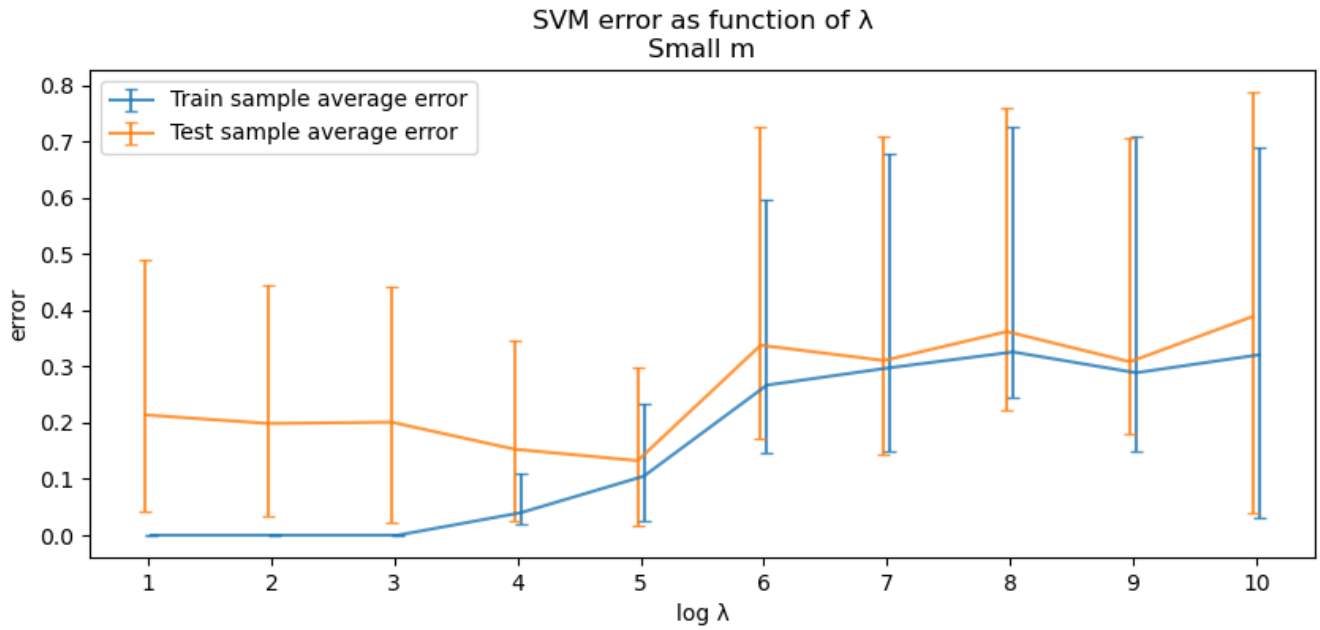
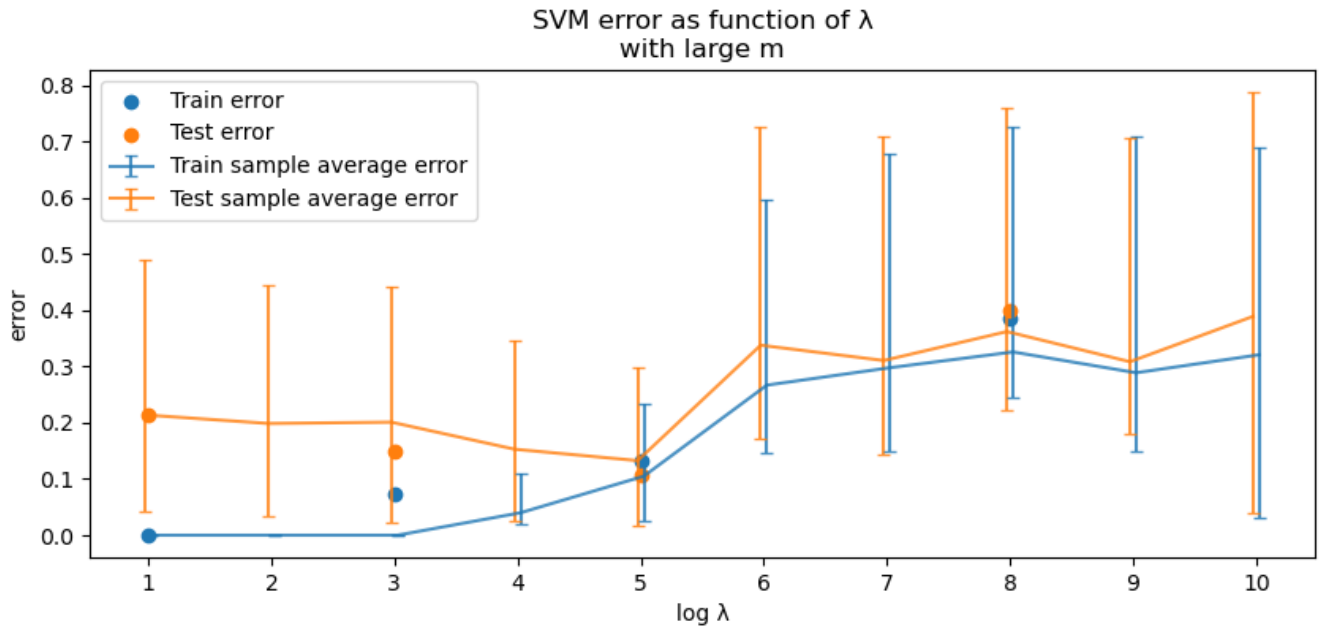
2.

(a)



(b)



(c)

Sample Size

Since we are talking about linear separator, we would expect small sample size to get lower error, as it has higher chance of being separable (if we think of m=2 it is obvious we can find linear separator but for some huge m it is expected that we will see some overlapping samples). The graph shows what we expected.

In terms of error, we expected that large sample size we get lower error sine the model can generalize better. We can see that for reasonable λ values it is what we expected.

Training Error as function of λ:

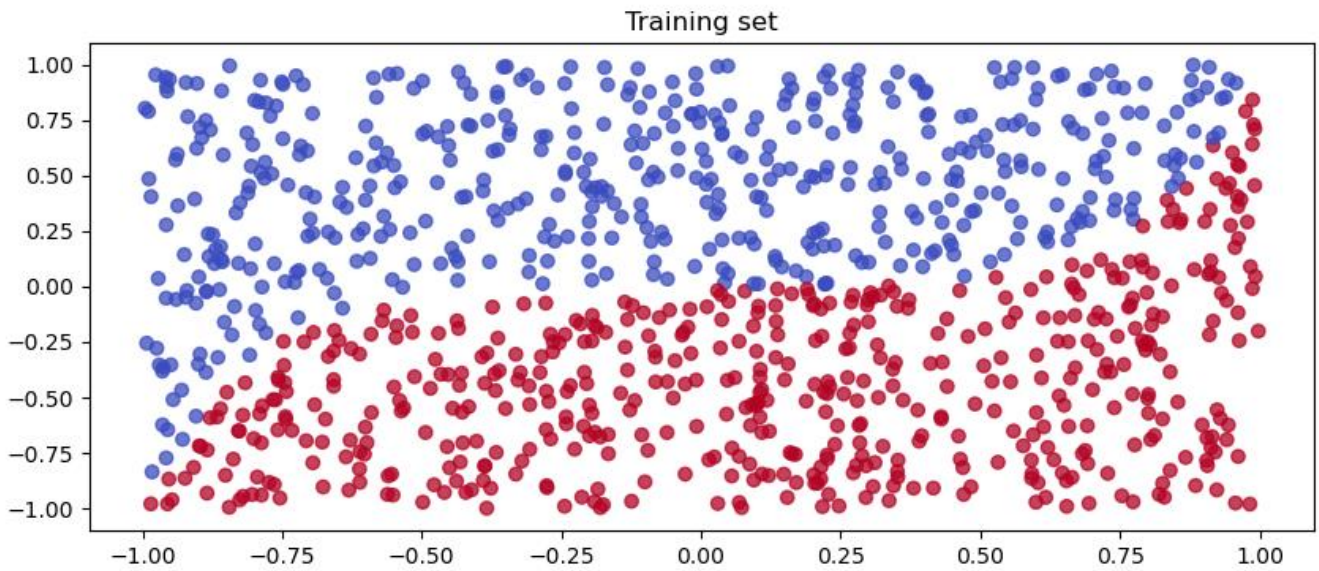
The error on train sample should be increasing function of λ since for small λ values the quadratic program will mostly try to minimize the hinge lost, which as shown in class is greater than the error. We can see the graph act as expected

Test Error as function of λ:

We expected for small values to cause overfitting and for large values we expected that the minimization will mostly be on $||w||$ and not as much on the hinge loss, which will encourage large margin but the hinge lost could be high as well, results in high error. Hence we expected somewhere near the average λ to yield the best result, which we can see is what really happened.

4.

(a)



Observing the plot, it is clear we need some kind of polynomial separator since we cant fit a line without get wrong labels on the corners, thus we would like to use kernel function that will allow for non-linear separator.

(b)

polynomial kernel errors by (λ, k) [sorted by error]:

λ, k	Error
(1.0, 8.0)	0.051000000000000004
(1.0, 5.0)	0.059
(10.0, 8.0)	0.059
(100.0, 8.0)	0.061
(10.0, 5.0)	0.064
(100.0, 5.0)	0.064
(1.0, 2.0)	0.069
(100.0, 2.0)	0.069

linear soft-SVM errors by λ:

λ	Error
1.0	0.063
10.0	0.063
100.0	0.063

We can see the best results are for the polynomial kernel achieved by $\lambda = 1, k = 8$,

And for the linear soft-SVM the results are the same for all three values of λ

When using these parameters on the entire training set we got the following errors:

For polynomial kernel soft-SVM: 0.005

For linear soft-SVM: 0.04

(c)

The polynomial kernel got better results as expected, since as can be seen by the plot in section (a), any linear separator will get large error on these samples.

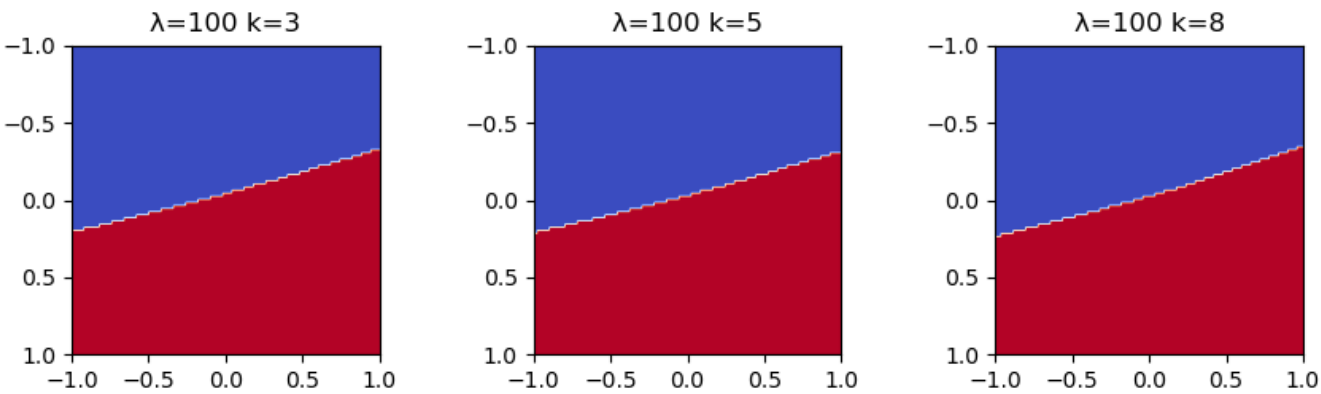
With the polynomial kernel we get polynomial separator, which can curve and thus get better results.

(d)

polynomial SVM might get a better validation error when the examples are not separable by linear separator, because it is like mapping the samples to a space where they are linearly separable (or close)

but if the samples are already separable by linear separator, the polynomial SVM can cause overfitting or map the samples to another space where they are not separable and then we might get worse results than linear SVM

(e)



(f)

As seen in the table in section (b), for $k = 5$ we will choose $\lambda = 1$.

i.

$$w = \sum_{i=1}^{m} \alpha_i \psi(x_i)$$

where α_i are the coefficients form the solution to the quadratic formula and using

$$\psi(x) = \sqrt{B(k,t)} * \Pi_{i=1}^{d} x_i^{t_i}$$

such that:

$$x = (x_1, x_2, \dots, x_d)$$
$$t \in I_d^k$$

$$B(k,t) := \binom{k}{t_0, t_1 \dots, t_d}$$

ii.

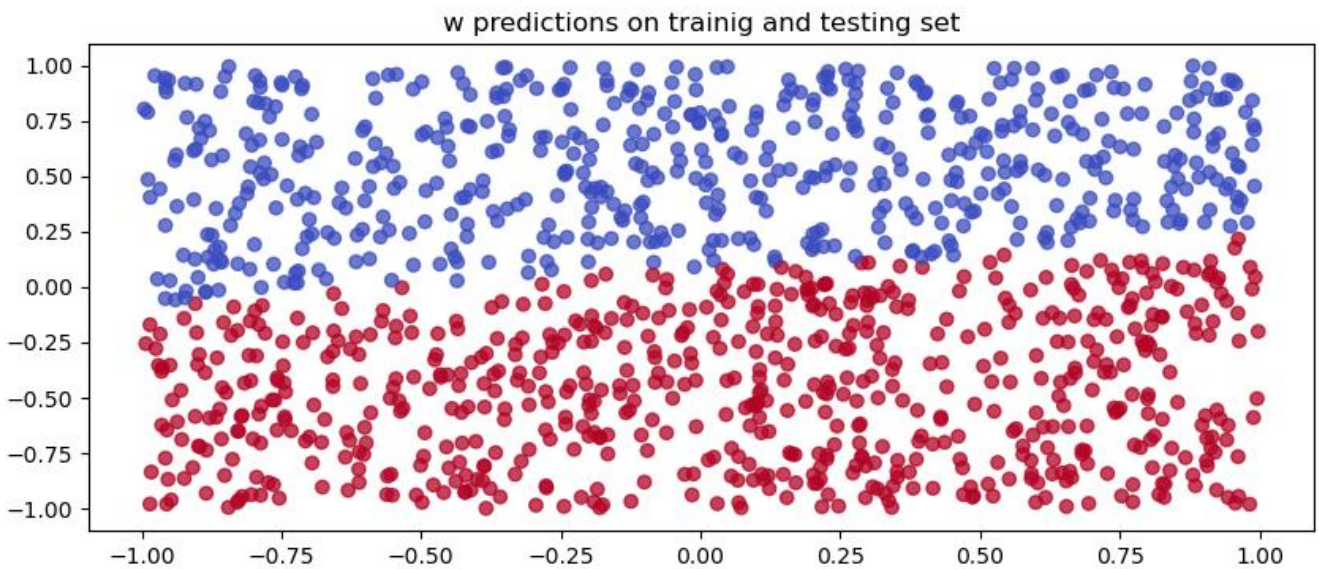
W=

(0.2618024100754538, −3.2005815118971315, −0.02239608474556537, −0.17887960507800724, −0.006582061520956273, −0.007080407152782971, 0.3812796773337209, 0.012084139189132296, −0.06418504152690917, −0.005955023408958214, 0.0014776345953271782, 0.007124242425358851, −0.12010296600830983, −0.00019049841678248192, 0.010961719231138733, 0.43232440378956916, 0.014971437680699, 0.04505020894236659, 0.009032519988696468, 0.06304429502245872, 0.1110760188787827)

iii. $\langle w, \psi(x) \rangle =$

$$0.2618024100754538 * x(1)^0 \ * x(2)^0$$
$$-3.2005815118971315 * x(1)^0 * x(2)^1$$
$$-0.02239608474556537 * x(1)^0 * x(2)^2$$
$$-0.17887960507800724 * x(1)^0 * x(2)^3$$
$$-0.006582061520956273 * x(1)^0 * x(2)^4$$
$$-0.007080407152782971 * x(1)^0 * x(2)^5$$
$$+0.3812796773337209 * x(1)^1 * x(2)^0$$
$$+0.012084139189132296 * x(1)^1 * x(2)^1$$
$$-0.06418504152690917 * x(1)^1 * x(2)^2$$
$$-0.005955023408958214 * x(1)^1 * x(2)^3$$
$$+0.0014776345953271782 * x(1)^1 * x(2)^4$$
$$+0.007124242425358851 * x(1)^2 * x(2)^0$$
$$-0.12010296600830983 * x(1)^2 * x(2)^1$$
$$-0.00019049841678248192 * x(1)^2 * x(2)^2$$
$$+0.010961719231138733 * x(1)^2 * x(2)^3$$
$$+0.43232440378956916 * x(1)^3 * x(2)^0$$
$$+0.014971437680699 * x(1)^3 * x(2)^1$$
$$+0.04505020894236659 * x(1)^3 * x(2)^2$$
$$+0.009032519988696468 * x(1)^4 * x(2)^0$$
$$+0.06304429502245872 * x(1)^4 * x(2)^1$$
$$+0.1110760188787827 * x(1)^5 * x(2)^0$$

iv.



5.

Let χ be the set of all undirected graphs over n vertices numbered 1,…,n with degree at most 7.

$Y = \{0,1\}$

$G: \chi \to \mathbb{N}^n$ as the described function

(a)

We cannot know how D is defined,

We'll show an example of D which is not realizable:

<u>Example:</u> for every graph x, $(x, y = 1) \in D \Leftrightarrow all\ vertecies\ in\ x\ degrees\ is \geq 4$

So D is a distribution over $X \times Y$. Also, the label of a graph x is deterministic function of $g(x)$,

Also, there's no $v \in N^n$ such that h_v will get error 0 on D.

Let x_1 - graph with n vertices such that every vertex in x_1 is degree 4

Let x_2 - graph with n vertices such that every vertex in x_2 is degree 4 except vertex 1 which is degree 7

So:

$g(x_1) = (\overset{n\ times}{4, \dots, 4})$

$g(x_2) = (7, \overset{n-1\ times}{4, \dots, 4})$

$if\ v = (\overset{n\ times}{4, \dots, 4})$ we'll get $h_v(x_1) = 1\ and\ h_v(x_2) = 0$ and we get an error on x_2

If $v = (7, \overset{n-1\ times}{4, \dots, 4d})$ well get $h_v(x_1) = 0\ and\ h_v(x_2) = 1$ and we get an error on x_1

For the described D there's no such v which we get an $h_v(x_1) = h_v(x_2) = 1$.

And thus we get only the agnostic PAC bound for which we know we get dependence on ϵ^2

(b)

Let us calculate the size of the hypothesis class:

$H = \{h_v: \chi \to Y | v \in N^n, h_v \neq 0\}$

Any labeled undirected graph size n can have $\binom{n}{2}$ different edges as the way to pick 2 vertices

and for any graph we can choose between assign a specific edge or not

thus we got $2^{\binom{n}{2}}$ labeled graphs with n vertices, also we should leave the graph with no edges because $v \neq 0$.

In conclusion $|H| = 2^{\binom{n}{2}} - 1$

As we showed in class the PAC-learning upper bound for the sample complexity of learning H is:

We'll describe the upper bound for both Pac cases (if D is Realizable by H and if D is not Realizable by H), for unknown D we can use only the agnostic case

For the realizable case:

$$m \geq \frac{\log(|H|) + \log\left(\frac{1}{\delta}\right)}{\epsilon} = \frac{\log\left(2^{\binom{n}{2}} - 1\right) + \log\left(\frac{1}{\delta}\right)}{\epsilon} = \frac{\mathrm{O(n)} + \log\left(\frac{1}{\delta}\right)}{\epsilon} = O(n)$$

For the agnostic case:

$$m \geq \frac{2\log(|H|) + 2\log\left(\frac{2}{\delta}\right)}{\epsilon^2} = \frac{2\log\left(2^{\binom{n}{2}} - 1\right) + 2\log\left(\frac{2}{\delta}\right)}{\epsilon^2} = \frac{2\mathrm{O(n)} + 2\log\left(\frac{2}{\delta}\right)}{\epsilon^2} = O(n)$$

(c)

For any $x_1, x_2 \in \chi$ such that $x_1 \neq x_2$

There are 2 cases: $g(x_1) = g(x_2)\ or\ g(x_1) \neq g(x_2)$

<u>$g(x_1) = g(x_2)$:</u>

we can only label $\big((x_1, 1), (x_2, 1)\big)\ or\ \big((x_1, 0), (x_2, 0)\big)$:

$g(x_1) = g(x_2)$ there's no such $v \in N^n$ such that $h_v\big(g(x_1)\big) = 0\ and\ h_v\big(g(x_2)\big) = 1$ or

$h_v\big(g(x_1)\big) = 1\ and\ h_v\big(g(x_2)\big) = 0$

<u>$g(x_1) \neq g(x_2)$:</u>

we can only label $\big((x_1, 1), (x_2, 0)\big)\ or\ \big((x_1, 0), (x_2, 1)\big)\ or\ ((x_1, 0), (x_2, 0))$ but there's no $v \in \mathbb{N}^n$ such that $h_v(x_1) = 1\ and\ h_v(x_2) = 1$ and thus we cannot shutter any group size 2 for that case,

Thus the VC dimension of H is 1

As we showed in class the better upper bound for the sample complexity of learning H depends on the VC dimension of H is:

We'll describe the upper bound for both Pac cases (if D is Realizable by H and if D is not Realizable by H), for unknown D we can use only the agnostic case

For the realizable case:

$$m \geq \frac{\mathrm{VC(H)} + \log\left(\frac{1}{\delta}\right)}{\epsilon} = \frac{1 + \log\left(\frac{1}{\delta}\right)}{\epsilon}$$

For the agnostic case:

$$m \geq \frac{\mathrm{VC(H)} + \log\left(\frac{2}{\delta}\right)}{\epsilon^2} = \frac{1 + \log\left(\frac{2}{\delta}\right)}{\epsilon^2}$$

6.

For $\gamma \coloneqq \max\{\gamma(w,S) | w \ seperates \ S\}$

We showed in class the Preceptron performs at most $\frac{1}{\gamma^2}$ updates.

Thus we should find $\frac{1}{\gamma^2}$ of the Sample S.

We also showed in class Hard-SVM algorithm with input S return $\widehat{w}$ which is a maximal-margin separator.

Now we can describe the algorithm who returns an upper bound on Preceptron number of updates:

Input: a labeled sample S of labeled examples from X × Y

Output: an upper bound on the number of updates that the Perceptron algorithm would require if it was run on this sample.

1. runs Hard-SVM on sample S
    a. if Hard-SVM gets an error while running then the algorithm returns -1
    b. else w ← Hard-SVM output
        i. R← sample with max norm between all samples
        ii. Dxmin ← $\min_{i \leq m} |< w, x_i >|$
        iii. $\gamma \leftarrow \frac{1}{R} * \frac{Dxmin}{\|w\|}$
        iv. Return $\frac{1}{\gamma^2}$

complexity:

Hard-SVM has polynomial runtime complexity,

For b.i we need to find the maximal norm in X, calculating norm is $O(d)$ and finding max is $O(m)$ so this section is $O(m * d)$

b.ii takes $O(m * d)$ because inner product of two vectors in $\mathbb{R}^d$ is $O(d)$ and get min of m items is $O(m)$

b.iii calculating norm of w: $O(d)$

b.iv $O(1)$

in conclusion, we get **polynomial runtime complexity**

7. (a)

In order to express the problem

$$\text{Minimize}_{w\in\mathbb{R}^d}\ \lambda\|w\|^2 + \Sigma_{i=1}^{m}\left[\ell^{\text{h}}\big(w,(x_i,y_i)\big)\right]^2$$

in terms of

$$\text{Minimize}_{w\in\mathbb{R}^d}\ \frac{1}{2}z^T * H * z + \langle u, z\rangle, \qquad s.t.\ Az \geq v,$$

we will define variable ξ and rewrite the problem as

$$\text{Minimize}_{w\in\mathbb{R}^d}\ \lambda\|w\|^2 + \Sigma_{i=1}^{m}\xi_i^2,$$

$$s.t\ \forall i, \xi_i \geq 0 \wedge y_i\langle w, x_i\rangle\ \geq 1 - \xi_i$$

(b)

since we only need sum of squares of ξ, we don't need to use u, we can compile it into H:

$$H = \begin{bmatrix} 2\lambda * I_d & 0 \in M_{\mathbb{R}_{d\times m}} \\ 0 \in M_{\mathbb{R}_{m\times d}} & 2 * I_m \end{bmatrix}, \qquad u \equiv 0 \in \mathbb{R}^{d+m}, \quad v = (\overbrace{0, \dots 0}^{m\ times}, \overbrace{1, \dots 1}^{m\ times})$$

$$A = \begin{bmatrix} 0 \in M_{\mathbb{R}_{m\times d}} & I_m \\ \begin{pmatrix} y_1x_1 \\ \vdots \\ y_mx_M \end{pmatrix} & I_m \end{bmatrix} \qquad z = (w_1 \dots w_d,\ \xi_1 \dots \xi_m)$$

8. (a)

Claim: $\nexists\, f: \mathbb{R}^+ \to \mathbb{R} : f(\|x\|_2) = \|x\|_1\ \forall x \in \mathbb{R}^d$

Proof: assume in contradiction that exists such function, Let R be that function.

So $R: \mathbb{R}^+ \to \mathbb{R} : R(\|x\|_2) = \|x\|_1\ \forall x \in \mathbb{R}^d$

Let $w \coloneqq (\overset{d\ times}{1, \dots, 1})$

$$\|w\|_2 = (|1|^2 + \cdots + |1|^2)^{\frac{1}{2}} = \left|\sqrt{d}\right|$$

$$\|w\|_1 = (|1| + \cdots + |1|)^{\frac{1}{1}} = d$$

We assumed $R(\|x\|_2) = \|x\|_1$, therefore $R\left(\sqrt{|d|}\right) = d \Rightarrow R(x) = x^2$

But for $w' \coloneqq (2, \overset{d-1\ times}{0, \dots, 0})$ we get

$$\|w'\|_2 = (|2|^2 + |0|^2 \dots + |0|^2)^{\frac{1}{2}} = \sqrt{|2|^2} = |2|$$

$$\|w'\|_1 = (|2| + |0| \dots + |0|)^{\frac{1}{1}} = |2|$$

And $R(\|w'\|_2) = \|w'\|_2^2 = |2|^2 = 4 \neq \|w'\|_1$.

we assumed such f existed and found 2 vectors which the assumption doesn't hold for them, thus no such f existed.

Therefore, the representer theorem does not hold for the objective given in the question.

(b)

The representer theorem guarantees that if we can represent objective in a specific form there's a solution of a form $w = \Sigma_{i=1}^{m} \alpha_i \psi(x_i)$, where $\alpha = (\alpha_1, \dots \alpha_m) \in \mathbb{R}^m$.

Since we can't represent the given objective in that form, we can only know there **might be** a solution w in the form above, but we can't **know** that such solution exists or doesn't exist.

Hence, we can't really infer anything based on the fact that the representer theorem does not hold for the given subjective.

9.(a)

To prove that $K(x,x') = (x_7 + x_3) * x_1'$ cannot be a kernel function for any feature mapping, we need to show $\nexists\ \psi: \mathbb{R}^{\mathrm{d}} \to \mathbb{R}^n$ such that $\langle \psi(x), \psi(x') \rangle = K(x,x')$.

from inner products property: $\langle u, u \rangle = \|u\|^2 > 0\ \forall\ u \not\equiv 0$. Hence:

$$\exists \text{ such } \psi \Rightarrow K(x,x) = \langle \psi(x), \psi(x) \rangle > 0\ \forall\ x \not\equiv 0$$

even though for $x \in \mathbb{R}^d$such that $x_1 < 0 \wedge\ x_3, x_7 \geq 0$ we get that

$x \not\equiv 0 \wedge K(x,x) \leq 0$ which means $\langle \psi(x), \psi(x) \rangle \leq 0$ for some $x \not\equiv 0$, in contradiction to inner product properties. Hence, no such ψ exists.

(b)

Assume in contradiction that $\exists\ \psi: \mathbb{R}^d \to \mathbb{R}^n$ such that

$$\langle \psi(x), \psi(x') \rangle = K(x,x') = 3 - (x_1 - x_2)(x'_1 - x'_2)$$

Thus, for $0 \not\equiv x = x'$: $K(x,x) = \langle \psi(x), \psi(x) \rangle > 0$

$$K(x,x) = 3 - (x_1 - x_2)^2 = 3 + 2x_1x_2 - x_1^2 - x_2^2$$

for $d = 2, x = (0,3)$ we get $K(x,x) = 3 - 9 = -6 < 0 \Rightarrow K(x,x) < 0 \wedge K(x,x) > 0$.

we assumed that $\exists\ \psi: \mathbb{R}^d \to \mathbb{R}^n$ such that $\langle \psi(x), \psi(x') \rangle = K(x,x')$ and got contradiction, hence no such ψ exists and K cant be a kernel function.

(c)

let $\psi(x) = (x_1^4,\ \exp(x_3 + x_5),\ \frac{1}{x_1}) \Rightarrow$

$$\langle \psi(x), \psi(x') \rangle = x_1^4 * x_1'^4 + \exp(x_3 + x_5) * \exp(x'_3 + x'_5) + \frac{1}{x_1} * \frac{1}{x_1'} =$$

$$(x_1x_1')^4 + e^{x_3 + x_5 + x'_3 + x'_5} + \frac{1}{x_1x_1'} = f(x,x')$$

we found some $\psi: \mathbb{R}^d \to \mathbb{R}^n$ that satisfy $\langle \psi(x), \psi(x') \rangle = f(x,x')$, hence f can be a kernel function.