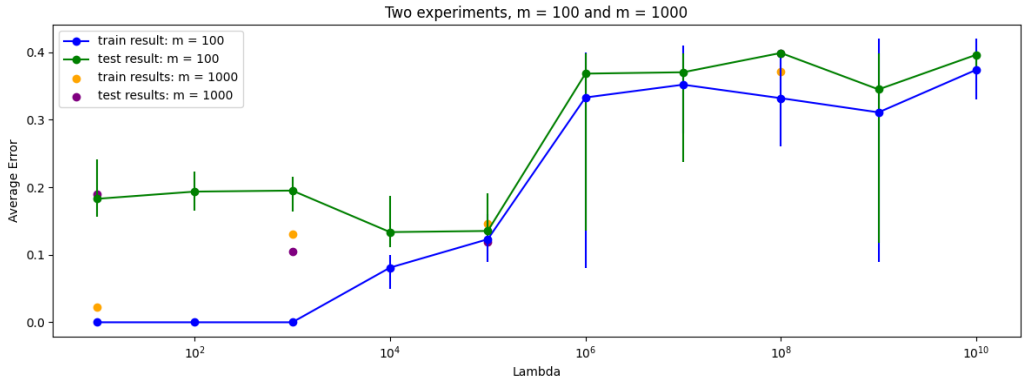
# Exercise 2

Ilana Pervoi 318271640, Omri Bar Oz 313325961

## Question 2:

a+b)




c)

- Sample Size:
  - Training error: We expect to get a smaller training error with smaller sample size (in our case m = 100).
    As the distribution might not be separable, for a smaller sample size there is a higher chance to be separable.
    For a larger size of a sample on the other hand, the sample is a better representative of the distribution, hence there's a higher chance of not be separable and it will cause us for higher error on the training sample.
  - Test error: We expect to get a smaller training error with larger sample size. As the sample size is higher, it is a better representative of the distribution and will result in lower estimation error.
  - Results: As shown in the plot above, the orange dots (higher sample size of the training set) are above the blue line (smaller sample size of the training set) as we expected. Moreover, the purple dots (higher sample size of the test set) are below the green line (smaller sample size of the test set), or close to it as we expected.

- $\lambda$ tradeoff:
  - Training error: We expect that as the $\lambda$ increases, the training error will increase as well.
    For small $\lambda$, we expect the SVM program to find a separator with small margin, which will result with smaller training error. As $\lambda$ increases, we expect SVM to find separator with smaller norm, which will result in larger margin, which will result in higher approximation error.
  - Test error: We expect that as the $\lambda$ increases, the test error will first decrease until the optimal value for $\lambda$, and then it will increase.

For small $\lambda$, we expect the SVM program to find a separator with small margin, which will result in high estimation error.

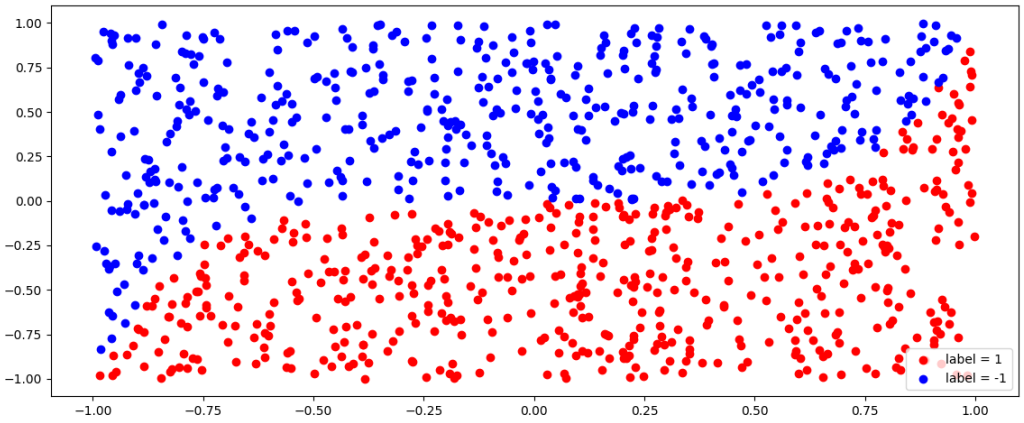
When $\lambda$ increases, we expect the test error to decrease until it meets the training error, where the $\lambda$ optimal value is, and the tradeoff between the norm of the separator and the hinge loss is optimal.

As $\lambda$ continues to grow, we put more weight on the norm of the separator, and SVM will minimize the norm of the separator. This results in higher margin, which will cause us to penalize even correct labeled examples and eventually will cause a higher test error.

- Our results: as we see in the plot above, the training error does increase as we expected, and the test error does decrease until it meets the training error, and then starts to increase.

## Question 4:

a)




we notice in the plot that there isn't a linear predictor that separates all the dots, but there is a polynomial separator that can separate all the blue and red dots more accurately.

b)

| $\lambda \setminus k$ | 2 | 5 | 8 |
|---|---|---|---|
| 1 | 0.069 | 0.059 | 0.051 |
| 10 | 0.069 | 0.064 | 0.059 |
| 100 | 0.069 | 0.064 | 0.061 |

The best result was with $\lambda = 1, k = 8$ with average error of 0.051.
After running with the chosen parameters on the entire training set, we got error of 0.03.

When running the 5-fold cross validation on the soft SVM without kernel we got:

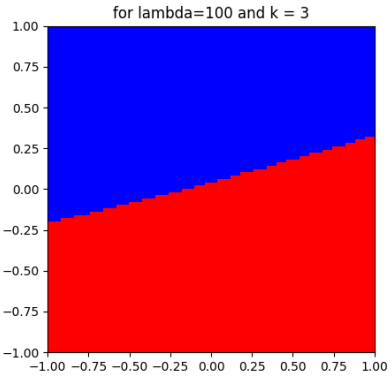
| $\boldsymbol{\lambda = 1}$ | $\boldsymbol{\lambda = 10}$ | $\boldsymbol{\lambda = 100}$ |
|---|---|---|
| 0.063 | 0.063 | 0.063 |

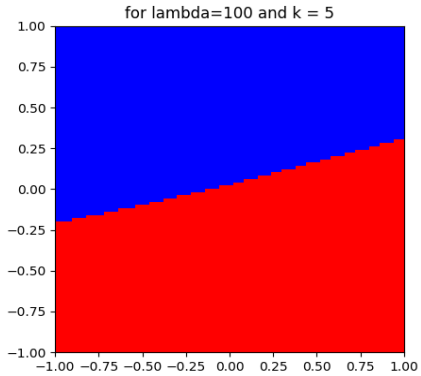
$\lambda = 1$ was chosen, and after running again with it on the entire training set we got error of 0.04.

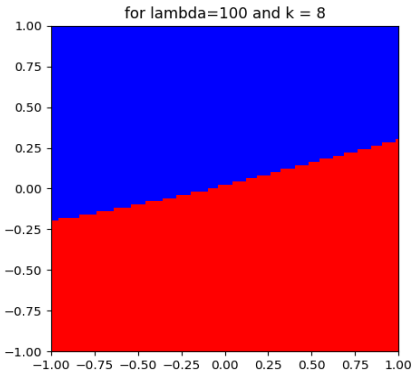
c) The polynomial kernel SVM got a better error than the linear soft SVM.
This behavior is as we expected, as we can see from the distribution of the examples in $\mathbb{R}^2$ in sub-question 4.a, there isn't a linear separator, but there is a polynomial separator.

d) For a general classification problem, if the examples in the distribution are separable by a linear separator, the linear soft SVM will result in better validation error than polynomial kernel soft SVM, as using the polynomial kernel might result in a new feature space that is not separable, which will cause it to return worse separator.
On the other hand, if the examples are not separable in the original space and are separable in the new feature space, then the polynomial kernel will return a better separator which will result in a better validation error.

e)



f) The best $\lambda$ for $k = 5$ is $\lambda = 1$.

a. To calculate $w$ we use:

$$w = \Sigma_{i=1}^{m} \alpha(i) \cdot \psi(\mathrm{x_i})$$

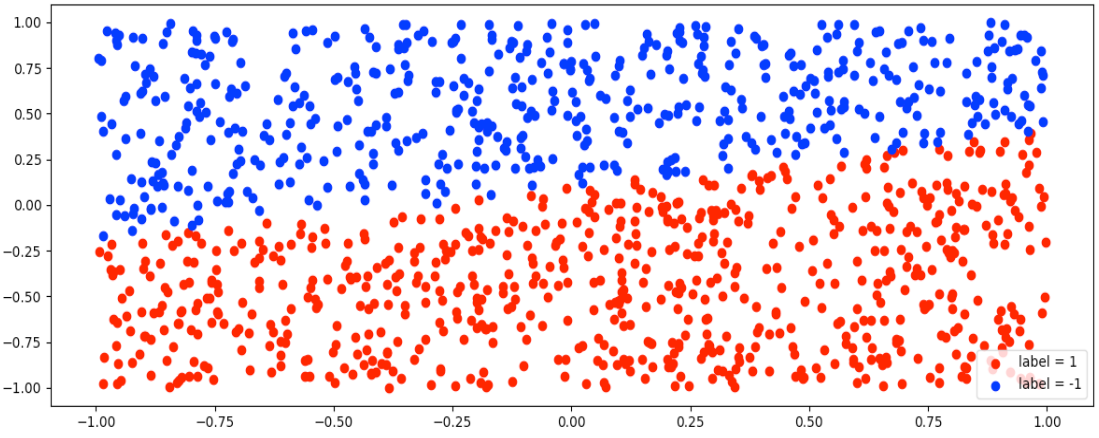
where $x_i$ are our examples.

b. $w = (0.14411052,\ 0.14411052,\ 0.10190152,\ 0.05883287,$
$0.02941644, 0.0131554, 0.14411052, -0.00428699,$
$-0.00803827, -0.00474096, -0.0023112)$

c. The multivariate polynomial that is generated is:

$$0.1441 - 1.4987x(2) - 0.0030x(2)^2 - 0.2222x(2)^3 - 0.0023x(2)^4 - 0.0243x(2)^5 + 0.4430x(1) + 0.0117x(1) \cdot x(2) - 0.0315x(1) \cdot x(2)^2 - 0.0029x(1) \cdot x(2)^3 - 0.0075x(1) \cdot x(2)^4 + (0.0326)x(1)^2 - 0.1787x(1)^2 \cdot x(2) + 0.0005x(1)^2 \cdot x(2)^2 - 0.0250x(1)^2 \cdot x(2)^3 + 0.1518x(1)^3 + 0.0077x(1)^3 \cdot x(2) - 0.0016x(1)^3 \cdot x(2)^2 + 0.0072x(1)^4 - 0.0151x(1)^4 \cdot x(2) + 0.0289x(1)^5$$

d.



## Question 5:

Let $\mathcal{X}$ be the set of all undirected graphs over $n$ vertices numbered $1, \dots, n$ with degree at most 7.
Define $rank_x(v_i)$ to be the degree of the $i'th$ node in the graph $x$.
For a graph $x \in \mathcal{X}$, define the mapping $g: \mathcal{X} \to \mathbb{N}^n$, where coordinate $i$ in the vector $g(x)$ is the degree of vertex $i$ in the graph $x$.
Let $\mathcal{H} = \{h_v: \mathcal{X} \to \mathcal{Y} \mid v \in \mathbb{N}^n, h_v \not\equiv 0\}$, where $h_v(x) = \mathbb{I}[g(x) = v]$.

**Claim 1:** *Let* $x, x' \in \mathcal{X}, x \neq x'$ and *Let* $v \in \mathbb{N}^n$. If $h_v(x) = 1$ then $h_v(x') = 0$.
**Proof:**
*Let* $x, x' \in \mathcal{X}, x \neq x'$ and *Let* $v \in \mathbb{N}^n$ *s.t* $h_v(x) = 1$.
$x \neq x'$ then exists $v_i \in \{v_1, \dots, v_n\}$ such that $rank_x(v_i) \neq rank_{x'}(v_i)$.
because $h_v(x) = 1$ then $v[i] = rank_x(v_i) \neq rank_{x'}(v_i)$ which implies $h_v(x') = 0$. ■

**Claim 2:** $|\mathcal{H}| \leq 8^n$.
**Proof:**
Let $V = \{v \in \mathbb{N}: \forall i \leq n: 0 \leq v(i) \leq 7\}$.
For each coordinate $i$ There are 8 options. There are $n$ such coordinates, which means $|V| = 8^n$. Now we will show $\forall x \in \mathcal{X}: g(x) \in V$, which implies that $|\mathcal{H}| \leq |V| = 8^n$.
Let $x \in \mathcal{X}$. Let $v_i$ be the $i$'th node in $x$.
Then $0 \leq rank_x(v_i) \leq 7$ which implies $0 \leq g(x)(i) \leq 7$ which suggests that $g(x) \in V$. ■

a) Claim 1 implies that for every $h_v \in \mathcal{H}$, there is only 1 graph in the distribution, which is labeled 1, and all other graphs are labeled 0. This means that $\mathcal{D}$ is not realizable by $\mathcal{H}$ because it can have more than 1 graph with label 1.
Then according to the PAC boundaries for the agnostic case, we get dependence of $\frac{1}{\epsilon^2}$ and it is the smallest that we can get with the PAC boundaries as $\mathcal{D}$ isn't realizable by $\mathcal{H}$.

b) Using claim 2, and the PAC-learning upper bound for agnostic settings we showed in class, and that $VC(\mathcal{H}) \leq \log_2 |\mathcal{H}|$ we get:

$$m \leq \frac{VC(\mathcal{H}) + \log\left(\frac{1}{\delta}\right)}{\epsilon^2} \leq \frac{\log_2|\mathcal{H}| + \log\left(\frac{1}{\delta}\right)}{\epsilon^2}$$
$$\leq \frac{\log_2(8^n) + \log\left(\frac{1}{\delta}\right)}{\epsilon^2} = \frac{3n + \log\left(\frac{1}{\delta}\right)}{\epsilon^2} = O(n) \quad \blacksquare$$

c) We will now prove that $VC(\mathcal{H}) = 1$.
First, for a group of size 1:
Let $v_1 = \vec{0} \in \mathbb{N}^n, v_2 = (1, 1, 0, \dots, 0) \in \mathbb{N}^n$. Let $x \in \mathcal{X}$ be the empty graph with $n$

vertices. So $h_{v_1}(x) = 1$. This implies that $h_{v_1} \in \mathcal{H}$. Also, $h_{v_2}(x) = 0$. Because exists a graph in $\mathcal{X}$ such that only the 2 first vertices are connected, it implies that $h_{v_2} \in \mathcal{H}$.
We labeled the set $\{x\}$ in all possible labels with hypothesis from $\mathcal{H}$ which implies that $VC(\mathcal{H}) \geq 1$.
Now we will show that there isn't any $h_v \in \mathcal{H}$ such that for $x_1, x_2 \in \mathcal{X}, x_1 \neq x_2$: $h_v(x_1) = h_v(x_2) = 1$. This implies that for every $x_1, x_2 \in \mathcal{X}$, the set $\{x_1, x_2\}$ can't be labeled with all possible label combinations.
This is shown immediately from Claim 1.
This implies that $VC(\mathcal{H}) < 2$ which implies that $VC(\mathcal{H}) = 1$. ■

## Question 6:

Upper Bound on Num Perceptron Updates Psuedo:
**input:** A training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$.
**output:** An upper bound on the number of updates that the Perceptron algorithm will do, or -1 if it won't terminate (S is not separable).

1. $w \leftarrow soft - SVM(S, 1)$
2. For $(x_i, y_i) \in S$:
   2.1. If $y_i < x_i, w > \leq 0$: // w is not a separator
      2.1.1. $return\ -1$
3. $R \leftarrow \max_{1 \leq i \leq n} \{||x_i||\}$
4. $w' \leftarrow hard - SVM(S)$
5. $\gamma_{w'} \leftarrow \frac{1}{R} \cdot \min_{1 \leq i \leq n} \left\{ \frac{|<w', x_i>|}{||w'||} \right\}$
6. $return \frac{1}{\gamma_{w'}^2}$

## Question 7: Quadratic program

Given the following optimization objective (modified version of soft-SVM):

$$Minimize_{w \in \mathbb{R}^d}\ \lambda \|w\|^2 + \sum_{i=1}^{m} [l^h(w, (x_i, y_i))]^2$$

a) The quadratic minimization problem with constraints that is equivalent to the problem is:

$$Minimize_{w \in \mathbb{R}^d}\ \lambda \|w\|^2 + \sum_{i=1}^{m} \xi_i^2$$

$$\forall i, \qquad y_i < w, x_i > \geq 1 - \xi_i \quad and \quad \xi_i \geq 0$$

When the auxiliary variables are the same as in the soft-SVM implementation.

b) The unknows vector is $z = (w_1, \dots, w_d, \xi_1, \dots, \xi_m)$
Hence, to solve the problem we wrote above, we will set:

$$u = \left(\underbrace{0, \dots, 0}_{m+d}\right)$$

As there is no linear component of the unknows in the objective.

$$H = 2\begin{pmatrix} \lambda I_{d\times d} & 0_{m\times d} \\ 0_{m\times d} & I_{m\times m} \end{pmatrix}$$

As the $\xi_i$'s are now a part of the quadratic component of the objective.

$$v = (\underbrace{0, \dots, 0}_{m}, \underbrace{1, \dots, 1}_{m})$$

The constants of the constraints.

$$A = \begin{pmatrix} 0_{m\times d} & I_{m\times m} \\ B_{m\times d} & I_{m\times m} \end{pmatrix}$$

Where $B_i = y_i x_i$ is a row in $B$

- Notice that the only change from the soft-SVM implementation is the optimization objective $\rightarrow u, H$.

## Question 8: The representer theorem

a) Presenting the following optimization objective:

$$Minimize_{w\in\mathbb{R}^d} \lambda\|w\|_1 + \sum_{i=1}^{m} < w, x_i >$$

in the representer theorem form, yields that we need R to be:

$$R(\|w\|_2) = \lambda\|w\|_1$$

We will show that a function such that cannot output the correct $\|w\|_1$ for all $w \in \mathbb{R}^d$.
Let $w_1 = (1, \dots, 1) \in \mathbb{R}^d$.

$$\|w_1\|_2 = \|(1, \dots, 1)\|_2 = \left(\sum_{i=1}^{d} 1^2\right)^{\frac{1}{2}} = d^{\frac{1}{2}}$$

$$\|w_1\|_1 = \|(1,\dots,1)\|_1 = \sum_{i=1}^{d} |1| = d$$

$$\rightarrow R\left(d^{\frac{1}{2}}\right) = \lambda d$$

$$\rightarrow (*)\ R(x) = \lambda x^2, \qquad \forall x \in \mathbb{R}$$

Now let's look at $w_2 = (10, 0, \dots, 0) \in \mathbb{R}^d$

$$\|w_2\|_2 = 10$$

$$\|w_2\|_1 = 10$$

$$\rightarrow\ R(\|w_2\|_2) = R(10) \underbrace{=}_{(*)} \lambda 10^2 = \lambda 100 \neq \lambda 10 = \lambda \|w_2\|_1$$

That shows that although $R$ as required in for the representer theorem works for $w_1$, it doesn't work for $w_2$. Hence, the representer theorem do not hold for the given optimization objective ■.

b) If the representer theorem does not hold for the optimization objective, we can only infer that its optimal solution cannot be expressed in the way the theorem guarantees.
In other words, the representer theorem is a one-sided theorem, hence the fact that it does not hold only means that other approach may be needed for solving the problem (no information if it is possible).

**Question 9:**

a) By definition, $K(x, x') =< \Psi(x), \Psi(x') > = < \Psi(x'), \Psi(x) >= K(x', x)$.
We will prove that the functions in a, b are not kernel functions by showing that exists $x, x'$ that for them $K(x, x') \neq K(x', x)$

$$K(x, x') \coloneqq \big(x(7) + x(3)\big) \cdot x'(1)$$

For $x = (0,0,1,0,0,0,1), x' = (1,0,0,0,0,0,0)$ we obtain:

$$K(x, x') = (1+1) \cdot 1 = 2 \neq 0 = (0+0) \cdot 0 = K(x', x)$$ ■

b) $K(x, x') \coloneqq 3 - \big(x(1) - x(2)\big) \cdot \big(x'(1) - x'(2)\big)$
By definition, $K(x, x') =< \Psi(x), \Psi(x') >$.
For $x = x'$ we get: $K(x, x) = < \Psi(x), \Psi(x) >= \|\Psi(x)\|_2 \geq 0$.
In our case, for $x = (3,0)$ we obtain:

$$K(x, x) = 3 - (3-0) \cdot (3-0) = 3 - 9 = -6 < 0$$ ■

c) $f(x, x') = \big(x(1) \cdot x'(1)\big)^4 + e^{x(3)+x(5)+x'(3)+x'(5)} + \frac{1}{x(1) \cdot x'(1)} = x(1)^4 \cdot x'(1)^4 + e^{x(3)+x(5)} \cdot e^{x'(3)+x'(5)} + x(1)^{-1} \cdot x'(1)^{-1}$

We will define $\Psi$ as follows:

$$\forall x \in \mathbb{R}^d, d > 5: \ \Psi(x) = \left(x(1)^4, e^{x(3)+x(5)}, x(1)^{-1}\right)$$

and for $x, x'$ we get:

$$f(x, x') = x(1)^4 \cdot x'(1)^4 + e^{x(3)+x(5)} \cdot e^{x'(3)+x'(5)} + x(1)^{-1} \cdot x'(1)^{-1} = \\ < \Psi(x), \Psi(x') > \ \blacksquare$$