Exercise 5 - Theory and SVM

Yonatan Greenshpan
208406191

1

(a) In order to show that $k = 5k_1 + 4k_2$ is a kernel we need to show that there exists $\varphi: \mathbb{R}^n \to \mathbb{R}^m$ satisfying $\forall x,y \quad k(x,y) = \varphi(x)^T\varphi(y)$.

Given k_1 and k_2 are kernels we conclude that there are two functions $\varphi_1(x), \varphi_2(x)$ satisfying

$k_1(x,y) = \varphi_1(x)^T\varphi_1(y)$

$k_2(x,y) = \varphi_2(x)^T\varphi_2(y)$

For $k(x,y)$ the following equation holds:

$$k(x,y) = 5k_1(x,y) + 4k_2(x,y) = 5\varphi_1(x)\varphi_1(y) + 4\varphi_2(x)\varphi_2(y) = (\sqrt{5}\varphi_1(x), 2\varphi_2(x)) \cdot (\sqrt{5}\varphi_1(y), 2\varphi_2(y))$$

We show that there exist functions $\varphi(x) = (\sqrt{5}\varphi_1(x), 2\varphi_2(x))$ that satisfies $k(x,y) = \varphi(x)\cdot\varphi(y)$ and therefore $k = 5k_1 + 4k_2$ is a kernel.

(c) We are looking for $\varphi(x)$ such that:

$$k(x,y) = 3\min(x,y) = \varphi(x)\cdot\varphi(y)$$

We will define $\varphi(x)$ as follow:

$\varphi: S \to \mathbb{R}^N$

$$\varphi(x) = \vec{u}(u_1,\dots,u_N) = (\overbrace{3,\dots,3}^{x},0,\dots,0)$$

$u_1 = u_2 = \dots = u_x = 3$

$u_{x+1} = u_{x+2} = \dots = u_N = 0$

$$k(x,y) = \varphi(x)\cdot\varphi(y) = (3,\dots,3,0,\dots,0)\cdot(3,\dots,3,0,\dots,0) = \underbrace{3\cdot3+\dots+3\cdot3}_{\min(x,y)} + 0\cdot0+\dots+0\cdot0 =$$

$$= 3\cdot3+\dots+3\cdot3 = 3\cdot3\,\min(x,y) = 9\min(x,y)$$

(b) Notations: $\varphi_1: \mathbb{R}^n \to \mathbb{R}^m \quad m \geq n$

$\varphi_2: \mathbb{R}^n \to \mathbb{R}^d \quad d \geq n$

$w = (w_0,\dots,w_m)$ the linear seperator of φ_1

It is given that the data is linear seperable in $\mathbb{R}^m \Rightarrow \forall x^i \in X: \operatorname{sgn}(w\varphi_1(x^i)) = y^i$

We will define: $w' = (w'_1,\dots,w'_m,\dots,w'_{m+k})$. We want to show $w'\varphi(x) = w\varphi_1(x)$.

It holds that $(w'_1,\dots,w'_{m+k})\begin{bmatrix}x_1\\ \vdots \\ x_{m+k}\end{bmatrix} = w'_1x_1+\dots+w'_mx_m+\dots+w'_{m+k}x_{m+k} = w_1\sqrt{5}x_1+\dots+w_m\sqrt{5}x_m+\dots+w'_{m+k}x_{m+k}$

$$\Downarrow$$

$$(w'_1,\dots,w'_m,\dots,w'_{m+k}) = \underbrace{\left(\frac{w_1}{\sqrt{5}},\dots,\frac{w_m}{\sqrt{5}},0,\dots,0\right)}$$

These are the weights that seperate the data in $m+k$ dimensions.

② We will solve the following optimization problem:

$$\max R(h,S) = 200 \cdot h^{\frac{2}{3}} \cdot S^{\frac{1}{3}}$$

s.t.

$$170S + 20h = 20{,}000 \Rightarrow 17S + 2h = 2000$$

We will do so with the Lagrange multipliers.

$$L(h,S,\lambda) = 200\, h^{\frac{2}{3}} S^{\frac{1}{3}} - \lambda(2h + 17S - 2000)$$

$R(h,S)$ is maximal when.

(1) - $\frac{\partial L}{\partial h} = 0 \Rightarrow \frac{400}{3} h^{-\frac{1}{3}} S^{\frac{1}{3}} - 2\lambda = 0$

(2) - $\frac{\partial L}{\partial S} = 0 \Rightarrow \frac{200}{3} h^{\frac{2}{3}} S^{-\frac{2}{3}} - 17\lambda = 0$

(3) - $\frac{\partial L}{\partial \lambda} = 0 \Rightarrow 2h + 17S = 2000$

(1) $\frac{400}{6}\left(\frac{S}{h}\right)^{\frac{1}{3}} = \lambda$

(2) $\frac{200}{51}\left(\frac{h}{S}\right)^{\frac{2}{3}} = \lambda$

(1)=(2) $\frac{200}{3}\left(\frac{S}{h}\right)^{\frac{1}{3}} = \frac{200}{51}\left(\left(\frac{h}{S}\right)^{\frac{1}{3}}\right)^{-1} = \frac{200}{51}\left(\left(\frac{S}{h}\right)^{-\frac{1}{3}}\right)^{-1}$

$\left(\frac{S}{h}\right)^{\frac{1}{3}} = \alpha$

$\frac{200}{3}\alpha = \frac{200}{51}(\alpha^{-1})^{-1}$

$51\alpha = \frac{3}{\alpha^2}$

$\alpha^3 = \frac{3}{51}$

$\frac{S}{h} = \frac{3}{51}$

$3h = 51S$

$h = 17S$

(3) - $2 \cdot 17S + 17S = 2000$

$\boxed{S = 39.22}$

$\boxed{h = 666.67}$

This mean that we will employ 667 hours of labor and by 39 tons of steel and it will give us $\approx 51{,}777$ l

(3)

(a) $VC(H) = 2$

Let $d_1, d_2 \in D$. We will show that all subsets of $\{d_1, d_2\}$ are consistent with H. The empty set is trivial. For subset of one point only we will choose d_1 without limitation of generality and denote $d_1 = (x_1, y_1)$ and r is the distance from the origin. $h = \{(x_1, x_2) \mid \begin{matrix} x_1^2 + x_2^2 \geq 0 \\ x_1^2 + x_2^2 \leq r + \varepsilon \end{matrix}, 0 < \varepsilon \in \mathbb{R}\}$ contain d_1 and with the same logic it holds for d_2. In case of the subset $\{d_1, d_2\}$ and the corresponding distances from the origin r_1 and r_2 we can conclude that the following hypothesis is consistent with $\{d_1, d_2\}$: $h = \{(x_1, x_2) \mid \begin{matrix} (x_1^2 + x_2^2) \geq \min(r_1, r_2) \\ (x_1^2 + x_2^2) \leq \max(r_1, r_2) \end{matrix}\}$. All together we get that $VC(H) \geq 2$.

However, for three points p_1, p_2, p_3 with $r_1 < r_2 < r_3$ and $r_1 = r_2 < r_3$ it is impossible for any h to be consistent with the subset $\{p_1, p_3\}$ because from the definition of annulus if p_1 in h then p_2 in h too. Therefore $VC(H) < 3$.

All together $2 \leq VC(H) < 3 \Rightarrow VC(H) = 2$

(b) Here is an example for algorithm that learn concept $c \in C$ using the hypothesis $h \in H$. The input of the algorithm is $D = \{x_i\}_{i=1}^n$ labeled data and its output is $h \in H$ that tries to approximate c.

The algorithm:

Algorithm:

Initialize $r_1 = 0, r_2 = \infty$; $c = \{(x_1, x_2) \mid \begin{matrix} x_1^2 + x_2^2 \leq r_2 \\ x_1^2 + x_2^2 \geq r_1 \end{matrix}\}$

For each x in D:

  if x in C: keep going to the next point

  else:

    $r_1 = \max\{r_1, x_1^2 + x_2^2\}$

    if $r_1 = r_2$ return $h(r_1, r_2)$ and finish

Return $h(r_1, r_2)$

⊛ c isn't a variable but just a notation to the concept. The update r_1 or r_2 c is updated too.

Regarding time complexity the algorithm iterate over all the data once

!!! Continue next Page !!!

The algorithm itself keep that the closest to the origin point that classified as 1 define r_1 and the furthest point that classified as 1 defined r_2. In this way the algorithm L ensures that H captures C correctly and it achieves consistency.

To calculate the sample complexity let define 2 circles c_1 and c_2 with radius $r_1 - r_2$. The probability of all possitive points falling withing either of the 2 circles is at most $(1-\frac{\varepsilon}{2})^m$ and the probability not falling in any circle is $2(1-\frac{\varepsilon}{2})^m$

We look for m s.t. $P(err \geq \varepsilon) \leq 2(1-\frac{\varepsilon}{2})^m \leq \delta$

$P(err \geq \varepsilon) \leq 2e^{-\frac{\varepsilon m}{2}} \leq \delta \quad / \ln$

$\ln(\frac{\delta}{2}) \geq -\frac{\varepsilon m}{2}$

$\frac{m}{2} \geq \frac{\ln(\frac{\delta}{2})}{-\varepsilon} = \frac{\ln(\frac{2}{\delta})}{\varepsilon}$

The sample complexity is $m \geq \frac{1}{\varepsilon} \cdot 2\ln(\frac{2}{\delta})$

(c) In (a) we showed that $VC(H) = 2$ and therefore we will get the boundary:

$m \geq \frac{1}{\varepsilon}\left(4\log\frac{2}{\delta} + 8VC(H)\log_2\frac{13}{\varepsilon}\right) = 2993$

In (b) we get $m \geq \frac{1}{0.05} \cdot 2\ln\left(\frac{2}{0.05}\right) = 147$

As we can see, the boundary found in (b) is much more tight than the general formula and we got smaller m in it. It makes sense that (b) requires smaller m because it uses additional information such as the specific geometry of the hypothesis space.

4

(a) $X = \mathbb{R}$

$n \in \mathbb{N}$

H_m - hypothesis space of all "x-node decision tree" with $n \leq m$

We get $VC(H_3) = 4$. and in order to prove it we need to show that there exists a set of 4 points that shattered H and that there is no set of 5 points that can.

Shattering four points: For $+++-$ we can easily shatter with three nodes decision tree (we can think of each node as a partition on one dimensional axis ~~blablabla~~) $\Rightarrow VC(H_3) \geq 4$

Inability to shatter five points: in order to show that it isn't possible we will find a labeling of five points that can't be capture by any 3 node decision tree. For $++++-$ we allways will have 2 points of the same label on one side of split violating the classification. $\Rightarrow VC(H_3) < 5$

$$4 \leq VC(H_3) < 5$$

$$\Downarrow$$

$$VC(H_3) = 4$$

(b) $VC(H_m) = 2^{m-1}$

In order to prove it we will show that H_m shattering 2^{m-1} points but can't shatter 2^m points.

Considered 2^{m-1} points labeled with all possible combinations these points can be perfectly shattered by m simply mapping each combination to a unique decision tree with $m-1$ nodes $\Rightarrow VC(H_m) \geq 2^{m-1}$

We will use the piegon hole principle to show that there is no set of 2^m that H_m can shatter. The largest decision tree in H_m has 2^m leaf nodes. However, if we have $2^{m-1}+1$ points by the piegonhole principle we must have at least 2 points mapped to the same node - which mean it will misclassify at least one point. $\Rightarrow VC(H_m) < 2^{m-1}+1$

All in all: $VC(H_m) = 2^{m-1}$