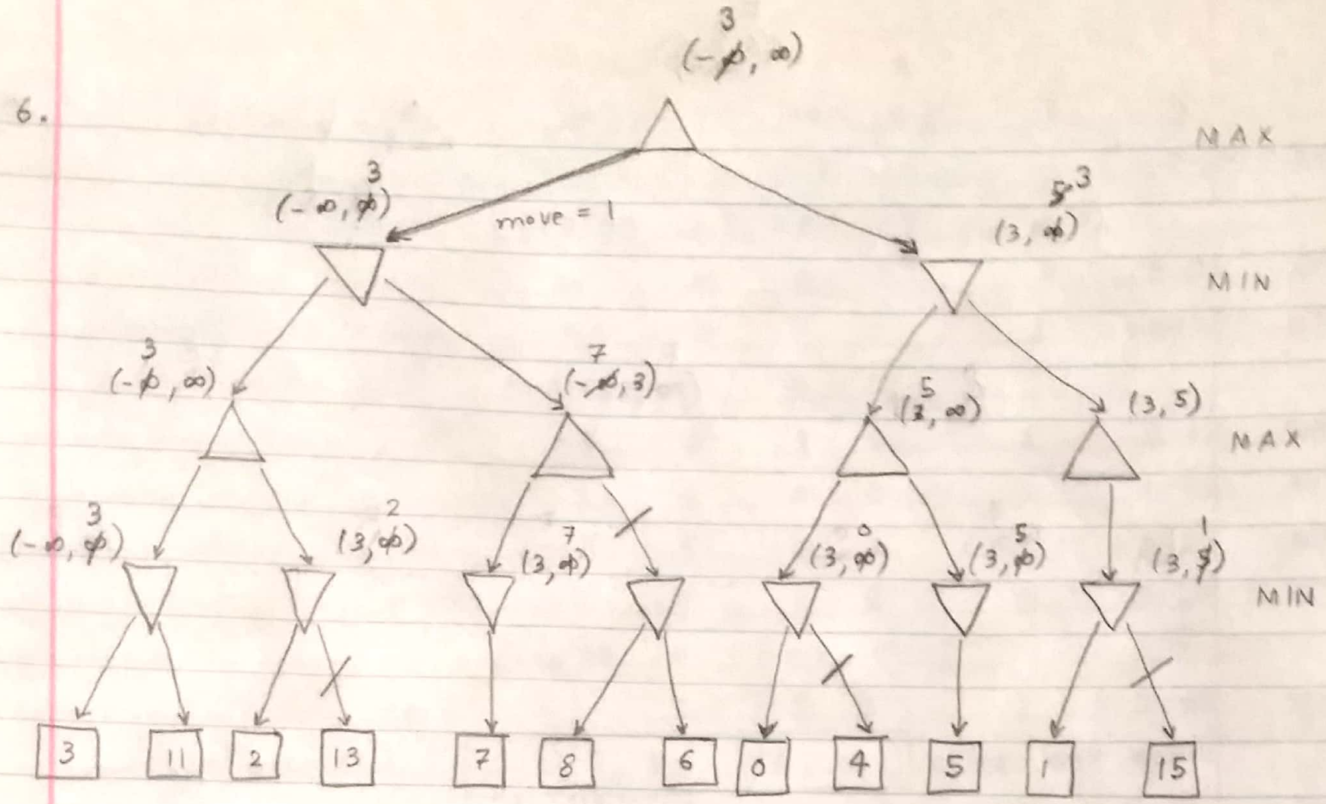
Preprocessing step: Insert all the keys of set T into a height-balanced BST such as an AVL-tree, using the class label as the auxiliary key. This is the training phase. The time complexity of this step is $O(n \log n)$.

Test phase: Given y and k, we want to determine the class label of y using k-nearest neighbor algorithm. Let T be the tree built in the training (preprocessing) step.

```
1. count = 0;   Set = { };  // set implemented as a vector
2. for j ← 1 to k do:
       y = nearest (S, x); // assume y is the node returned
       if label(y) == 1:
           count++;
       delete (S, y); set.pushback(y);
3. for each y in Set:
       insert (S, y)
4. if (count > k/2)
       return 1;
   else:
       return 0;
```

Since nearest (S, x) and insert (S, y) take time $= O(\log n)$ and each of these operations is performed k times, the total # of operations during test phase is $O(k \log n)$.

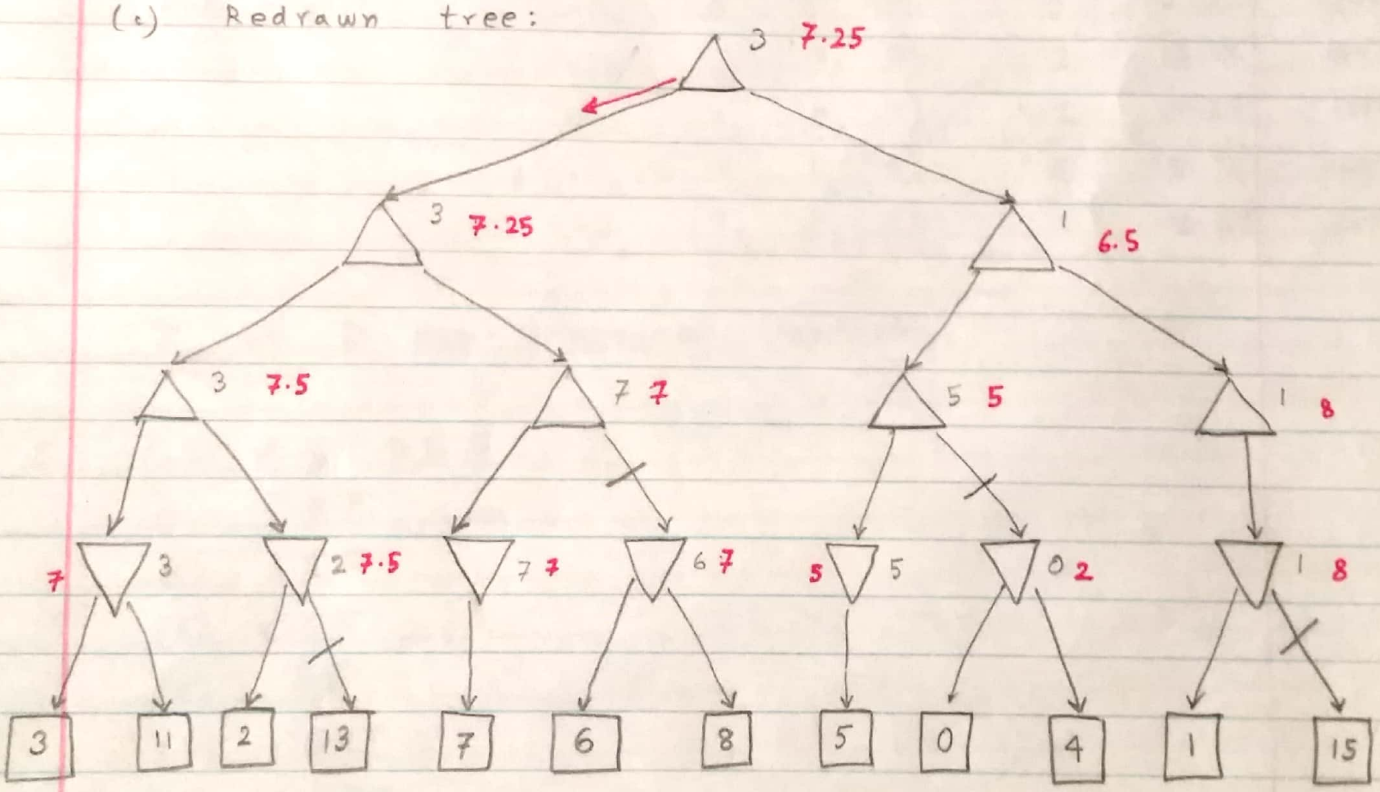
6.



(a) done!

(b) optimal move = 1

(c) Redrawn tree:



(d) Root value = 7.25, move 1 (left move) is the optimal move

No pruning.

4. Initial weight:

	x_0	x	y	z	t	y	
Initial weight:	(0.05	0.1	−0.1	0)			
	−1	0	0	0	1	−0.05	wrong!
	(−0.05,	0.1,	−0.1,	0)			
	−1	0	0	1	1	0.05	Correct!
Training with all inputs →	−1	0	1	0	1	−0.05	wrong!
	(−0.15,	0.1,	0,	0)			
	−1	0	1	1	1	0.15	correct!
	−1	1	0	0	1	0.25	correct!
	−1	1	0	1	1	0.25	correct!
$\times -0.1$	−1	1	1	0	0	0.25	wrong!
	(−0.05,	0,	−0.1,	0)			
$\times 0.1$	−1	1	1	1	1	−0.05	wrong!

Final weight: $\boxed{(-0.15,\ 0.1,\ 0,\ 0.1)}$ after one epoch.

	x_0	x	y	z	t	y	
Testing all inputs with new weights →	(−1,	0,	0,	0)	1	0.15	correct!
	(−1,	0,	0,	1)	1	0.25	correct!
	(−1,	0,	1,	0)	1	0.15	correct!
	(−1,	0,	1,	1)	1	0.25	correct!
	(−1,	1,	0,	0)	1	0.15	correct!
	(−1,	1,	0,	1)	1	0.25	correct
	(−1,	1,	1,	0)	0	0.15	wrong
	(−1,	1,	1,	1)	1	0.25	correct

7 of 8 are processed correctly.

5. (a) A B C D E F
(b) C E F
(c) B E F
(d) C E F
(e) E F
(f) C D E F
(g) B D E F
(h) C E F
(i) E F

7. (a) $p(\text{yes} \mid \text{Red, SUV, dom})$

$$\propto p(\text{Red} \mid \text{yes}) * p(\text{SUV} \mid \text{yes}) * p(\text{dom} \mid \text{yes}) * p(\text{yes})$$

$$= \frac{3}{5} \times \frac{1}{5} \times \frac{2}{5} \times \frac{1}{2} = \frac{6}{250}$$

$p(\text{No} \mid \text{Red, SUV, dom})$

$$\propto \frac{2}{5} \times \frac{3}{5} \times \frac{3}{5} \times \frac{1}{2} \propto \frac{18}{250}$$

Since $p(\text{No}) > p(\text{yes})$, the Naive Bayes classifies this instance as NO.

(b) Price of the car:

$$p(\text{price} = 12K \mid \text{yes}) = e^{-\frac{(18-12)^2}{20}} = e^{-36/20}$$

$$p(\text{price} = 12K \mid \text{no}) = e^{-\frac{(10-12)^2}{12}} = e^{-1/3}$$

Now:

$p(\text{yes} \mid \text{yellow, SUV, domestic, price} = 12K)$

$$\propto p(\text{yellow} \mid \text{yes})\, p(\text{SUV} \mid \text{yes})\, p(\text{dom} \mid \text{yes})\, p(\text{price}=12K \mid \text{yes})$$

$$= \frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} \times e^{-9/5}$$

$p(\text{No} \mid \text{yellow, SUV, domestic, price} = 12K)$

$$\propto p(\text{yellow} \mid \text{no})\, p(\text{SUV} \mid \text{no})\, p(\text{dom} \mid \text{no})\, p(\text{price}=12K \mid \text{no})$$

$$= \frac{3}{5} \times \frac{3}{5} \times \frac{3}{5} \times e^{-1/3}$$

Since $e^{-1/3} > e^{-9/5}$, and since $27 > 4$, clearly the answer is $\boxed{\text{NO}}$

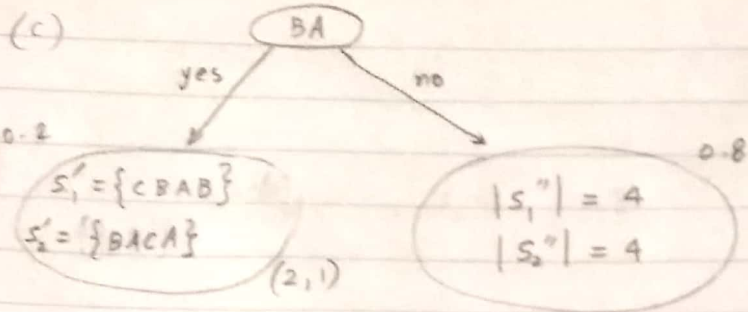
8 (a) The number of features = number of substrings of length 2 in s_1 and s_2 = 9.

\# of instances in the training set = 10.

\# of class labels = 2

(b) Entropy before any query = 1 since $p(x \in S_1) = \frac{1}{2}$ and $p(x \in S_2) = \frac{1}{2}$, so $H = -\frac{1}{2}\lg\left(\frac{1}{2}\right) - \frac{1}{2}\lg\left(\frac{1}{2}\right) = 1$.

(c)

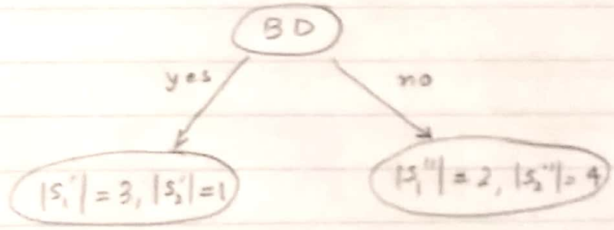


$H(\frac{1}{2}, \frac{1}{2}) = 1$ $\qquad$ $H(\frac{1}{2}, \frac{1}{2}) = 1$

$$H_{after} = 0.2\, H\left(\frac{1}{2}, \frac{1}{2}\right) + 0.8\, H\left(\frac{1}{2}, \frac{1}{2}\right)$$

$$= 1$$

info gain $= 1 - 1 = 0$



$$H\left(\frac{1}{4}, \frac{3}{4}\right) = \left(-\frac{1}{4} \lg \frac{1}{4} - \frac{3}{4} \lg \frac{3}{4}\right) \times 0.4 + 0.6 \times \Bigg(\underbrace{-\frac{1}{2} \lg \frac{1}{2}}_{1/2} - \frac{1}{4} \lg \frac{1}{4}\Bigg)$$

$$= (0.2 + 0.6 - 0.3 \lg 3) + 0.6$$

$$= 1.4 - 0.3 \lg 3$$

info gain $= 0.3 \lg 3 - 0.4 = 0.075488$

(c) Since the latter is larger, BD is a better query than BA.

(d) The two subproblems are:

yes: $S_1' = \{ADBD\}$ $\quad$ $S_2' = \{CADA\}$

no: $S_1'' = \{ABDB, CBAB, DCAB, BDCD\}$

$S_2'' = \{BDAB, AACB, DCAC, BACA\}$