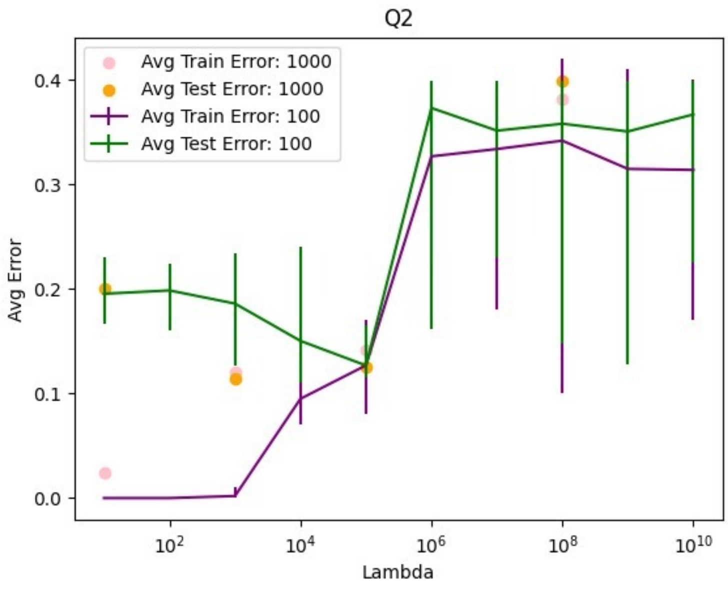
Introduction to Machine Learning
Exercise 2

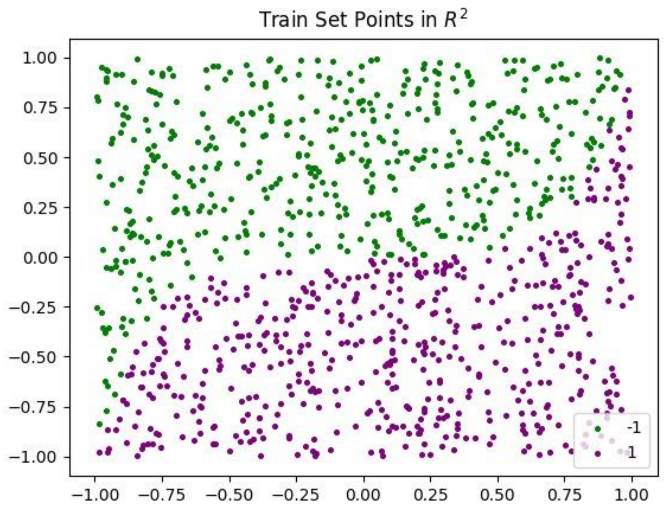
2a, 2b



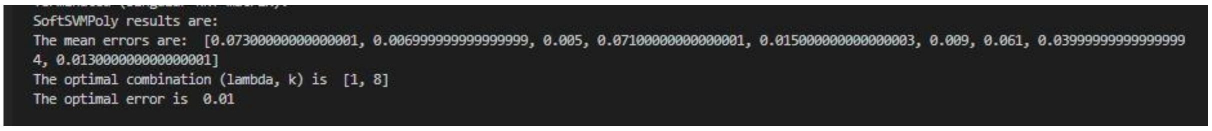
2c

We would expect that when the sample size m is bigger, the train error will be bigger. That's since it is more difficult to find a separator when there are a lot of examples.
On the other hand, the test error will be smaller. That's since as m grows, the separator will be more accurate.
The trend in the training error as a function of λ will be that the train error will grow with λ. That's since for small λ the training error won't make a lot of mistakes. The results are what we expected.
For a small λ the test error will be bigger, since there will be overfitting. That's as well what we expected.

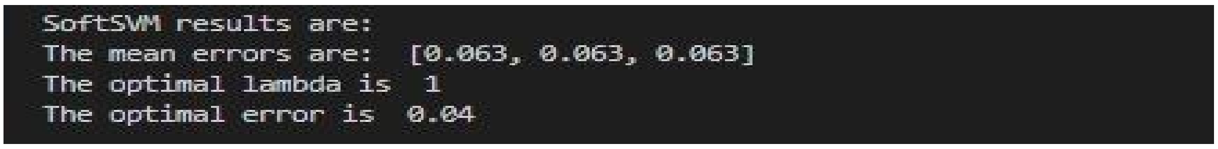
4a



It may be better to use kernel since the separator in this case won't be linear.
By using kernel we will be able to find a linear separator that will decrease the error.



```
SoftSVMPoly results are:
The mean errors are:  [0.07300000000000001, 0.006999999999999999, 0.005, 0.07100000000000001, 0.015000000000000003, 0.009, 0.061, 0.03999999999999999
4, 0.013000000000000001]
The optimal combination (lambda, k) is  [1, 8]
The optimal error is  0.01
```

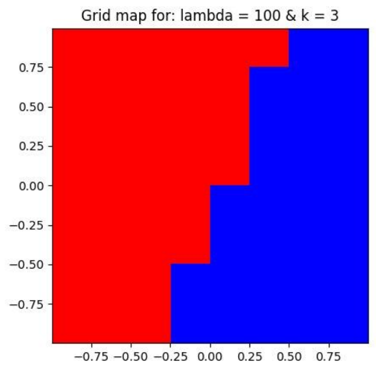


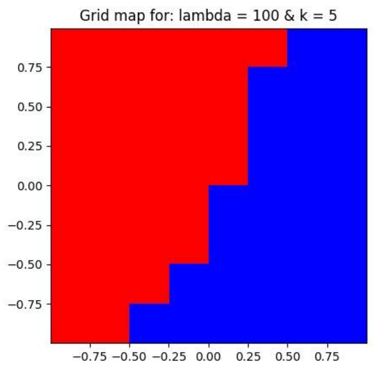
```
SoftSVM results are:
The mean errors are:  [0.063, 0.063, 0.063]
The optimal lambda is  1
The optimal error is  0.04
```

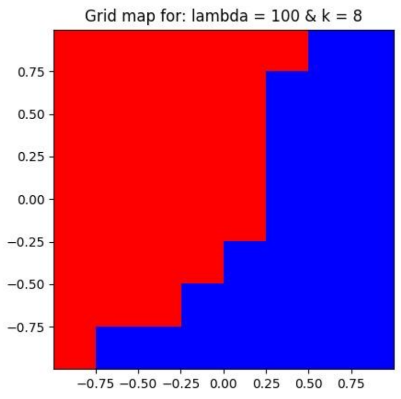
4c
The polynomial kernel achieved a better validation error, that's indeed what we expected. The reason for that is that, as we can see in the plot- the scattering isn't separable linearly. When we used the polynomial kernel we changed the dimension of the sample, which helped us achieve a linear separator.

4d
A reason why a polynomial SVM might get a better validation error than linear soft SVM is- as we mentioned in the **4c**- when we can't find a linear predictor, polynomial SVM is better since it helps us find one, which decrease the number of errors of the predictor, which decrease the validation error. On the other hand when we can find a linear separator the linear soft SVM Algorithm is better, since using the kernel will make a very small sample error- which will cause overfitting.

4e







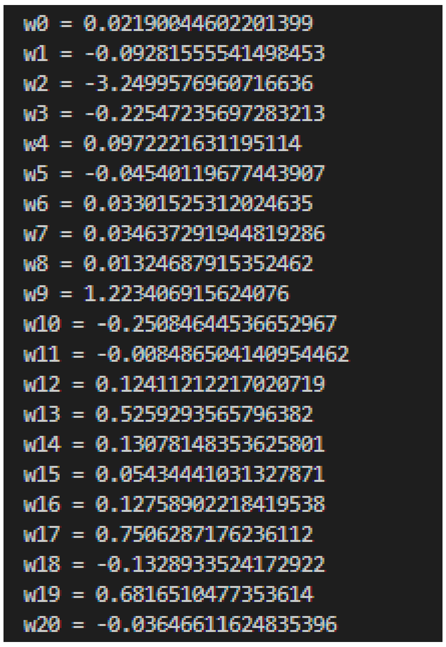
4f

i

The formula that we used to convert α to w: $w = \sum_{i=1}^{m} \alpha(i) \cdot \Psi(x_i)$, where

$\Psi(x_i)$[t] = $\sqrt{B(k,t)} \prod_{j=1}^{d} x_i[j]^{t(j)}$ for $B(k,t) = \binom{k}{t(0),...,t(d)}$ and $t \in I_d^k$

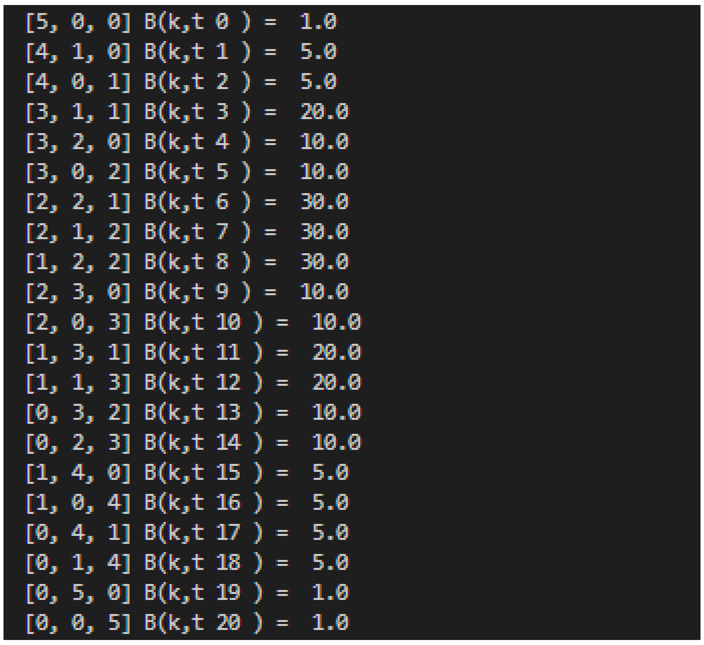
ii



```
w0 = 0.02190044602201399
w1 = -0.09281555541498453
w2 = -3.2499576960716636
w3 = -0.22547235697283213
w4 = 0.0972221631195114
w5 = -0.04540119677443907
w6 = 0.03301525312024635
w7 = 0.034637291944819286
w8 = 0.01324687915352462
w9 = 1.223406915624076
w10 = -0.25084644536652967
w11 = -0.008486504140954462
w12 = 0.12411212217020719
w13 = 0.5259293565796382
w14 = 0.13078148353625801
w15 = 0.05434441031327871
w16 = 0.12758902218419538
w17 = 0.7506287176236112
w18 = -0.1328933524172922
w19 = 0.6816510477353614
w20 = -0.03646611624835396
```

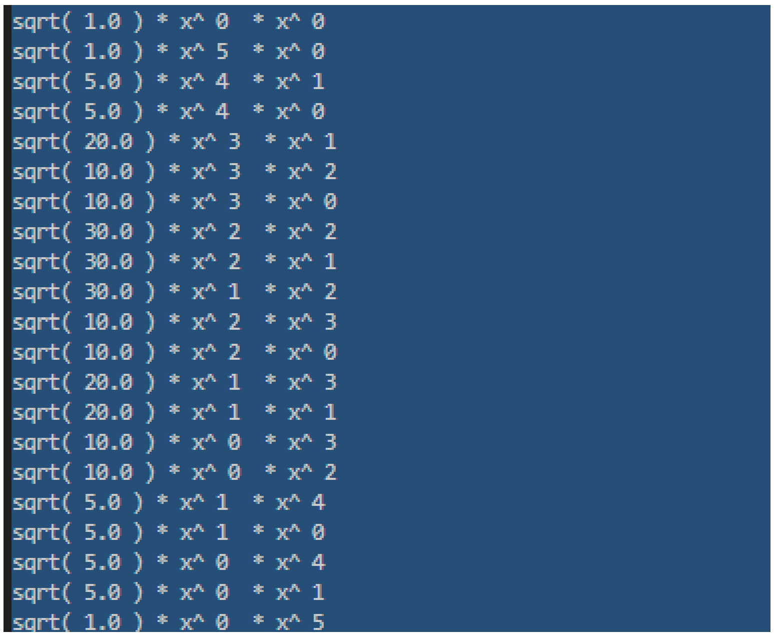
iii

This table represents: t[i] , B(k,t[i])



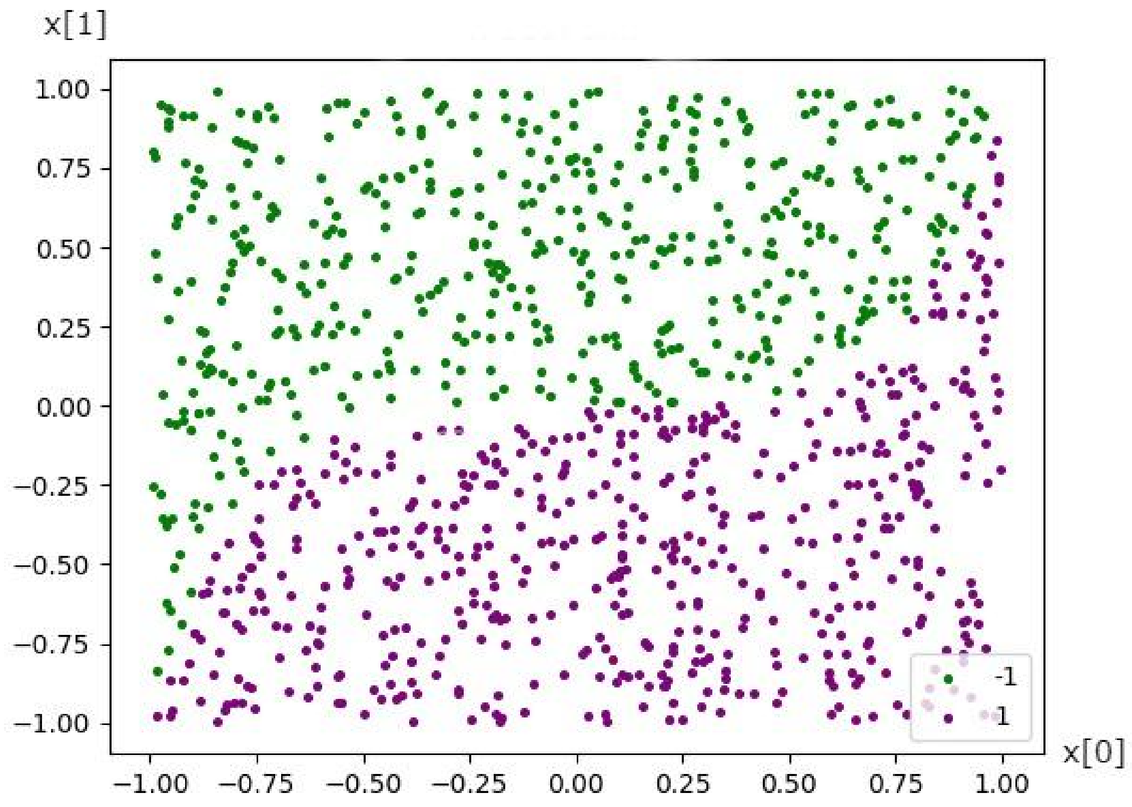
```
[5, 0, 0] B(k,t 0 ) =  1.0
[4, 1, 0] B(k,t 1 ) =  5.0
[4, 0, 1] B(k,t 2 ) =  5.0
[3, 1, 1] B(k,t 3 ) =  20.0
[3, 2, 0] B(k,t 4 ) =  10.0
[3, 0, 2] B(k,t 5 ) =  10.0
[2, 2, 1] B(k,t 6 ) =  30.0
[2, 1, 2] B(k,t 7 ) =  30.0
[1, 2, 2] B(k,t 8 ) =  30.0
[2, 3, 0] B(k,t 9 ) =  10.0
[2, 0, 3] B(k,t 10 ) =  10.0
[1, 3, 1] B(k,t 11 ) =  20.0
[1, 1, 3] B(k,t 12 ) =  20.0
[0, 3, 2] B(k,t 13 ) =  10.0
[0, 2, 3] B(k,t 14 ) =  10.0
[1, 4, 0] B(k,t 15 ) =  5.0
[1, 0, 4] B(k,t 16 ) =  5.0
[0, 4, 1] B(k,t 17 ) =  5.0
[0, 1, 4] B(k,t 18 ) =  5.0
[0, 5, 0] B(k,t 19 ) =  1.0
[0, 0, 5] B(k,t 20 ) =  1.0
```

We've written the coordinates of w in 4fii, so the vector $\Psi(x)$, for which the inner product $< w, \Psi(x) >$ happens is :



```
sqrt( 1.0 ) * x^ 0  * x^ 0
sqrt( 1.0 ) * x^ 5  * x^ 0
sqrt( 5.0 ) * x^ 4  * x^ 1
sqrt( 5.0 ) * x^ 4  * x^ 0
sqrt( 20.0 ) * x^ 3  * x^ 1
sqrt( 10.0 ) * x^ 3  * x^ 2
sqrt( 10.0 ) * x^ 3  * x^ 0
sqrt( 30.0 ) * x^ 2  * x^ 2
sqrt( 30.0 ) * x^ 2  * x^ 1
sqrt( 30.0 ) * x^ 1  * x^ 2
sqrt( 10.0 ) * x^ 2  * x^ 3
sqrt( 10.0 ) * x^ 2  * x^ 0
sqrt( 20.0 ) * x^ 1  * x^ 3
sqrt( 20.0 ) * x^ 1  * x^ 1
sqrt( 10.0 ) * x^ 0  * x^ 3
sqrt( 10.0 ) * x^ 0  * x^ 2
sqrt( 5.0 ) * x^ 1  * x^ 4
sqrt( 5.0 ) * x^ 1  * x^ 0
sqrt( 5.0 ) * x^ 0  * x^ 4
sqrt( 5.0 ) * x^ 0  * x^ 1
sqrt( 1.0 ) * x^ 0  * x^ 5
```

iiii



5a

As we learned in class- for any finite hypothesis class H, and any distribution D, the smallest dependence on ϵ is: $\epsilon \geq \frac{log(|H|) + log(\frac{2}{\delta})}{m}$

The reason for that is that this will cause the minimum error.

5b

From the data given in the question, we can tell that $|H|$ is the number of all variations edges between all n vertices (with the given restrictions). Hence, $|H|$ is exponential with n (at least), and since that: $log(|H|) \geq log(e^{n}) = n,$ and therefore: $O(H) = O(n).$

5c

The VC dimension of H is 1. That's since if $VC = 2$ we won't be able to separate between x,x' : if their v is the same, their label will be the same, and if their v isn't the same, their labels can't be the same either.

6

input: $S = ((x_1, y_1), \dots, (x_m, y_m))$

$output\text{: } l = upper\ bound\ on\ the\ number\ of\ updates\ that\ the\ Perceptron\ algorithm\ would\ require$

```
1. $run\ hard - SVM\ algorithm$
2. $if\ hard - SVM\ found\ a\ solution\ w$:          $(separable\ case)$
    2.1. $R = max\ ||x_i||$
    2.2. $for\ i\ in\ range(m)$:
        2.2.1. $\gamma_i(w) = \frac{1}{R} \cdot min\frac{|<w,x_i>|}{||w||}$
    2.3. $\gamma_s = max\{\gamma_i(w)\}$
    2.4. $return\ \frac{1}{(\gamma_s)^2}$
3. $else\ return\ -1$
```

7a

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

$$minimize_{w\in R^d}\quad \lambda||w||^2 + \sum_{i=1}^{m} [max\{0,\ 1 - y_i < sw,\ x_i >\}]^2$$

We'll define: $max\{0,\ 1 - y_i < w,\ x_i >\} = \xi_i$

so:

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

$s.t$: $\quad \forall_{1\le i\le m}\ \xi_i > 0$

$\qquad \xi_i > 1 - y_i < w,\ x_i >$

7b

$H = [\,[2\lambda * I_d]\ ,\ [0]_{d*m}$

$\qquad [0]_{m*d}\ ,\ [2\ * I_m]\,]$

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

$A = [\,[y_1 * x_1,\ ..\ ,\ y_m * x_m]\ ,\ [I_m]$

$\qquad [0]_{m*d}\ ,\ [I_m]\,]$

$v = (1,\ \dots\ ,\ 1_d,\ 0,\ \dots\ ,0_m)$

8a

In order to to show that the conditions for the representer theorem do not hold for the following minimization problem, we'll falsely assume that there exists $R: \Re \rightarrow \Re \cup \{\infty\}$ s.t. $R(\, ||x||_2) = \lambda\, ||x||_1$ for all $x \in R^d$.

We'll take w=(1,0,0,...,0) and so $||w||_2 = \sqrt{1} = 1 = ||w||_1$.

Now we'll take a vector w' which $||w'||_2 < ||w||_2$, $w' = (\frac{2}{3}, \frac{2}{3}, 0,..., 0)$ and so

$||w'||_2 = \sqrt{\frac{8}{9}}$,which indeed makes $||w'||_2 < ||w||_2$ but $||w'||_1 = \frac{4}{3}$ and $||w'||_1 > ||w||_1$.

So R is not a monotonic non-decreasing function , and that contradicts our assumption.

So This minimization problem cannot be represented by the representer theorem.

8b

No. We cannot infer anything about this minimization problem above from the fact that the representer theorem does not hold. We can't definitely know that there isn't such a vector that minimizes this problem, and actually we can try to find such a vector by the different iterable techniques we learned in class, such as Gradient descent.

Although we can definitely infer that w (the vector) can't be represented as a sum of this linear combination of the mapping of x_i .

9a

In order to show that K is not a Kernel function we'll show that a mapping function ψ: $R^d \rightarrow R^k$ s.t $K(x, x') = < \Psi(x), \Psi(x') >$ doesn't exist.

Let's falsely assume that there is such a map.

Thus, $\forall x, x' \in R^d$

$k(x, x') = (x(7) + x(3)) * x'(1) = < \Psi(x), \Psi(x') >$

So for $d = 7,\ x = x',\ x(7) = x(3) = x(1) = 0,\ x(2) \neq 0$

$0 = (x(7) + x(3)) * x(1) = < \Psi(x), \Psi(x') > = |x|^2$ so $|x|^2 = 0$

But , $x(2) \neq 0$,so x isn't the 0 vector, so $|x|^2 > 0$ which is a contradiction.

9b

Like 9a we'll falsely assume that there is such a map.

Thus, ∀x,x'∈R^d,

k(x,x')= 3 - (x(1)-x(2)) * (x'(1)-x'(2))= <ψ(x),ψ(x')>

So for d=2, x=x', x(1)=$\sqrt{3}$,x(2)=0

k(x,x') = 3 - (x(1)-x(2)) * (x(1)-x(2)) = 3 - $\sqrt{3}^2$ = 0 =

<ψ(x),ψ(x')> = <ψ(x),ψ(x)> = $|x|^2$ so $|x|^2$ = 0.

But ,x(1)≠0 ,so x isn't the 0 vector, so $|x|^2$ >0 which is a contradiction.

9c

$< \Psi(x), \Psi(x') > \ = f(x, x') \ = \ (x_1 \cdot x'_1)^4 + e^{x_3 + x_5 + x'_3 + x'_5} + \frac{1}{(x_1 \cdot x'_1)}$

We’ll define the feature map to be:

$\Psi(x) \ = \ (x_1{}^4, \ e^{x_3 + x_5}, \ \frac{1}{x_1})$

We’ll prove that: $< \Psi(x), \Psi(x') > \ = f(x, x')$

$< \Psi(x), \Psi(x') > \ = \ < (x_1{}^4, \ e^{x_3 + x_5}, \ \frac{1}{x_1}), \ (x_1{}^4, \ e^{x_3 + x_5}, \ \frac{1}{x_1}) > \ =$

$x_1{}^4 \cdot x'_1{}^4 \ + \ e^{x_3 + x_5} \ \cdot \ e^{x'_3 + x'_5} \ + \ \frac{1}{x_1} \cdot \frac{1}{x'_1} \ =$

$(x_1 \cdot x'_1)^4 + e^{x_3 + x_5 + x'_3 + x'_5} + \frac{1}{(x_1 \cdot x'_1)} \ = \ f(x, x')$