Introduction to Learning and Analysis of Big Data HW 3

Tomer Elgavish, Shay Kricheli

January 2020

Question 1

(b)

In this section we were asked to perform 10-fold cross-validation in order to tune λ and σ from two given sets: $\lambda \in \{0.01, 0.1, 1\}$, $\sigma \in \{0.01, 0.05, 1, 2\}$ The following table exhibits the average cross-validation error for each pair of parameters:

$\sigma \setminus \lambda$	**0.01**	**0.1**	**1**
0.01	0.08	0.08	0.08
0.05	0.063	0.063	0.063
1	0.0765	0.0835	0.0840
2	0.0910	0.1405	0.1405

Table 1: Q1.b Cross Validation Error

The following table exhibits the error measured on the test set when learning from the entire training set:

$\sigma \setminus \lambda$	**0.01**	**0.1**	**1**
0.01	0.06	0.06	0.06
0.05	0.04	0.04	0.04
1	0.05	0.06	0.06
2	0.07	0.15	0.15

Table 2: Q1.b Error When Learning from Entire Test Set

The following table exhibits the difference between the values from the two tables above:

$\sigma \setminus \lambda$	**0.01**	**0.1**	**1**
0.01	0.02	0.02	0.02
0.05	0.023	0.023	0.023
1	0.0265	0.0235	0.0240
2	0.0210	0.0095	0.0095

Table 3: Q1.b Difference Between The Errors

- As one can see, the optimal pairs acquired from the cross-validation are $\sigma = 0.05$ and $\lambda = 0.01$, $\sigma = 0.05$ and $\lambda = 0.1$, $\sigma = 0.05$ and $\lambda = 1$.

- When training on the entire training set, one can also see that the optimal pairs are also $\sigma = 0.05$ and $\lambda = 0.01$, $\sigma = 0.05$ and $\lambda = 0.1$, $\sigma = 0.05$ and $\lambda = 1$.
- Since the cross validation procedure gave the optimal pairs for σ and λ, we can conclude it was indeed successful. That being said, even though the cross validation procedure was successful, in this case it may be redundant since the optimal pairs can be acquired when training on the entire test set which is less computationally-intensive as we witnessed when running the algorithm in both cases.

(c)

In this question we are asked to set $\lambda = 0.01$ and consider $\sigma \in \{0.01, 0.05, 1, 2\}$. For these values, we were asked to run the soft-margin RBF SVM and plot the function that each of the resulting classifiers induces on the original $\mathbb{R}^2$ space. The following plots display the heatmaps produced from the corresponding values of σ. The x and y axes values range between the values -15 and 15.

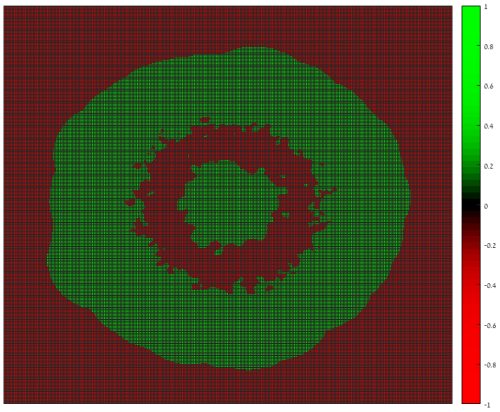

Figure 1: Q1.c - Heatmap for $\sigma = 0.01$

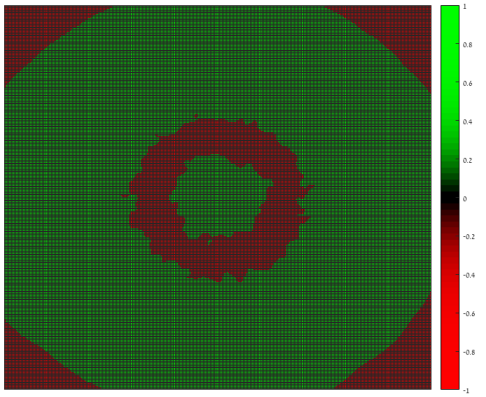

Figure 2: Q1.c - Heatmap for $\sigma = 0.05$

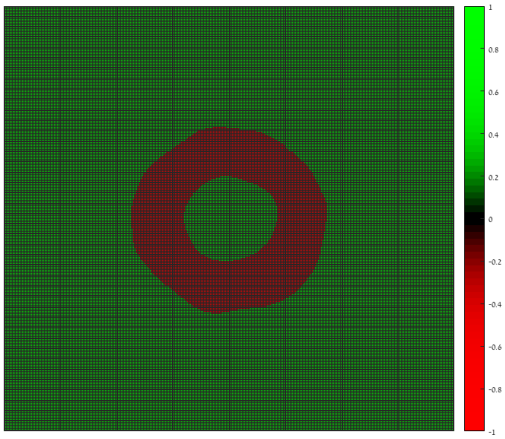

Figure 3: Q1.c - Heatmap for $\sigma = 1$

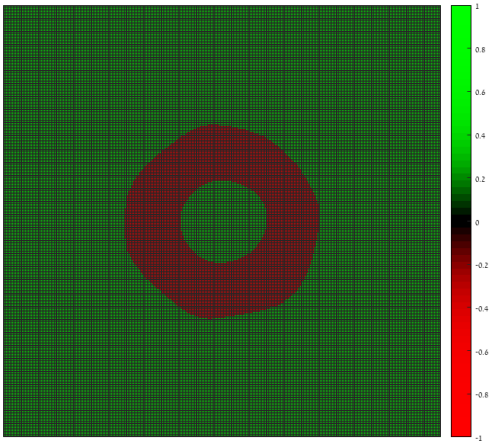

Figure 4: Q1.c - Heatmap for $\sigma = 2$

As one can see, each classifier produced by the algorithm induces several ring-shaped contours in the $\mathbb{R}^2$ space for values of x and y between -15 and 15. For small values of σ we see that the shape is more rugged. This is caused due to the fact that the Gaussian kernel values increase with the σ parameter and thus smaller values for σ will result in smaller values for the Gaussian kernel. Since the hypothesis is given by the sign of the inner product of the α vector produced by the soft-margin RBF SVM algorithm and the Gaussian kernel, for smaller values of the Gaussian kernel, there will be larger sensitivity for the values of the vector α for the sign function. Thus the result is increased sensitivity for close examples around the ring shapes that produce a more "noisy" output. For larger values of σ this effect subsides and we see that the ring shapes become more and more smooth and less sensitive to close examples.

Question 2

Let $\mathcal{X}$ be the set of all undirected graphs over n vertices numbered $1, ..., n$ with degree at most 5. Let $\mathcal{Y} = \{0, 1\}$. For a graph $x \in \mathcal{X}$, let us define the mapping $g : \mathcal{X} \to \mathbb{N}^n$ where the i'th coordinate of the vector $g(x)$ is the degree of vertex i in graph x. Let:

$$\mathcal{H} = \{h_v : \mathcal{X} \to \mathcal{Y} \mid v \in \mathbb{N}^n, h_v \not\equiv 0\} \ ; \ h_v(x) = \mathbb{I}[g(x) = v]$$

Let D be a distribution over $\mathcal{X} \times \mathcal{Y}$ and suppose that in this distribution, the label of a graph x is a deterministic function of $g(x)$.

(a)

In this section we are to show that the sample complexity of learning $\mathcal{H}$ as a function of n is $\Omega(n)$ using the cardinality of $\mathcal{H}$ and the PAC-learning lower bound we saw in class. First, let us consider the cardinality of $\mathcal{H}$: $\mathcal{H}$ is a set of hypothesis functions h_v that is each defined by a corresponding vector $v \in \mathbb{N}^n$ of degrees of the vertices for a graph $x \in \mathcal{X}$. Let us notice that the amount of these vectors is at most, the number of possibilities to produce an n-dimensional vector of integers in the range of 0 to 5. There are 6 possibilities for each coordinate in the vector and thus this amount is the number of series in length n with 6 possible values for each element - which is 6^n. Thus:

$$|\mathcal{H}| \leq 6^n \ (*)$$

Let $m_{\mathcal{H}}(\varepsilon, \delta)$ be the minimal amount of examples needed to learn $\mathcal{H}$. Let us assume the general agnostic case - for which we saw in class that:

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{2}{\varepsilon^2}\Big(\log(|\mathcal{H}|) + \log(2/\delta)\Big)$$

From $(*)$:

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{2}{\varepsilon^2}\Big(\log(6^n) + \log(2/\delta)\Big) = \frac{2}{\varepsilon^2}\Big(n\log(6) + \log(2/\delta)\Big) \to m_{\mathcal{H}}(\varepsilon, \delta) = O(n) \ \blacksquare$$

(b)

In class we saw that for the realizable case - $m_{\mathcal{H}}(\varepsilon, \delta)$ is inverse to first power of ε - meaning that as we require less error, we also require more examples at a rate of $\theta(\varepsilon^{-1})$. In the agnostic setting, where we make no assumption of realizability - $m_{\mathcal{H}}(\varepsilon, \delta)$ is inverse to second power of ε - meaning that as we require less error, we also require more examples at a rate of $\theta(\varepsilon^{-2})$ - which is "worse" since we require much more examples for the same change of the value of ε. In this question we are asked whether we can assume that the problem is realizable and thus use the "better" dependence on

ε. We'll show that the best dependence we can get is that of the agnostic case. To do so, we'll show that $\mathcal{H}$ is not realizable - meaning that there is no $h^* \in \mathcal{H}$ such that $err(h^*, \mathcal{D}) = 0$. Let

$$n = 2 \rightarrow V = \{u_1, u_2\}$$
$$\mathcal{X} = \{x_1, x_2\}$$

where:

$$x_1 = G_1 = (V, E_1) \ ; \ E_1 = \emptyset$$
$$x_2 = G_2 = (V, E_2) \ ; \ E_2 = \{(u_1, u_2)\}$$

Let:

$$\mathcal{D} = \{(x_1, 1), (x_2, 1)\}$$

By the definition of the hypothesis class, we'll get:

$$\mathcal{H} = \{h_{v_1}, h_{v_2}\}$$

where:

$$h_{v_1}(x) = \mathbb{I}[g(x) = v_1]$$
$$h_{v_2}(x) = \mathbb{I}[g(x) = v_2]$$

By the definition of g:

$$g(x_1) = (0, 0)$$
$$g(x_2) = (1, 1)$$

Let us assume without loss of generality, by the definition of $\mathcal{H}$:

$$v_1 = (0, 0)$$
$$v_2 = (1, 1)$$

Thus:

$$h_{v_1}(x_1) = \mathbb{I}[g(x_1) = v_1] = \mathbb{I}[(0, 0) = (0, 0)] = 1 \ ; \ h_{v_1}(x_2) = \mathbb{I}[g(x_2) = v_1] = \mathbb{I}[(1, 1) = (0, 0)] = 0$$
$$h_{v_2}(x_1) = \mathbb{I}[g(x_1) = v_2] = \mathbb{I}[(0, 0) = (1, 1)] = 0 \ ; \ h_{v_2}(x_2) = \mathbb{I}[g(x_2) = v_2] = \mathbb{I}[(1, 1) = (1, 1)] = 1$$

We see that there isn't any $h^* \in \mathcal{H}$ that labels all the examples in the distribution correctly and thus $\mathcal{H}$ is not realizable and the best dependence we can get is:

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{2}{\varepsilon^2}\Big(n \log(6) + \log(2/\delta)\Big)$$

(c)

Let us assume (as was mentioned in the course forum) that $n > 1$. We'll show that:

$$VC(\mathcal{H}) = 1$$

Since we assumed that $n > 1$ - we can thus conclude that $VC(\mathcal{H}) \geq 1$. Let us assume without loss of generality that $n = 2$ and we'll show that $VC(\mathcal{H}) \neq 2$. This case corresponds to the same case presented in the last section - as we saw that two examples cannot be correctly labeled by any one hypothesis function from $\mathcal{H}$ and thus $VC(\mathcal{H}) \neq 2$. Thus by the definition of VC - we'll get that $VC(\mathcal{H}) = 1$. Thus, we can get a better upper bound for the sample complexity of learning $\mathcal{H}$ as a function of n and ε:

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{2}{\varepsilon^2}\Big(VC(\mathcal{H}) + \log(2/\delta)\Big) = \frac{2}{\varepsilon^2}\Big(1 + \log(2/\delta)\Big)$$

Question 3

Let $S = \{(x_1, y_1), \cdots, (x_m, y_m)\}$ and let us consider a Gradient Descent algorithm that attempts to minimize the following objective:

$$\min_{w \in \mathbb{R}^d} \left[\lambda ||w|| + \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2 \right]$$

(a)

In this section we are to show that the objective is convex. Let us denote:

$$C = \mathbb{R}^d \ ; \ f(w) = \lambda ||w|| + \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$

Thus we can denote the objective as:

$$\min_{w \in C} f(w)$$

To prove convexity of this objective, we need to prove that C and f is convex:

- C is convex:
  $C = \mathbb{R}^d$ and thus let $a \in [0, 1]$ and $u, v \in \mathbb{R}^d$. The vector $\alpha u + (1 - \alpha)v$ is a linear combination of two vectors in $\mathbb{R}^d$ and thus by definition it is also in $\mathbb{R}^d$. Thus C is convex.

- f is convex:
  Let us denote:
  $$g_1(w) = \lambda ||w|| \ ; \ g_2(w) = \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$
  Then we get: $f(w) = g_1(w) + g_2(w)$. f is a conic combination of $g_1(w)$ and $g_2(w)$. First, we'll prove that $g_1 : \mathbb{R}^d \to \mathbb{R}$ is convex by definition:
  Let $u, v \in \mathbb{R}^d$, $\alpha \in [0, 1]$. g_1 is convex if: $g_1(\alpha u + (1 - \alpha)v) \leq \alpha g_1(u) + (1 - \alpha) g_1(v)$. Now:
  $$g_1(\alpha u + (1 - \alpha)v) = \lambda ||\alpha u + (1 - \alpha)v|| \overset{1}{\leq} \lambda \left[||\alpha u|| + ||(1 - \alpha)v|| \right] \overset{2}{=} \lambda \left[\alpha ||u|| + (1 - \alpha)||v|| \right] =$$
  $$= \alpha \cdot (\lambda ||u||) + (1 - \alpha) \cdot (\lambda ||v||) = \alpha \cdot g_1(u) + (1 - \alpha) \cdot g_1(v) \ \blacksquare$$
  The inequality in 1 is due to applying the triangular inequality of norms and the equality in 2 is due to the property of absolute scalability of norms.
  Now we'll show that $g_2 : \mathbb{R}^d \to \mathbb{R}$ is convex. Let us define:
  $$g_{2_i}(w) = (\langle w, x_i \rangle - y_i)^2 \ ; \ \forall 1 \leq i \leq m$$
  Thus:
  $$g_2(w) = \sum_{i=1}^{m} g_{2_i}(w)$$
  as in g_2 is a conic combination of all m functions g_{2_i}. Now we'll show that each function g_{2_i} is convex. Let us define:
  $$h_{2_i}(z) = (z - y_i)^2 \ ; \ \forall 1 \leq i \leq m$$
  Thus:
  $$g_{2_i}(w) = h_{2_i}(\langle w, x_i \rangle) \ ; \ \forall 1 \leq i \leq m$$

Now:

$$\frac{\partial^2}{\partial z^2} h_{2_i}(z) = 2 \geq 0$$

and thus $h_{2_i}(z)$ is convex by the second derivative test. Since inner product is a linear function, we'll get that $g_{2_i}(w)$ is a composition of a convex and linear functions and thus convex. Thus $g_2(w)$ is convex and therefore f is convex. Hence any local minimum it has is also global. Therefore, running gradient descent will be efficient in finding the minimum of f, and will output a vector w close to the global minimum.

(b)

Suppose that Gradient Descent is run on S with a step size η. Now, by the Gradient Descent algorithm definition:

$$w^{(t+1)} = w^{(t)} - \eta \nabla f(w^{(t)})$$

where $w^{(t)}$ and η are given. Let us then calculate the gradient of $f(w^{(t)})$ by definition:

$$\nabla f(w^{(t)}) = \left(\frac{\partial f(w^{(t)})}{\partial w^{(t)}(1)}, \cdots, \frac{\partial f(w^{(t)})}{\partial w^{(t)}(d)} \right)$$

Let us consider the partial derivative of $f(w^{(t)})$ by $w^{(t)}(j)$ - the j'th coordinate of $w^{(t)}$:

$$\frac{\partial f(w^{(t)})}{\partial w^{(t)}(j)} = \frac{\partial}{\partial w^{(t)}(j)} \left[\lambda ||w^{(t)}|| + \sum_{i=1}^{m} (\langle w^{(t)}, x_i \rangle - y_i)^2 \right] = \lambda \frac{\partial \, ||w^{(t)}||}{\partial w^{(t)}(j)} + \frac{\partial}{\partial w^{(t)}(j)} \sum_{i=1}^{m} (\langle w^{(t)}, x_i \rangle - y_i)^2 =$$

$$= \lambda \frac{\partial}{\partial w^{(t)}(j)} \sqrt{\sum_{i=1}^{d} (w^{(t)}(i))^2} + \frac{\partial}{\partial w^{(t)}(j)} \sum_{i=1}^{m} \left(\sum_{k=1}^{d} w^{(t)}(k) \cdot x_i(k) - y_i \right)^2 =$$

$$= \frac{2\lambda w^{(t)}(j)}{2\sqrt{\sum_{i=1}^{d} (w^{(t)}(i))^2}} + \sum_{i=1}^{m} 2(\langle w^{(t)}, x_i \rangle - y_i) \cdot x_i(j) = \lambda \frac{w^{(t)}(j)}{||w^{(t)}||} + 2 \sum_{i=1}^{m} x_i(j) \cdot (\langle w^{(t)}, x_i \rangle - y_i)$$

Now, plugging in, we'll get:

$$w^{(t+1)} = w^{(t)} - \eta \left(\lambda \frac{w^{(t)}(1)}{||w^{(t)}||} + 2 \sum_{i=1}^{m} x_i(1) \cdot (\langle w^{(t)}, x_i \rangle - y_i), \cdots, \lambda \frac{w^{(t)}(d)}{||w^{(t)}||} + 2 \sum_{i=1}^{m} x_i(d) \cdot (\langle w^{(t)}, x_i \rangle - y_i) \right) =$$

$$= \left(w^{(t)}(1) \left(1 - \frac{\eta \lambda}{||w^{(t)}||} \right) - 2\eta \sum_{i=1}^{m} x_i(1) \cdot (\langle w^{(t)}, x_i \rangle - y_i), \cdots, w^{(t)}(d) \left(1 - \frac{\eta \lambda}{||w^{(t)}||} \right) - 2\eta \sum_{i=1}^{m} x_i(d) \cdot (\langle w^{(t)}, x_i \rangle - y_i) \right) =$$

$$= w^{(t)} \left(1 - \frac{\eta \lambda}{||w^{(t)}||} \right) - 2\eta \left(\sum_{i=1}^{m} x_i(1) \cdot (\langle w^{(t)}, x_i \rangle - y_i), \cdots, \sum_{i=1}^{m} x_i(d) \cdot (\langle w^{(t)}, x_i \rangle - y_i) \right)$$

(c)

Suppose that Stochastic Gradient Descent is run on S with a step size η. Let $1 \leq i \leq m$ be the index of the random example chosen at the $t+1$ iteration of the algorithm. Now, by the Stochastic Gradient Descent algorithm definition:

$$w^{(t+1)} = w^{(t)} - \eta \left(\nabla R(w^{(t)}) + \nabla \ell \left(w^{(t)}, (x_i, y_i) \right) \right)$$

where $R(w^{(t)}) = \lambda||w^{(t)}||$, $\ell\Big(w^{(t)}, (x_i, y_i)\Big) = (\langle w^{(t)}, x_i\rangle - y_i)^2$ and $w^{(t)}$, η are given. In the last section we already calculated the gradient of $R(w^{(t)})$:

$$\nabla R(w^{(t)}) = \left(\frac{\partial}{\partial w^{(t)}(1)}[\lambda||w^{(t)}||], \cdots, \frac{\partial}{\partial w^{(t)}(d)}[\lambda||w^{(t)}||]\right) = \left(\lambda\frac{w^{(t)}(1)}{||w^{(t)}||}, \cdots, \lambda\frac{w^{(t)}(d)}{||w^{(t)}||}\right)$$

Thus, we are now left with calculating the gradient of $\ell\Big(w^{(t)}, (x_i, y_i)\Big)$:

$$\nabla\ell\Big(w^{(t)}, (x_i, y_i)\Big) = \left(\frac{\partial}{\partial w^{(t)}(1)}\Big[(\langle w^{(t)}, x_i\rangle - y_i)^2\Big], \cdots, \frac{\partial}{\partial w^{(t)}(d)}\Big[(\langle w^{(t)}, x_i\rangle - y_i)^2\Big]\right) =$$

$$= \Big(2\cdot(\langle w^{(t)}, x_i\rangle - y_i)\cdot x_i(1), \cdots, 2\cdot(\langle w^{(t)}, x_i\rangle - y_i)\cdot x_i(d)\Big)$$

Now, plugging in, we'll get:

$$w^{(t+1)} = w^{(t)} - \eta\left(\left(\lambda\frac{w^{(t)}(1)}{||w^{(t)}||}, \cdots, \lambda\frac{w^{(t)}(d)}{||w^{(t)}||}\right) + \Big(2\cdot(\langle w^{(t)}, x_i\rangle - y_i)\cdot x_i(1), \cdots, 2\cdot(\langle w^{(t)}, x_i\rangle - y_i)\cdot x_i(d)\Big)\right) =$$

$$= \left(w^{(t)}(1)\left(1 - \frac{\eta\lambda}{||w^{(t)}||}\right) - 2\cdot\eta x_i(1)\cdot(\langle w^{(t)}, x_i\rangle - y_i), \cdots, w^{(t)}(d)\left(1 - \frac{\eta\lambda}{||w^{(t)}||}\right) - 2\cdot\eta x_i(d)\cdot(\langle w^{(t)}, x_i\rangle - y_i)\right) =$$

$$= w^{(t)}\left(1 - \frac{\eta\lambda}{||w^{(t)}||}\right) - 2\eta\Big(x_i(1)\cdot(\langle w^{(t)}, x_i\rangle - y_i), \cdots, x_i(d)\cdot(\langle w^{(t)}, x_i\rangle - y_i)\Big) =$$

$$= w^{(t)}\left(1 - \frac{\eta\lambda}{||w^{(t)}||}\right) - 2\eta(\langle w^{(t)}, x_i\rangle - y_i)\cdot x_i$$

Question 4

Let us consider a space of examples $\mathcal{X} = \mathbb{R}^d$.

(a)

Let's assume by contradiction that the following function $K(x, x') = -x(1)x'(1)$ is a kernel function for a feature mapping $\psi : \mathcal{X} \to \mathcal{F}$. Therefore, for the case $x' = x$, we get:

$$K(x, x) = -x(1)x(1) = -x(1)^2 = \langle\psi(x), \psi(x)\rangle$$

We have that $-x(1)^2 \leq 0$ since we have a minus sign in front of a squared term, while $\langle\psi(x), \psi(x)\rangle \geq 0$ by a property of inner products. By the assumption we have $-x(1)^2 = \langle\psi(x), \psi(x)\rangle$ and thus $K(x, x) = 0, \forall x \in \mathcal{X}$. Let us consider the following vector: $z = (1, \underbrace{0, 0, \cdots, 0}_{\text{d-1}})^T \in \mathcal{X}$. From the conclusion, we get: $K(z, z) = -z(1)^2 = 0$ but $-z(1)^2 = -1 \neq 0$, in contradiction to the assumption that $K(x, x')$ is a kernel function. Therefore, for any feature mapping ψ, $K(x, x')$ cannot be a kernel function. ■

(b)

Like in the previous sections, let's assume by contradiction that the following function $K(x,x') = \big(x(1)+x(2)\big)\big(x'(3)+x'(4)\big)$ is a kernel function for a feature mapping $\psi : \mathcal{X} \to \mathcal{F}$. Again, for the case $x' = x$, we get:

$$K(x,x) = \big(x(1)+x(2)\big)\big(x(3)+x(4)\big) = \langle \psi(x), \psi(x) \rangle$$

Again, we have $\langle \psi(x), \psi(x) \rangle \geq 0$. Let us consider the following vector $z = (1, 2, -1, -2, \underbrace{0, 0, \cdots, 0}_{\text{d-4}})^T \in \mathcal{X}$. Since $z(3) = -z(1)$ and $z(4) = -z(2)$ we'll have:

$$K(z,z) = \big(z(1)+z(2)\big)\big(z(3)+z(4)\big) = \big(z(1)+z(2)\big)\big(-z(1)-z(2)\big) = -\big(z(1)+z(2)\big)^2 \leq 0$$

Thus we'll have $0 = K(z,z) = -\big(z(1)+z(2)\big)^2 = -(1+2)^2 = -9$ in contradiction. ■

Question 5

In an election poll, n random draws of voters are selected. Each of the voters is asked to which party they like. There are 30 possible parties. Let us assume that all people gave one of the parties as an answer. Let p_i be the fraction of people in the poll who answered that they will vote for party i. In this question we are to use Hoeffding's bound to calculate the smallest number n such that with a probability of at least 97%, for all $i \in \{1, ..., K\}$ the proportion in the population of people who like party i is within 5% from p_i.
Let us define a set of n random variables for each party $i \in \{1, ..., K\}$:

$$\{z_{i_j}\}_{j=1}^{n}$$

Such that:

$$z_{i_j} = \mathbb{I}[\text{the } i\text{'th voter voted for the } j\text{'th party}]$$

By the definition of p_i we'll get:

$$p_i = \frac{1}{n}\sum_{j=1}^{n} z_{i_j}$$

Let ψ_i be the actual proportion in the population of people who like party i. Thus we are looking for the minimal n that holds:

$$\mathbb{P}\left[\bigwedge_{i=1}^{K}\Big(|p_i - \psi_i| < 0.05\Big)\right] > 0.97$$

This corresponds to:

$$\mathbb{P}\left[\bigvee_{i=1}^{K}\Big(|p_i - \psi_i| \geq 0.05\Big)\right] \leq 1 - 0.97 = 0.03$$

By the union bound, we have:

$$\mathbb{P}\left[\bigvee_{i=1}^{K}\Big(|p_i - \psi_i| \geq 0.05\Big)\right] \leq \sum_{i=1}^{K}\mathbb{P}\Big[|p_i - \psi_i| \geq 0.05\Big]$$

We'll want to show that the sum on the right is less or equal to 0.03 and from that we'll infer what is required. To do so, let us consider the i'th party and use Hoeffding's bound for the value of $\frac{0.03}{K}$. From that we'll get:

$$\sum_{i=1}^{K}\mathbb{P}\Big[|p_i - \psi_i| \geq 0.05\Big] \leq \sum_{i=1}^{K}\frac{0.03}{K} = K \cdot \frac{0.03}{K} = 0.03$$

Therefore let us consider Hoeffding's bound for the i'th party:

$$\mathbb{P}\left[|p_i - \psi_i| \geq 0.05\right] \leq 2e^{-2 \cdot 0.05^2 \cdot n} \leq \frac{0.03}{K}$$

Solving for n we get:

$$n \geq \frac{\ln\left(\frac{2}{0.03}\right) + \ln K}{2 \cdot 0.05^2}$$

For $K = 30$ we'll get:

$$n \geq \frac{\ln\left(\frac{2}{0.03}\right) + \ln 30}{2 \cdot 0.05^2} = 1520.18$$

And thus the minimal required n will be:

$$n = \lceil 1520.18 \rceil = 1521$$