CS M146 Final Exam

Jingyue Shen

TOTAL POINTS

77 / 100

QUESTION 1

1 True or False 10 / 14

- **- 0 pts** all correct
- **- 2 pts** a. incorrect
- **- 2 pts** b. incorrect
- ✓ **- 2 pts c. incorrect**
- ✓ **- 2 pts d. incorrect**
- **- 2 pts** e. incorrect
- **- 2 pts** f. incorrect
- **- 2 pts** g. incorrect

QUESTION 2

Hidden Markov Models 9 pts

2.1 a 3 / 3

- ✓ **- 0 pts Correct**
- **- 3 pts** incorrect

2.2 b 2 / 3

- **- 0 pts** Correct
- ✓ **- 1 pts partially incorrect computation**
- **- 2 pts** incorrect computation
- **- 3 pts** incorrect

2.3 c 0 / 3

- **- 0 pts** Correct
- **- 2 pts** incorrect justification
- ✓ **- 3 pts incorrect**

QUESTION 3

3 Naive Bayes 12 / 12

- ✓ **- 0 pts Correct**
- **- 1.5 pts** a) incomplete
- **- 3 pts** a) incorrect
- **- 1 pts** b) minor mistake
- **- 4 pts** b) incorrect
- **- 2 pts** c) incorrect
- **- 1 pts** c) minor mistake
- **- 3 pts** d) incorrect

QUESTION 4

Kernels and SVM 25 pts

4.1 a.i 3 / 3

- ✓ **- 0 pts Correct**
- **- 1 pts** minor error
- **- 2 pts** unclear prove, but partially correct
- **- 3 pts** Incorrect

4.2 a.ii 2 / 5

- **- 0 pts** Correct
- **- 1 pts** minor error
- **- 2 pts** Partially correct
- ✓ **- 3 pts You haven't reach the key point yet, but you are on the way**
- **- 4 pts** Wrong way!! 1 point for proving v^Av >=0 with a special v (or B)
- **- 5 pts** Incorrect

4.3 b.i 4 / 4

- ✓ **- 0 pts Correct**
- **- 2 pts** first blank incorrect
- **- 2 pts** second blank incorrect

+ infinity

4.4 b.ii 4 / 4

- ✓ **- 0 pts Correct**
- **- 2 pts** minor incorrect
- **- 4 pts** incorrect. (A typical wrong statement is saying that it turns out to be hard SVM.)

4.5 b.iii 0 / 3

- **- 0 pts** False statement, with reasonable explanation
- **- 2 pts** False statement, without/ with wrong explanation
- ✓ **- 3 pts True statement**

4.6 b.iv 3 / 3

✓ - 0 pts **True statement, mentioned dual form, support vectors or similar**

- 1 pts True statment, mentioned a perceptron-style update, but fail to discuss the difference with SVM (i.e. stating that alpha is # of mistakes)

- 2 pts True statement without/ with wrong explanation

- 3 pts False statement

4.7 b.v 3 / 3

- 1 pts W is wrong: either w1/w2 does not equal to +1 or w1,w2 are negative

- 0.5 pts b is wrong: either b is positive; or b does not suitable for W

- 1.5 pts svs are wrong (0.5 for each)

✓ - 0 pts **Correct**

QUESTION 5

Short Answer Questions 38 pts

5.1 Adaboost 1 / 3

- 0 pts Correct

- 2 pts Correct point, Incorrect justification

- 3 pts Incorrect answer

- 1 pts Wrong decision stump

✓ - 2 pts **Circled positive points on one side of the decision stump but reasoning is correct**

You can select a vertical line that learns to classify all points as positive. That will have only one incorrect classification - the negative point. Hence it has the least error.

5.2 Clustering 4 / 4

✓ - 0 pts **Correct**

- 2 pts Incorrect explanation

- 4 pts Incorrect

- 1 pts Insufficient explanation

5.3 LOOCV 0 / 3

- 0 pts Correct

✓ - 3 pts **Incorrect**

5.4 Probability 4 / 4

✓ - 0 pts **Correct**

- 1 pts Wrong denominator

- 1 pts Wrong numerator

- 4 pts Incorrect

5.5 Multiclass 6 / 6

✓ - 0 pts **Correct**

- 2 pts Minor mistake / Didn't sum over all examples / Didn't sum over all classes

- 4 pts Only procedure / Attempt to derive(taken log somewhere in the derivation)

- 6 pts Incorrect

- 3 pts Mostly correct formulation

- 1 pts Tiny mistake

5.6 PAC_i 3 / 3

✓ - 0 pts **Correct (200 examples)**

- 3 pts Incorrect

- 2 pts Correct approach but no answer

- 1 pts minor mistake

5.7 PAC_ii 3 / 3

✓ - 0 pts **Correct (PAC theorem only shows the upper bound)**

- 1 pts Incorrect explanation

- 3 pts Incorrect

5.8 Generative vs Discriminative 4 / 4

✓ - 0 pts **Both correct**

- 2 pts One incorrect answer

- 4 pts Both incorrect

5.9 VC Dimension 4 / 8

- 0 pts Correct

- 8 pts Incorrect

✓ - 4 pts **VC(DT3)=8 w/ explanation or examples**

- 2 pts incorrect Prove VC(DT3) >= 8

- 2 pts Prove VC(DT_k) < 9 (=2^3 + 1)

- 2 pts minor mistake

QUESTION 6

6 Name and Id 2 / 2

✓ - 0 pts **Correct**

gradescope

CM146: Introduction to Machine Learning	Winter 2018

Final Exam

Mar. 22^{nd}, 2018

- This is a closed book exam. Everything you need in order to solve the problems is supplied in the body of this exam.
- This exam booklet contains **five** problems.
- You have 150 minutes to earn a total of 100 points.
- **Besides giving the correct answer, being concise and clear is very important. To get the full credit, you must show your work and explain your answers.**

Good Luck!

Name and ID: (2 Point) Jingyue Shen 704797256

Name		/2
True/False Questions		/14
Hidden Markov Models		/9
Naive Bayes		/12
Kernels and SVM		/25
Short Answer Questions		/38
Total		/100

1 True or False [14 pts]

Choose either True or False for each of the following statements. For the statement you believe it is *False*, please give your brief explanation of it. Two points for each question. *Note: the credit can **only** be granted if your explanation for the false statement is correct. Also note, a negated statement is not counted as a correct explanation.*

(a) Training a k-class classification model using one-against-all is always faster than using one-vs-one because one-vs-one requires to train more binary classifiers.

False. Suppose each class has m examples, train a classifier on m examples requires time p. Then for one-against-all it runs in $O(k^2 p)$, since it requires all examples to train a classifier. for one-vs-one, it runs in $O(\binom{k}{2} \times 2p) = O(k(k-1)p)$. if k is large, then one vs one actually runs faster than one vs all

(b) We would expect the support vectors to remain the same in general as we move from a linear kernel to higher order polynomial kernels.

False. For example, if data is not linearly separable, then in linear kernel, support vectors are those within the boundaries and misclassified, but when move to higher order kernel, the data can be mapped to be linearly separable, those formerly are support vectors may be correctly classified in the higer dimension

(c) In a mistake-driven algorithm such as the Perception algorithm, if we make a mistake on example x_i with label y_i, we update the weights w so that we can guarantee that we now predict y_i correctly.

True.

(d) Consider a classification problem with n features. The VC dimension of the corresponding (linear) SVM hypothesis space is larger than that of the corresponding logistic regression hypothesis space.

False. the hypothesis space of logistic regression is all sigmoid funtion, $\frac{1}{1+e^{-z}}$, which has larger VC dimension

(e) A 3-layer neural network with non-linear activation functions can learn non-linear decision boundaries.

True.

(f) In AdaBoost, the weight associated with each weak learner can be negative (less than 0).

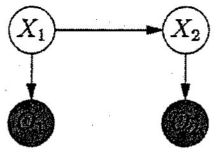
False. all weights should be greater than 0

(g) Using MAP to estimate model parameters always give us better performance.

False. We may falsely assume some prior, which can lead to worse performance

2 Hidden Markov Models [9 pts]

Consider the following Hidden Markov Model.



X_1	$\Pr(X_1)$
0	0.3
1	0.7

X_t	X_{t+1}	$\Pr(X_{t+1}\|X_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

X_t	O_t	$\Pr(O_t\|X_t)$
0	A	0.9
0	B	0.1
1	A	0.5
1	B	0.5

Suppose that $O_1 = A$ and $O_2 = B$ is observed.

(a) **(3 pts)** What is the probability of $P(O_1 = A, O_2 = B, X_1 = 0, X_2 = 1)$? (don't need to calculate number)

$P(O_1=A, O_2=B, X_1=0, X_2=1) = P(O_2=B|X_2=1)\,P(X_2=1|X_1=0)$
$P(O_1=A|X_1=0)\,P(X_1=0)$

$= 0.3 \times 0.9 \times 0.6 \times 0.5$

(b) **(3 pts)** What is the most likely assignment for X_1 and X_2? given $O_1=A$, $O_2=B$

$X_1=0$ $X_2=0$ $0.3\times0.4\times0.9\times0.1$

$$P(X_1, X_2 | O_1=A, O_2=B) = \frac{P(O_1=A, O_2=B | X_1, X_2)\,P(X_2|X_1)\,P(X_1)}{P(O_1=A, O_2=B)}$$

∴ most likely is. $X_1=0$ $X_2=1$

(c) **(3 pts)** [True/False] Based on the independent assumptions in HMM, the random variable O_1 is independent of the random variable X_2. Justify your answer.

True. The value of O_1 is only decided by the emission probability of state X_1

the value of X_2 does not affect state X_1's emission probability $P(O_1|X_1)$

3 Naive Bayes [12 pts]

Data the android is about to play in a concert on the Enterprise and he wants to use a Naive Bayes classifier to predict whether he will impress Captain Picard. He believes that the outcome depends on whether Picard has been reading Shakespeare or not for the three days before the concert. For the previous five concerts, Data has observed Picard and noted on which days he read Shakespeare. His observations look like this:

D1 (Day 1)	D2 (Day 2)	D3 (Day 3)	LC (LikedConcert)
1	1	0	yes
0	0	1	no
1	1	1	yes
1	0	1	no
0	0	0	no

(a) **(3 pts)** What does the modeling assumption make in the Naive Bayes model?

The reading behavior of each day is independent of reading behavior of other days

(b) **(4 pts)** Show the Naive Bayes model that Data obtains using maximum likelihood from these instances. (Write down the numerical values of the model parameters.)

$P(D_1 = True) = \frac{3}{5}$

$P(D_2 = True) = \frac{2}{5}$

$P(D_3 = True) = \frac{3}{5}$

$P(LC = True) = \frac{2}{5}$

$P(D_1 = True \mid LC = True) = 1$

$P(D_2 = True \mid LC = True) = 1$

$P(D_3 = True \mid LC = True) = \frac{1}{2}$

$P(D_1 = True \mid LC = False) = \frac{1}{3}$

$P(D_2 = True \mid LC = False) = 0$

$P(D_3 = True \mid LC = False) = \frac{2}{3}$

(c) **(2 pts)** If Picard reads Shakespeare only on day 1 and day 2, how likely is he to enjoy Data's concert?

$$P(LC = yes \mid D_1 = 1, D_2 = 1, D_3 = 0) = \frac{P(D_1=1 \mid LC=yes)\, P(D_2=1 \mid LC=yes)\, P(D_3=0 \mid LC=yes)\, P(LC=yes)}{P(D_1=1 \mid LC=yes)\, P(D_2=1 \mid LC=yes)\, P(D_3=0 \mid LC=yes)\, P(LC=yes) + P(D_1=1 \mid LC=No)\, P(D_2=1 \mid LC=No)\, P(D_3=0 \mid LC=No)\, P(LC=No)}$$

$$= \frac{1 \times 1 \times \frac{1}{2} \times \frac{2}{5}}{1 \times 1 \times \frac{1}{2} \times \frac{2}{5} + \frac{1}{3} \times 0 \times \frac{1}{3} \times \frac{3}{5}}$$

$$= 1$$

(d) **(3 pts)** Estimate $P(LC = yes \mid D_2 = 1)$.

$$P(LC = yes \mid D_2 = 1) = \frac{P(D_2 = 1 \mid LC = yes)\, P(LC = yes)}{P(D_2 = 1)} = \frac{1 \times \frac{2}{5}}{\frac{2}{5}} = 1$$

$\frac{3}{5} \times \frac{3}{5}$

$\frac{3}{5} \times \frac{2}{5}$

$\frac{2}{5} \times \frac{3}{5}$

$\frac{2}{5} \times \frac{2}{5}$

4 Kernels and SVM [25 pts]

(a) **(8 pts)** Properties of Kernels

i. **(3 pts)** Given n training examples $\{x_i\}_{i=1}^n$, the kernel matrix $\mathbf{A}$ is an $n \times n$ square matrix, where $\mathbf{A}(i,j) = K(x_i, x_j) = \Phi(x_i)^T\Phi(x_j)$. Prove that the kernel matrix is symmetric (i.e, $A_{i,j} = A_{j,i}$).
hints: Your proof will not be longer than 2 or 3 lines.

$A_{i,j} = k(x_i, x_j) = \phi(x_i)^T \phi(x_j) = \sum_k x_{ik} x_{jk}$.

$A_{j,i} = k(x_j, x_i) = \phi(x_j)^T \phi(x_i) = \sum_k x_{jk} x_{ik}$.

$\therefore A_{i,j} = A_{j,i}$

$x_{ik}\ x_{1k}$.

ii. **(5 pts)** Prove that the kernel matrix $\mathbf{A}$ is positive semi-definite.
hints: (1) Remember that an $n \times n$ matrix $\mathbf{A}$ is positive semi-definite if and only if for any n dimensional vector $\mathbf{v} \neq \mathbf{0}$, we have $\mathbf{v}^T\mathbf{A}\mathbf{v} \geq 0$. (2) Consider a matrix $\mathbf{B} = [\Phi(x_1), \cdots, \Phi(x_n)]$ and use it to prove A is positive semi-definite.

(1) let $\vec{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$ then $\vec{v}^T A = (\sum_i v_i A_{i1}\ \ \sum_i v_i A_{i2} \cdots \sum_i v_i A_{in})$

$\Rightarrow \vec{v}^T A \vec{v} = (v_1 \sum_i v_i A_{i1} + v_2 \sum_i v_i A_{i2} + \cdots + v_n \sum_i v_i A_{in})$

$=$

$v_i 1 \quad v_2 \sum v_i \quad v_3$

$v_1\ v_1 A_{11}$
$v_1\ v_2 A_{21}$
$v_1\ v_3 A_{31}$
$\vdots$
$v_1\ v_n A_{n1}$

$v_2\ v_1 A_{12}$
$v_3\ v_1 A_{13}$
$\vdots$
$v_n\ v_1 A_{1n}$.

$v_1 v_2 A_{12}$
$2 v_1 v_3 A_{13}$

(b) **(17 pts)** Given a dataset $D = \{x_i, y_i\}, x_i \in \mathbb{R}^k, y_i = \{-1, +1\}, 1 \leq i \leq N$.
A hard SVM solves the following formulation

$$\min_{w,b} \frac{1}{2}w^Tw \quad \text{s.t } \forall i, y_i(w^Tx_i + b) \geq 1, \tag{1}$$

and soft SVM solves

$$\min_{w,\xi_i,b} \frac{1}{2}w^Tw + C\sum_i \xi_i \quad \text{s.t } \forall i, y_i(w^Tx_i + b) \geq 1 - \xi_i, \quad \forall i, \xi_i \geq 0 \tag{2}$$

i. **(4 pts)** Complete:
If $C =$ 1000000, soft SVM will behave exactly as hard SVM.
In order to reduce over-fitting, one should decrease (decrease or increase) the value of C.

ii. **(4 pts)** Show that when $C = 0$, the soft SVM returns a trivial solution and cannot be a good classification model.

When $C=0$, we do not include the slack variable in the objective function, we can set $W=0$ and ε_i to very large ~~ve~~ value to make $\frac{1}{2}W^TW = 0$. In this case the classifier returned does not genalize any misclassification

iii. **(3 pts)** [True/False] The slack variable ξ_i in soft SVM for a data point x_i always takes the value 0 if the data point is correctly classified by the hyper-plane. Explain your answer.

True. ε_i is used to add penalization to those points within boundary and misclassified, so for correctly classified points, we do not need to add cost ε_i in objective function

iv. **(3 pts)** [True/False] The optimal weight vector w can be calculated as a linear combination of the training data points. Explain your answer. [You do not to prove this.]

True. In the dual form of SVM, $W = \sum \alpha_i y_i x_i$. ~~we~~ α_i is the weight of each data points, only those points within boundary and misclassified has weight greater than zero.

Since we update W using $y_i x_i$ when making mistakes or fall within boundaries

hard SVM

v. **(3 pts)** We are given the dataset in Figure 1 below, where the positive examples are represented as black circles and negative points as white squares. (The same data is also provided in Table 1 for your convenience). Recall that the equation of the separating hyperplane is $\hat{y} = \mathbf{w}^T\mathbf{x} + b$.

i. Write down the parameters for the learned linear decision function.

$W = (w_1, w_2) =$ ___(1, 1)___. $b =$ ___-1___

ii. Circle all support vectors in Figure 1.

$x_2 = -x_1 + 1$

$y: x_1 + x_2 - 1 = 0$

index	x_1	x_2	y
1	0	0	−
2	0	-4	−
3	-1	-1	−
4	-2	-2	−
5	3	0	+
6	0	3	+
7	1	1	+
8	3	-1	+

Table 1: The dataset S

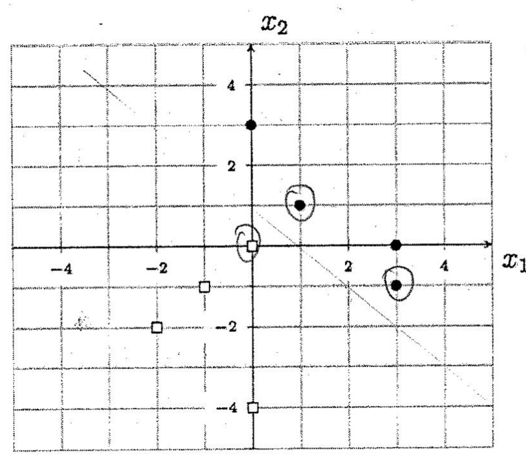


Figure 1: Linear SVM

5 Short Answer Questions [38 pts]

Most of the following questions can be answered in one or two sentences. Please make your answer concise and to the point.

(a) **(3 pts)** Consider training a classifier using AdaBoost with decision stumps (pick a horizontal or a vertical line, and one side of the half-space is positive and the other one is negative) on the following dataset:

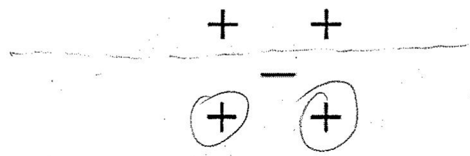
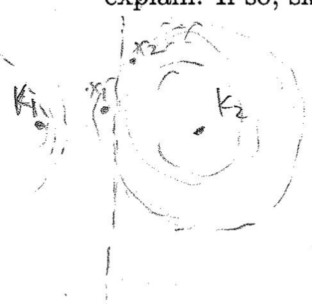


Figure 2: Example 2D dataset for Boosting

Which example(s) will have their weights increased at the end of the first iteration? Circle them and justify.

To best classify this examples, we can draw a horizontal line so that examples above line are positive and below are negative. Then the two circled are misclassified. So adaboost will increase these two misclassified sample's weight.

(b) **(4 pts)** Suppose we clustered a set of N data points using two different clustering algorithms: k-means and Gaussian mixtures. In both cases we obtained 2 clusters and both algorithms return the same set of cluster centers. Can 2 points that are assigned to different clusters in the kmeans solution be assigned to the same cluster in the Gaussian mixture solution? If no, explain. If so, sketch an example and explain in 1-2 sentences.



Yes. In k-means, given the example, x_1 will be classified to k_1 and x_2 to k_2 since x_1 is close to k_1, x_2 is close to k_2. But in GMM, $P(k_2|x_1)$ can be greater than $P(k_1|x_1)$ since it might be the case that the Gaussian distribution of k_1 is very steep and thus probability drops very fast.

(c) **(3 pts)** Suppose you are running a learning experiment on a new algorithm for binary classification. You have a data set consisting of 100 positive and 100 negative examples. You plan to use leave-one-out cross-validation (i.e., 200-fold cross-validation) to evaluate a baseline method: a simple majority function (i.e., returns the most frequent label on the training set as the prediction). What is the average cross-validation accuracy of the baseline? (Only need to write down the number).

$\frac{100}{199}$

$P(B|A) = 0.01$

A. T.C.

B. good / bad

(d) **(4 pts)** P(Good Movie | Includes Tom Cruise) = 0.01
P(Good Movie | Tom Cruise absent) = 0.1
P(Tom Cruise in a randomly chosen movie) = 0.01
What is P(Tom Cruise is in the movie | Not a Good Movie)?

$$P(\text{T.C. in movie} \mid \text{Not good}) = \frac{P(\text{not good} \mid \text{T.C. in movie})\, P(\text{T.C. in movie})}{P(\text{not good})}$$

$$= \frac{(1-0.01)\times 0.01}{0.9\times 0.99 + 0.99\times 0.01}$$

$$= \frac{99\times 0.01}{0.9+0.01}$$

$$= \frac{0.01}{0.91} = \frac{1}{91}$$

(e) **(6 pts)** We can easily extend the binary Logistic Regression model to handle multi-class classification. Lets assume we have K different classes, and posterior probability for class k is given as

$$P(y=k|X=x) = \frac{\exp(w_k^T x)}{\sum_{k'=1}^{K-1} \exp(w_{k'}^T x)} \tag{3}$$

where x is a d dimensional vector and w_k is the weight matrix for the k^{th} class.
Assuming dataset D consists of n examples, derive the log likelihood condition for this classifier.

hints: Let I_{ik} be an indicator function, where $i = 1,\ldots,n$ and $I_{ik} = \begin{cases} 1 & \text{if } y_i = k \\ 0 & \text{if } y_i \neq k \end{cases}$

(Full points if the derivation is mathematically correct. 2 points if you can describe the procedure for deriving.)

$$L = \prod_{i=1}^{n} \frac{\sum_{k=1}^{K} I_{ik} \exp(W_k^T X)}{\sum_{k'=1}^{K} \exp(W_{k'}^T X)}$$

$$\Rightarrow \log L = \sum_{i=1}^{n} \log \frac{\sum_{k=1}^{K} I_{ik} \exp(W_k^T X)}{\sum_{k'=1}^{K} \exp(W_{k'}^T X)}$$

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{K} I_{ik} \exp(W_k^T X) - \log \sum_{k'=1}^{K} \exp(W_{k'}^T X)$$

$$= \sum_{i=1}^{n} \left(\sum_{k=1}^{K} I_{ik} W_k^T X - \log \sum_{k'=1}^{K} \exp(W_{k'}^T X) \right)$$

(f) **(6 pts)** In class we learned the following PAC learning bound for consistent learners:
Theorem 1. Let H be a finite concept class. Let D be an arbitrary, fixed unknown distribution over X. For any ϵ, $\delta > 0$, if we draw a sample S from D of size

$$m > \frac{1}{\epsilon}\left(ln(|H|) + ln\frac{1}{\delta}\right) \tag{4}$$

then with probability at least $1 - \delta$, all hypothesis $h \in H$ have $err_D(h) \leq \epsilon$. Our friend Kai is trying to solve a learning problem that fits in the assumptions above.

i. Kai tried a training set of 100 examples and observed some test error, so he wanted to reduce the test error to half. How many examples should Kai use, according to the above PAC bound?

he needs 200 examples.

ii. Kai took your suggestion and ran his algorithm again, however the error on the test set did not halve. Do you think it is possible? explain briefly.

It is possible since we can ~~at~~ only have probability at least $1-\delta$ to see $error_D(h) \leq \epsilon$. So if δ here is relatively large, It is possible not see the improvement.

(g) **(4 pts)** List two differences between generative and discriminative learning models.

① generative model models $P(X|Y)$ i.e. how data are generated from each class

while discriminative model models $P(Y|X)$

② discriminative model only gives a boundary to separate one class from other class

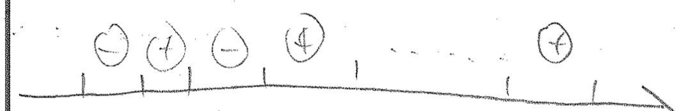
~~wh~~ while generative models ~~di~~ distribution of each class

(h) **(8 pts)** We define a set of functions $T = f(x) = I[x > a] : a \in \mathbb{R}^1$, where $I[x > a]$ is the indicator function returning 1 if $x > a$ and returning 0 otherwise. For input domain $X = \mathbb{R}^1$, and a fixed positive number k, consider a concept class DT_k consisting of all decision trees of depth at most k where the function at each non-leaf node is an element of T. Note that if the tree has only one decision node (the root) and two leaves, then $k = 1$.

Determine the VC dimension of DT_3, and prove that your answer is correct.

First, the VC Dimension of T is 1 since when 2 points labeled as $\oplus \ \ominus \longrightarrow$, T can not shatter it.

We observe that at depth k, DT_k made 2^{k-1} decisions and thus has 2^k intervals. We can represent the k-th layer on $\mathbb{R}^1$ as :



So there is 2^k intervals represented by the function that describes DT_3.

So give 2^k points, DT_3 can shatter them given any labeling $\therefore VC \geq 2^k$

$VC = 2^k + 1$.