1 VC Dimension [15 pts]

We define a set of concepts

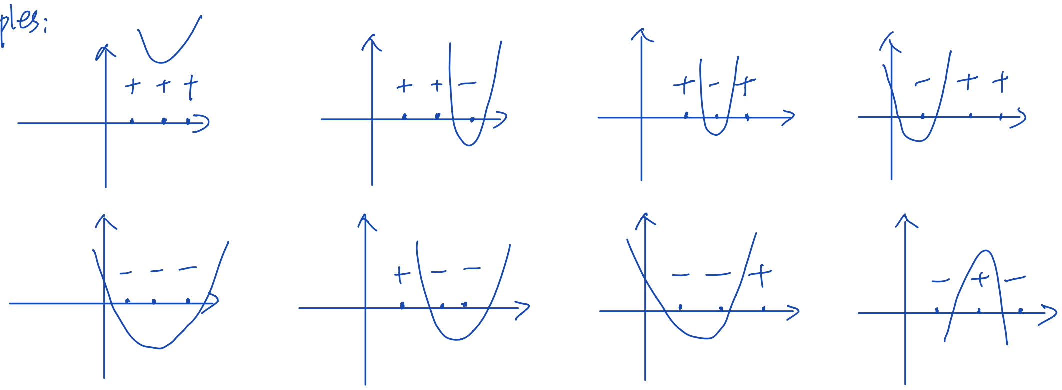
$$H = \{sgn(ax^2 + bx + c); a, b, c, \in R\},$$

where $sgn(\cdot)$ is 1 when the argument $\cdot$ is positive, and 0 otherwise. What is the VC dimension of H? Prove your claim.

$VC(H) = 3$.

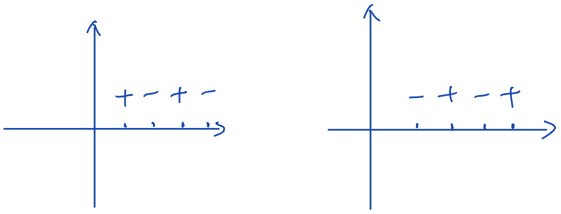
Give examples and counterexamples to prove this.

examples:



They can be shattered in any cases that 3 points in a line.

Counterexamples:



They cannot be shattered as shown.

Since $3 \leq VC(H) < 4$, so $VC(H) = 3$.

2 Kernels [15 pts]

Given vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ in $\mathbb{R}^2$, define the kernel $K_\beta(\boldsymbol{x}, \boldsymbol{z}) = (1 + \beta \boldsymbol{x} \cdot \boldsymbol{z})^3$ for any value $\beta > 0$. Find the corresponding feature map $\phi_\beta(\cdot)$[1]. What are the similarities/differences from the kernel $K(\boldsymbol{x}, \boldsymbol{z}) = (1 + \boldsymbol{x} \cdot \boldsymbol{z})^3$, and what role does the parameter β play?

Assume $D=2$, $x = (x_1, x_2)$, $z = (z_1, z_2)$

$$K_\beta(x, z) = (1+\beta x \cdot z)^3 = [1+\beta(x_1 z_1 + x_2 z_2)]^3$$

$$= 1 + \beta^3(x_1^3 z_1^3 + x_2^3 z_2^3 + 3x_1^2 z_1^2 x_2 z_2 + 3x_1 z_1 x_2^2 z_2^2) + 3\beta(x_1 z_1 + x_2 z_2) + 3\beta^2(x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 x_2 z_2)$$

$$= 1 + 3\beta x_1 z_1 + 3\beta x_2 z_2 + 3\beta^2 x_1^2 z_1^2 + 3\beta^2 x_2^2 z_2^2 + 6\beta^2 x_1 z_1 x_2 z_2 + 3\beta^3 x_1^2 z_1^2 x_2 z_2 + 3\beta^3 x_1 z_1 x_2^2 z_2^2 + \beta^3 x_1^3 z_1^3 + \beta^3 x_2^3 z_2^3$$

$$\phi_\beta(x) = (1, \sqrt{3\beta}\, x_1, \sqrt{3\beta}\, x_2, \sqrt{3}\beta x_1^2, \sqrt{3}\beta x_2^2, \sqrt{6}\beta\, x_1 x_2, \sqrt{3}\beta^{\frac{3}{2}} x_1^2 x_2, \sqrt{3}\beta^{\frac{3}{2}} x_1 x_2^2, \beta^{\frac{3}{2}} x_1^3, \beta^{\frac{3}{2}} x_2^3)^T$$

So K_β is a kernal, since $\overset{\textcircled{1}}{K_\beta} = \phi_B(x)^T \phi_B(z)$, and $\overset{\textcircled{2}}{K_\beta}(x, z) = K_\beta(z, x)$.

Same process as doing $K(x, z) = (1+x \cdot z)^3$:

$$\phi(x) = (1, \sqrt{3}x_1, \sqrt{3}x_2, \sqrt{3}x_1^2, \sqrt{3}x_2^2, \sqrt{6}x_1 x_2, \sqrt{3}x_1^2 x_2, \sqrt{3}x_1 x_2^2, x_1^3, x_2^3)^T.$$

After calculating both the cases, I realize the β plays a important role to scale the terms. The constant term 1 stay unchanged, the 1-st order term was scaled by $\beta^{\frac{1}{2}}$, and the 2-nd order was scaled by β, 3-rd order was scaled by $\beta^{\frac{3}{2}}$. We can verify that the n-th order will be scaled by $\beta^{\frac{n}{2}}$.

when $0 < \beta < 1$, $\beta^{\frac{3}{2}} < \beta < \beta^{\frac{1}{2}}$, so that higher order have less weight.

when $\beta = 1$, $\beta^{\frac{1}{2}} = \beta = \beta^{\frac{3}{2}} = 1$, all of them have same weight, same as $\phi(x)$.

when $\beta > 1$, $\beta^{\frac{1}{2}} < \beta < \beta^{\frac{3}{2}}$, higher order have more weight.

Therefore, the coefficient make $\phi_\beta(x)$ become more flexible than $\phi(x)$.

3 SVM [20 pts]

Suppose we are looking for a maximum-margin linear classifier *through the origin*, i.e. $b = 0$ (also hard margin, i.e., no slack variables). In other words, we minimize $\frac{1}{2}||\boldsymbol{w}||^2$ subject to $y_n \boldsymbol{w}^T \boldsymbol{x}_n \geq 1, n = 1, \ldots, N$.

(a) Suppose we have two training examples, $\boldsymbol{x}_1 = (1, 1)^T$ and $\boldsymbol{x}_2 = (1, 0)^T$ with labels $y_1 = 1$ and $y_2 = -1$. What is $\boldsymbol{w}^*$ in this case?

(b) Suppose we now allow the offset parameter b to be non-zero. How would the classifier and the margin change in the previous question? What are $(\boldsymbol{w}^*, b^*)$? Compare your solutions with and without offset.

(a). Since the SVM uses these points as support (constrain) vectors, so

$$\begin{cases} y_1 w^T x_1 = 1 \\ y_2 w^T x_2 = 1 \end{cases}$$

$w^T x_1 = \frac{1}{y_1} = 1$

$w_1 x_{11} + w_2 x_{12} = 1$

$w_1 \cdot 1 + w_2 \cdot 1 = 1$

$w_1 + w_2 = 1$

$w^T x_2 = \frac{1}{y_2} = -1$

$w_1 x_{21} + w_2 x_{22} = -1$

$w_1 \cdot 1 + w_2 \cdot 0 = -1$

$w_1 = -1$

$$\begin{cases} w_1 = -1 \\ w_2 = 2 \end{cases}$$

$\therefore w^* = [-1, 2]^T$

$\frac{1}{2}||w||^2 = \frac{1}{2}(w_1^2 + w_2^2) = \frac{\sqrt{5}}{2}$

(b) Since the SVM uses these points as support vectors, so .

$$\begin{cases} y_1(w^T x_1 + b) = 1 \\ y_2(w^T x_1 + b) = 1 \end{cases}$$

$w^T x_1 + b = 1$

$w_1 x_{11} + w_2 x_{12} + b = 1$

$w_1 + w_2 + b = 1$

$-w^T x_2 - b = 1$

$-w_1 x_{21} - w_2 x_{22} - b = 1$

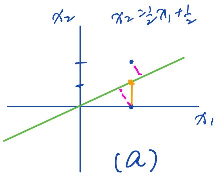
$-w_1 - b = 1$

$$\begin{cases} w_1 = 0 \\ w_2 = 2 \\ b = -1 \end{cases}$$

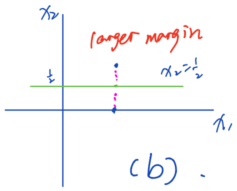
In order to minimize $\frac{1}{2}||w||^2$, so make $w_1 = 0 . b = -1$, then :

$$\begin{cases} w_1 = 0 \\ w_2 = 2 \\ b = -1 \end{cases} \quad \therefore \quad w^* = [0, 2]^T \quad b^* = -1 .$$

$\frac{1}{2}||w||^2 = \frac{1}{2}(w_1^2 + w_2^2) = 2 .$



(a)



(b) .

From both the calculation and graph,

We can know that the classifier with offset has a largeer margin, since $2 < \frac{\sqrt{5}}{2}$.

4. Twitter analysis using SVMs

4.1 Feature Extraction

(a) Implemented
(b) Implemented
(c) Implemented
(d) The output I got was: (560, 1811) (560,) (70, 1811) (70,)
It shows I successfully split it into training set and test set with a correct size.

4.2 Hyper-parameter Selection for a Linear-Kernel SVM

(a) Implemented
(b) It is important to use StratifiedKFold( ) to split the data, because we need the same distributions for both training set and test set. Generally speaking, we need the training and test data to be the same distributions when using any of the classifiers or learning methods. Otherwise, if we have different distributions (i.e. more proportion of positive data in training set than the test set), the accuracy rate we get from test data may be incorrect. It will show up if we do the extreme case: all the positive in the training set and the negative in the test set.
(c) Implemented
(d) The output I got was:

```
Linear SVM Hyperparameter Selection based on accuracy:
[0.70894195 0.71074376 0.80603268 0.81462711 0.81818274 0.81818274]
Linear SVM Hyperparameter Selection based on F1-Score:
[0.82968282 0.8305628  0.87547268 0.87486483 0.87656215 0.87656215]
Linear SVM Hyperparameter Selection based on AUROC:
[0.81054948 0.81107835 0.85755274 0.87123274 0.86957902 0.86957902]
```

Round them to the fourth decimal decimal place, and get the best C as shown in the table:

C	accuracy	F1-score	AUROC
10^(-3)	0.7089	0.8297	0.8105
10^(-2)	0.7107	0.8306	0.8111
10^(-1)	0.8060	0.8755	0.8576
10^0	0.8146	0.8749	0.8712
10^1	0.8182	0.8766	0.8696
10^2	0.8182	0.8766	0.8696
best value	0.8182 (When C=10 or 100)	0.8766 (When C=10 or 100)	0.8712 (When C=1)

4.3 Test Set Performance

(a) Implemented
(b) Implemented
(c) The output I got was:

```
C = 10
Test based on accuracy:0.7428571428571429
Test based on F1-Score:0.43749999999999994
Test based on AUROC:0.7463556851311952
C = 100
Test based on accuracy:0.7428571428571429
Test based on F1-Score:0.43749999999999994
Test based on AUROC:0.7463556851311952
C = 1
Test based on accuracy:0.7428571428571429
Test based on F1-Score:0.47058823529411764
Test based on AUROC:0.7405247813411079
```

Since we already picked C = 10 or 100 for “accuracy”, C = 10 or 100 for “F1-Score”, C = 1 for “AUROC” from previous question. Put these data into the table:

C	accuracy (C=10 or 100)	F1-score (C=10 or 100)	AUROC (C=1)
Test performance	0.7429	0.4375	0.7405