1 Theoretical Part

Based on Lecture 8 and Recitation 10

1. Let $k(\mathbf{x}, \mathbf{x}')$ be a valid PSD kernel. Provide a valid PSD kernel $\tilde{k}(\mathbf{x}, \mathbf{x}')$, constructed from k, which is guaranteed to be normalized. That is, for all $\mathbf{x}$ it holds that $\tilde{k}(\mathbf{x}, \mathbf{x}) = 1$. Prove your answer.

Let k': R^d X R^d -> R be defined as

˜k(x, x') = k(x, x') / sqrt(k(x, x) * k(x', x'))

First, we want to demonstrate that ˜k(x, x') is a valid positive semidefinite kernel. We can observe that ˜k(x, x') is symmetric, we need to show that it is PSD and not negative and normalized.
According to Mercer's theorem, because k(x, x') is a valid kernel, we can express it in a specific form:

K(x,x') = (Φ(x)(transpose)) Φ(x')

Therefore we can conclude:

˜k(x, x') = k(x, x') / sqrt(k(x, x) * k(x', x')) =

((Φ(x)(transpose)) Φ(x'))/ (sqrt((Φ(x)(transpose)) Φ(x))* sqrt((Φ(x')(transpose)) Φ(x'))) =

((Φ(x)(transpose)) Φ(x'))/(norma(Φ(x))* norma(Φ(x'))) = ((˜ Φ(x)) transpose) * (˜ Φ(x'))

Where ˜ Φ(x) = Φ(x)/norma(Φ(x)).

And now we showed that ˜k is normalized and also PSD.

2. Consider a data set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$, and a feature map $\psi : \mathbb{R}^d \to \mathcal{F}$ where $\mathcal{F}$ is some feature space. Give an example of a data set S and a feature map ψ such that S is not linearly separable in $\mathbb{R}^d$ (for $d \geq 2$) but that the transformed data set $S_\psi = \{(\psi(\mathbf{x}_i), y_i)\}_{i=1}^m$ is linearly separable in $\mathcal{F}$.

Given the dataset observed during the recitation, specifically involving two co-centric rings with varying radii, by the following mapping function we can transform and achieved into a linearly separable dataset in R^d: ψ(x1, x2) = (x1, x2, x1^2 + x2^2)

3. $k_1(\mathbf{x}, \mathbf{y})$ and $k_2(\mathbf{x}, \mathbf{y})$ are valid kernels, then:

$$k_\times(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y}) \cdot k_2(\mathbf{x}, \mathbf{y})$$

is also a valid kernel. To prove this we'll use the fact that valid kernels are positive semi-definite.
You may find the following identities helpful (but don't have to use them):

$$\mathbf{x}^\top A\mathbf{y} = Tr\left[\mathbf{x}^\top A\mathbf{y}\right] = Tr\left[\mathbf{y}\mathbf{x}^\top A\right] \tag{1}$$

$$Tr[AB] = \sum_i [AB]_{ii} = \sum_i \sum_j A_{ij} B_{ji} \tag{2}$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors while A and B are matrices.

To prove that the kernel k×(x, y) = k1(x, y) · k2(x, y) is also a valid kernel, we need to show that it satisfies the positive semidefinite property.
Let K1 and K2 be the kernel matrices corresponding to k1(x, y) and k2(x, y), respectively. The kernel matrix K for k×(x, y) is given by K = K1 · K2.
To demonstrate that K is a valid kernel, we must show that for any vector x, the quadratic form x'Kx is non-negative.
Considering x'Kx: x'Kx = x'(K1 · K2)x
Using the provided identities, we can simplify this expression:
x'Kx = Trace[x'(K1 · K2)x] = ∑[x'(K1 · K2)x]_ii
Expanding K1 · K2: x'Kx = ∑[x'K1K2x]_ii
Since K1 and K2 are valid kernel matrices, they can be expressed as K1 = A1'A1 and K2 = A2'A2, where A1 and A2 are matrices.

Substituting these expressions, we get: x'Kx = ∑[x'(A1'A1 · A2'A2)x]_ii
Let's define a new matrix A = A1A2. Since A1 and A2 are positive semidefinite, A = A1A2 is also positive semidefinite.
We can rewrite the expression as: x'Kx = ∑[x'A'Ax]_ii
Since A is positive semidefinite, the quadratic form x'A'Ax is non-negative for any vector x.
Therefore, we have shown that x'Kx is non-negative, proving that the kernel k×(x, y) = k1(x, y) · k2(x, y) is a valid kernel.

3. $k_1(\mathbf{x},\mathbf{y})$ and $k_2(\mathbf{x},\mathbf{y})$ are valid kernels, then:

$$k_\times(\mathbf{x},\mathbf{y}) = k_1(\mathbf{x},\mathbf{y}) \cdot k_2(\mathbf{x},\mathbf{y})$$

is also a valid kernel. To prove this we'll use the fact that valid kernels are positive semi-definite.
You may find the following identities helpful (but don't have to use them):

$$\mathbf{x}^\top A\mathbf{y} = Tr\left[\mathbf{x}^\top A\mathbf{y}\right] = Tr\left[\mathbf{y}\mathbf{x}^\top A\right] \tag{1}$$

$$Tr[AB] = \sum_i [AB]_{ii} = \sum_i \sum_j A_{ij}B_{ji} \tag{2}$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors while A and B are matrices.

(a) Let $k(\mathbf{x},\mathbf{y})$ be a valid kernel and suppose that K is the kernel's Gram matrix over some finite set of points $\{\mathbf{x}_i\}_{i=1}^N$, such that $K_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$. Show that for any finite set of N points, there exists some function $f : \mathcal{X} \mapsto \mathbb{R}^N$ such that:

$$k(\mathbf{x}_i,\mathbf{x}_j) = f^\top(\mathbf{x}_i) f(\mathbf{x}_j) \tag{3}$$

where $\mathcal{X}$ is space of the points $\mathbf{x}_i$. Using this fact, show that:

$$k_1(\mathbf{x},\mathbf{y}) \cdot k_2(\mathbf{x},\mathbf{y}) = \sum_i \sum_j g_i(\mathbf{x}) f_j(\mathbf{x}) f_j(\mathbf{y}) g_i(\mathbf{y}) \tag{4}$$

where $k_1(\cdot,\cdot)$ and $k_2(\cdot,\cdot)$ are valid kernels, and some functions $f, g : \mathcal{X} \mapsto \mathbb{R}^N$, where $f_i(\mathbf{x})$ denotes the i^{th} index of the output of $f(\mathbf{x})$.

(b) Conclude that $k_\times(\mathbf{x},\mathbf{y}) = k_1(\mathbf{x},\mathbf{y}) \cdot k_2(\mathbf{x},\mathbf{y}) = h^\top(\mathbf{x}) \cdot h(\mathbf{y})$ for some function $h(\cdot)$, thereby proving that $k_\times(\cdot,\cdot)$ is a valid kernel.

3.3.a

Given a valid kernel k(x, y) and its Gram matrix K computed from a finite set of N points {xi}, we want to show that there exists a function f : X → R^N such that k(xi, xj) = (f^T(xi))f(xj), where X represents the space of the points xi.

By using the eigendecomposition of the Gram matrix K as K = VSV^T, where V is the matrix of eigenvectors and S is the diagonal matrix of eigenvalues, we can define f(x) = [sqrt(λ1)v1(x), sqrt(λ2)v2(x), ..., sqrt(λN)vN(x)], where v_i(x) represents the i-th eigenvector associated with the i-th eigenvalue λ_i.

Calculating (f^T(xi))f(xj) simplifies to sqrt(λ1)v1(xi)^T v1(xj) + sqrt(λ2)v2(xi)^T v2(xj) + ... + sqrt(λN)vN(xi)^T vN(xj). Since the eigenvectors are orthogonal, v_i^T(xi) v_j(xj) equals 1 if i = j, and 0 if i ≠ j.

Thus, we can simplify (f^T(xi))f(xj) to sqrt(λ1)K_1j + sqrt(λ2)K_2j + ... + sqrt(λN)K_Nj, where K_ij represents the entries of the Gram matrix K.

Therefore, by defining f(x) as mentioned, we have shown that for any finite set of N points, there exists a function f : X → R^N such that k(xi, xj) = (f^T(xi))f(xj), where X represents the space of the points xi.

3.4.a

To show that k1(x, y) · k2(x, y) can be written as ∑∑gi(x)fj(x)fj(y)gi(y), where k1(·,·) and k2(·,·) are valid kernels and f, g: X → R^N are functions, we will use the result that states for any valid kernel k(x, y), there exist functions f: X → R^N and g: X → R^N such that k(x, y) = (f^T(x))g(y).
Let's consider the expression k1(x, y) · k2(x, y). Using the functions f and g, we can rewrite it as (f^T(x))g(y) multiplied by itself.
Expanding this expression, we have (f^T(x))^2g(y)^2. Since (f^T(x))^2 is a scalar value, we can bring it out of the summation.
Next, we can express g(y)^2 as a matrix G = g(y) g(y)^T, where G is an N × N symmetric matrix.
Now, we can rewrite the expression as (f^T(x))^2 G.
Expanding the expression (f^T(x))^2 G, we obtain ∑∑ gi(x)fj(x)fj(y)gi(y), where gi(x) represents the i-th component of the output of g(x), and fj(x) represents the j-th component of the output of f(x).
Therefore, we have shown that k1(x, y) · k2(x, y) can be represented as ∑∑gi(x)fj(x)fj(y)gi(y),

.b

By our previous Q, we have shown that k1(x, y)·k2(x, y) can be expressed as:
k1(x, y) · k2(x, y) = ∑∑ g_i(x) · f_j(x) · f_j(y) · g_i(y),
where i represents the index running on the first sigma and j represents the index running on the second sigma.

Now, let's define a new function h(x) as: h(x) = [g1(x)·f1(x), g2(x)·f2(x), ..., gi(x) · fj(x), ...].
Considering the inner product of the transpose of h(x) and h(y), we find:
(h(x))^T · h(y) = [g1(x) · f1(x), g2(x) · f2(x), ..., gi(x) · fj(x), ...]^T · [g1(y) · f1(y), g2(y) · f2(y), ..., gi(y) · fj(y), ...]
Expanding the inner product, we have: (h(x))^T · h(y) = ∑∑ (g_i(x)·f_j(x))· (g_i(y) · f_j(y)) which is equivalent to the expression we obtained earlier for k1(x, y) · k2(x, y).
Therefore, we have shown that k1(x, y) · k2(x, y) can be expressed as the inner product of the transpose of h(x) and h(y), where h(x) is defined as h(x) = [g1(x) · f1(x), g2(x) · f2(x), ..., gi(x) · fj(x), ...].

1.2 PCA

Based on Lecture 9 and Recitation 11

4. Let $X:\Omega \to \mathbb{R}^d$ be a random variable with zero mean and covariance $\Sigma \in \mathbb{R}^{d\times d}$. Show that for any $\mathbf{v} \in \mathbb{R}^d$, where $\|\mathbf{v}\|_2 = 1$, the variance of $\langle \mathbf{v}, X\rangle$ is not larger then variance obtained by the PCA embedding of X into a one-dimension subspace (assume that the PCA uses the actual Σ).

Let v ∈ ℝ^d such that ‖v‖^2 = 1 and denote X˜ ≡ ⟨v,X⟩. We want to calculate the expectation and variance of X˜.

Expectation: E[X˜] = ⟨v, E[X]⟩ = 0, since the projection of E[X] onto v is zero.

Variance: Var[X˜] = E[X˜^2] = E[(⟨v,X⟩)^2] = E[⟨v,X⟩ · ⟨v,X⟩] = ⟨v,X⟩ · ⟨v,X⟩ = (v^T Σ v) (using covariance matrix Σ) ≤ (u_1^T Σ u_1) (as v is a unit vector, and u_1 is the leading eigenvector of Σ) = λ_1 (since λ_1 is the largest eigenvalue of Σ)

At the end, the expectation of X˜ is E[X˜] = 0, and the variance of X˜ is Var[X˜] ≤ λ_1, where λ_1 is the largest eigenvalue of the covariance matrix Σ.

1.3 Convex optimization

Based on Lecture 11 and Recitations 2,12

1. Let $f_1,\dots,f_m : C \to \mathbb{R}$ be a set of convex functions and $\gamma_1,\dots,\gamma_m \in \mathbb{R}_+$. Prove from definition that $g(\mathbf{u}) = \sum_{i=1}^m \gamma_i f_i(\mathbf{u})$ is a convex function.
2. Give a counterexample for the following claim: Given two functions $f,g:\mathbb{R}\to\mathbb{R}$, define a new function $h:\mathbb{R}\to\mathbb{R}$ by $h = f\circ g$. If f and g are convex then h is convex as well.
3. Let $f: C\to\mathbb{R}$ be a function defined over a convex set C. Prove that f is convex iff its *epigraph* is a convex set, where $\mathrm{epi(f)} = \{(u,t) : f(u) \le t\}$.
4. Let $f_i : V \to \mathbb{R}$, $i \in I$. Let $f: V\to\mathbb{R}$ given by

$$f(u) = \sup_{i\in I} f_i(u).$$

If f_i are convex for every $i \in I$, then f is also convex.

1.3.1

We know that for each I belong to [m] Fi is a convex dunction.

By definition we get a belong [0,1]

aF(u) + (1-a)F(v) >= F(au + (1-a)v)   to each u,v belongs to C

multiply positive constant q we get:

a*q*F(u) + (1-a)*q*F(v) >= q*F(au +(1-a)v)

so we conclude that for each I Gi = qiFi(u) is a convex function

know we want to conclude it on the sum

we notice that: (the index on each sigma is I from 1 to m) ΣaGi(u) + (1-a)Gi(v) = Σ[aGi(u) + (1-a)Gi(v)] >= ΣGi(au+(1-a)v).

1.3.2

Counterexample: Let's consider a counterexample to the claim that if f and g are convex functions, then $h = f \circ g$ is also convex.
Take $f(x) = x^2$ and $g(x) = |x|$. Both f and g are convex functions individually, but let's examine the composition $h = f \circ g$.
$h(x) = f(g(x)) = f(|x|) = |x|^2 = x^2$.
The function $h(x) = x^2$ is not convex. To see this, we can examine the second derivative of h(x): $h''(x) = 2$,
which is a constant. Since the second derivative is positive (non-negative) everywhere, h(x) does not satisfy the definition of convexity.
Therefore, we have provided a counterexample where f and g are convex functions, but the composition $h = f \circ g$ is not convex.

1.3.3

Assume f is convex. We need to prove that its epigraph,epi(f), is a convex set.
Assume (u_1,t_1) and (u_2,t_2) are two points in epi(f). We want to show that for any λ between 0 and 1, the point $(\lambda u_1 + (1-\lambda)u_2, \lambda t_1 + (1-\lambda)t_2)$ is also in epi(f). Since (u_1, t_1) is in epi(f), we have $f(u_1) \le t_1$. Similarly, (u_2, t_2) being in epi(f) implies $f(u_2) \le t_2$.
Using the convexity of f, we have: $f(\lambda u_1 + (1-\lambda)u_2) \le \lambda f(u_1) + (1-\lambda)f(u_2)$ (convexity property) $\le \lambda t_1 + (1-\lambda)t_2$ (since $f(u_1) \le t_1$ and $f(u_2) \le t_2$)
Therefore, $(\lambda u_1 + (1-\lambda)u_2, \lambda t_1 + (1-\lambda)t_2)$ is in epi(f), and this shows that epi(f) is a convex set.
Now, let's prove the other direction. Assume epi(f) is a convex set, and we want to show that f is a convex function.
Consider any two points u_1 and u_2 in C, and let λ be a scalar between 0 and 1. We need to show that $f(\lambda u_1 + (1-\lambda)u_2) \le \lambda f(u_1) + (1-\lambda)f(u_2)$.
To do this, we consider the points $(u_1, f(u_1))$ and $(u_2, f(u_2))$ in the epigraph of f. Since epi(f) is convex, the point $(\lambda u_1 + (1-\lambda)u_2, \lambda f(u_1) + (1-\lambda)f(u_2))$ must also be in epi(f).
This implies that $f(\lambda u_1 + (1-\lambda)u_2) \le \lambda f(u_1) + (1-\lambda)f(u_2)$, as desired.
Therefore, we have shown that f is convex if and only if its epigraph, epi(f), is a convex set.

1.3.4

We will claim that the set Fi (when I belongs to I) is blocked above from the previeos Q we know that function is convex if and only if epigraph is a convex set and also epi(F) = {(u,t)|F(u)<t} therefore to every I belongs to I: Fi(u) <= t therefore we get:

epi(F) = (to all I belongs to I) Insertion(Fi) and because the insertion of convex sets define convex set we conclude that F is a convex function.

1.4 Sub-gradients for Soft-SVM Objective

Based on Lecture 11 and Recitations 2,12

The Soft-SVM objective, though convex, is not differentiable in all of it's domain due to the use of the hinge-loss. Therefore, to implement a sub-gradient descent solver for this problem we must first describe sub-gradients of the objective.

5. Given $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{\pm 1\}$. Show that the hinge loss is convex in $\mathbf{w}, b$. That is, define

$$f(\mathbf{w}, b) := \ell^{hinge}_{\mathbf{x},y}(\mathbf{w}, b) = \max\left(0, 1 - y(\mathbf{x}^\top \mathbf{w} + b)\right)$$

and show that f is convex in $\mathbf{w}, b$.

6. Deduce some sub-gradient of the hinge loss function $g \in \partial \ell^{hinge}_{\mathbf{x},y}(\mathbf{w}, b)$.
7. Let $f_1, \ldots, f_m : \mathbb{R}^d \to \mathbb{R}$ be a set of convex functions and $\mathbf{g}_k \in \partial f_k(\mathbf{x})$ for all $k \in [m]$ be sub-gradients of these functions. Define $f : \mathbb{R}^d \to \mathbb{R}$ by $f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x})$. Show that $\sum_k \mathbf{g}_k \in \partial \sum_k f_k(\mathbf{x})$.
8. Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \subseteq \mathbb{R}^d \times \{\pm 1\}$ be a sample and define $f : \mathbb{R}^d \to \mathbb{R}$ by:

$$f(\mathbf{w}, b) = \frac{1}{m}\sum_{i=1}^m \ell^{hinge}_{\mathbf{x}_i,y_i}(\mathbf{w}, b) + \frac{\lambda}{2}||\mathbf{w}||^2$$

Find a sub-gradient of f for any $\mathbf{w}$.

1.4.5

To prove that a function f: C → R is convex if and only if its epigraph, epi(f), is a convex set, we can observe that f is a maximum function between two linear functions in w. We need to establish two proofs:

i. A linear function is convex.

ii. The maximum of two convex functions is convex.

Two things we already showed

By demonstrating these two properties, we can conclude that if f is convex, then epi(f) is convex, and vice versa.

1.4.6

```
|          {                  hinge
|          {    (-yx,-y)         l (w,b) != 0
|          {                     x,y
|          {
|   G =    {                     hinge
|          {      0              l (w,b) = 0
|          {                     x,y
|
|
```

1.4.7

By definition of sub-gradient we get that for all i belong to [m], y belong to R^d:

Fi(y) >= Fi(x) + <Gi,y-x>

Therefore if we sum m inequality of those we get(i is the index wich run on every sigma at this line from 1 to m) ΣFi(y) >= ΣFi(x) + <Gi,yx>

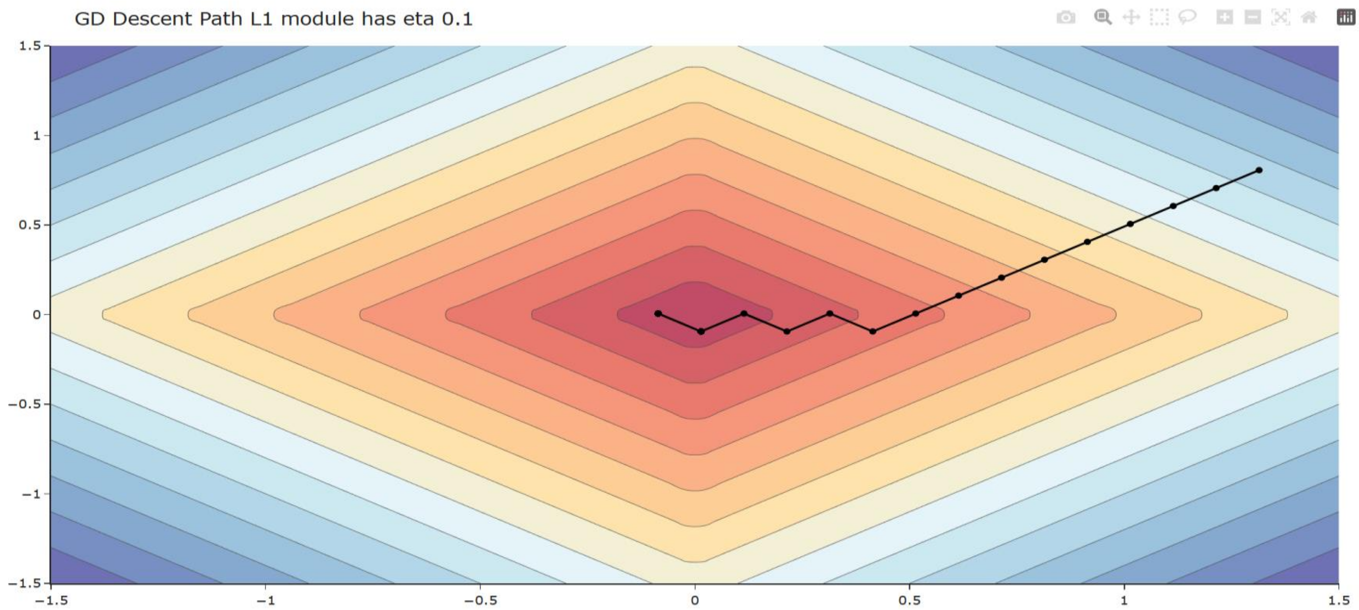
And from the definition of sub-gradient we get that (k is the indes which run on each sigma this line) Σ Gk belong ∂ Σ Fk(x) as was needed

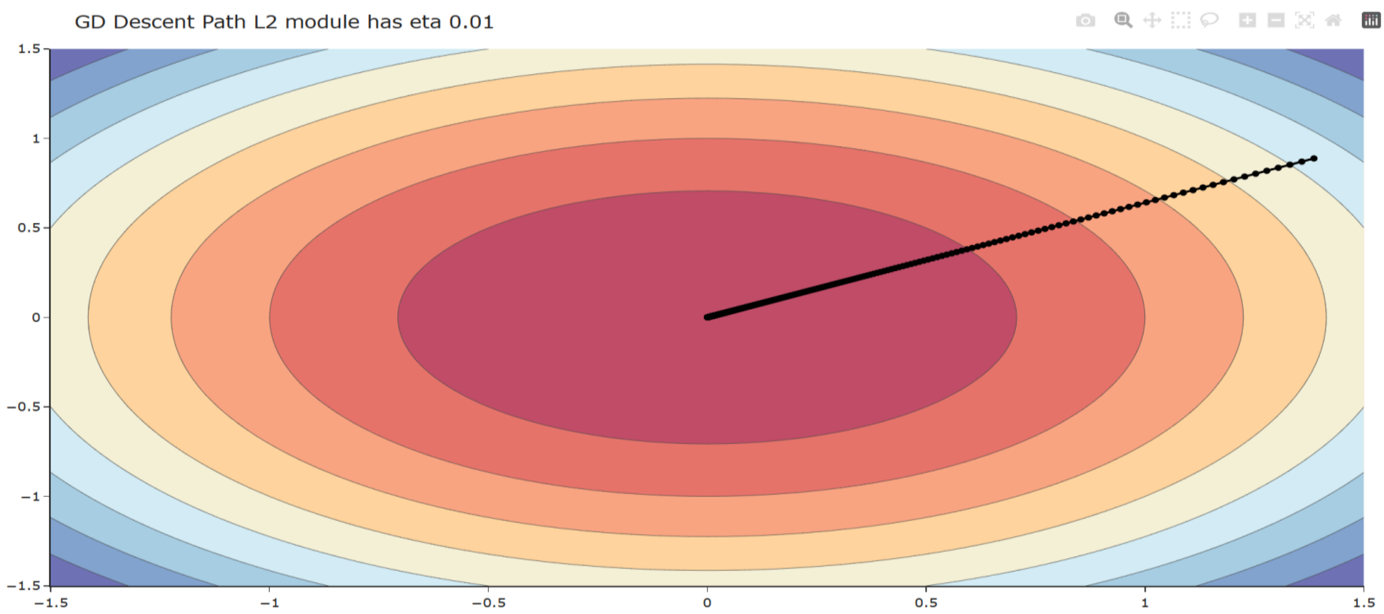
1.4.8

We define: { hinge
| { (-yx,-y) l (w,b) != 0
| { x,y
| {
| G = { hinge
| { 0 l (w,b) = 0
| { x,y
| (the index at each Σ gonna be I which run from 1 to m)
| hinge
| and we get that: (1/m) ΣGi+ λ(w,0) belong to ∂(1/m) Σ l (w,b) + (λ/2)*||w||^2
| Xi,Yi

PARTICAL PART

1. Plot the descent path for each of the settings described above (you can use the `plot_descent_path`). Add below the plots for $\eta = 0.01$ and explain the differences seen between the L1 and L2 modules.





כפי שלמדנו נורמה 1 יותר מאפסת פיצרים לעומת נורמה 2 שמאפסת פיצרים. ניתן לראות שבגרף של אל 1 הפונקציה מתאפסת כלומר מגיע לאחד הצירים כי חיפשנו שפיץ שיאפס הרבה דבליו לעומת אל 2 שבה נחפש את נקודת המפגש עבור לוס מינימלי לכן אל 2 מקווצת.

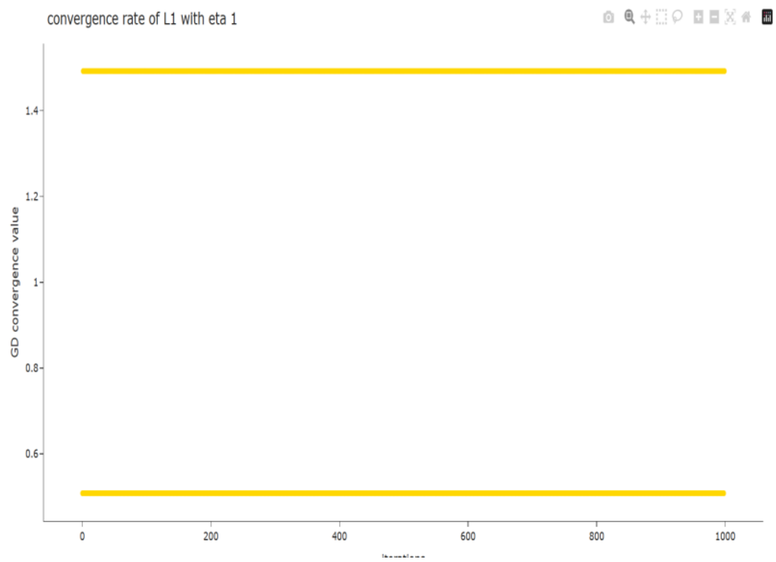
2. Describe two phenomena that can be seen in the descent path of the ℓ_1 objective when using GD and a fixed learning rate.

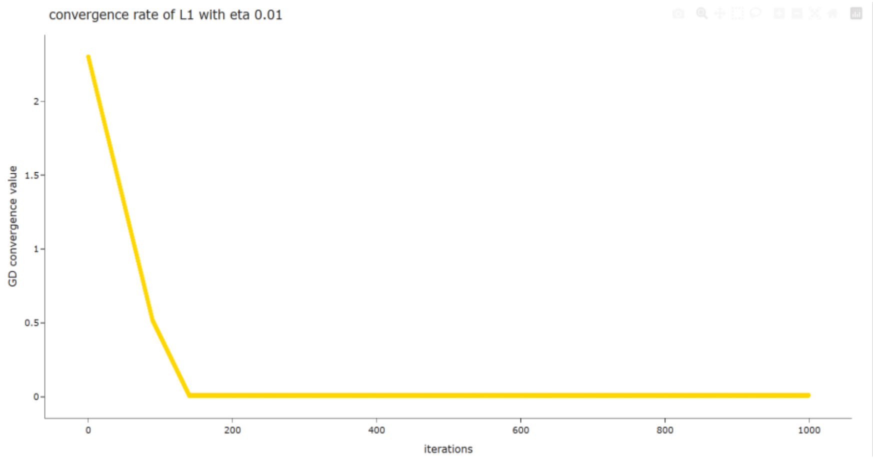
ראשית ניתן להבחין כי אל1 מנסה לאפס, כלומר להגיע לציר האיקס כמה שיותר מהר

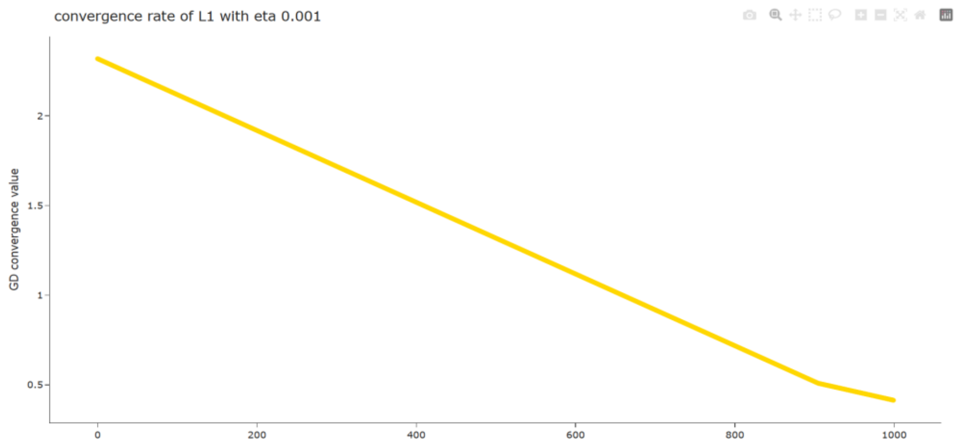
בנוסף לאחר ההגעה לציר האיקס נבחין בקפיצת של זיגזגים סביבו מעל ומתחת

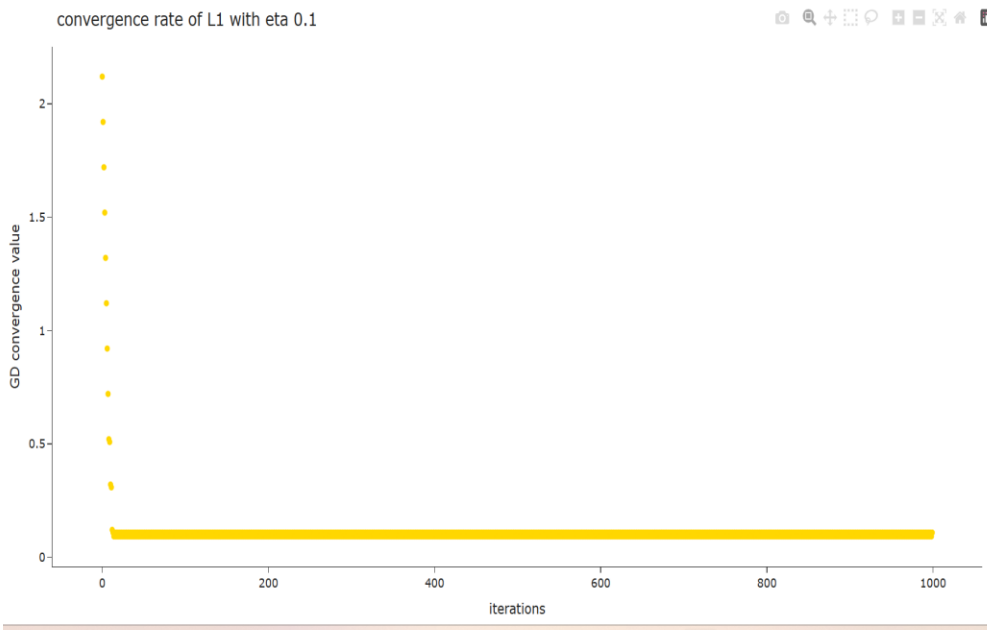
3. For each of the modules, plot the convergence rate (i.e. the norm as a function of the GD iteration) for all specified learning rates. Explain your results

L1

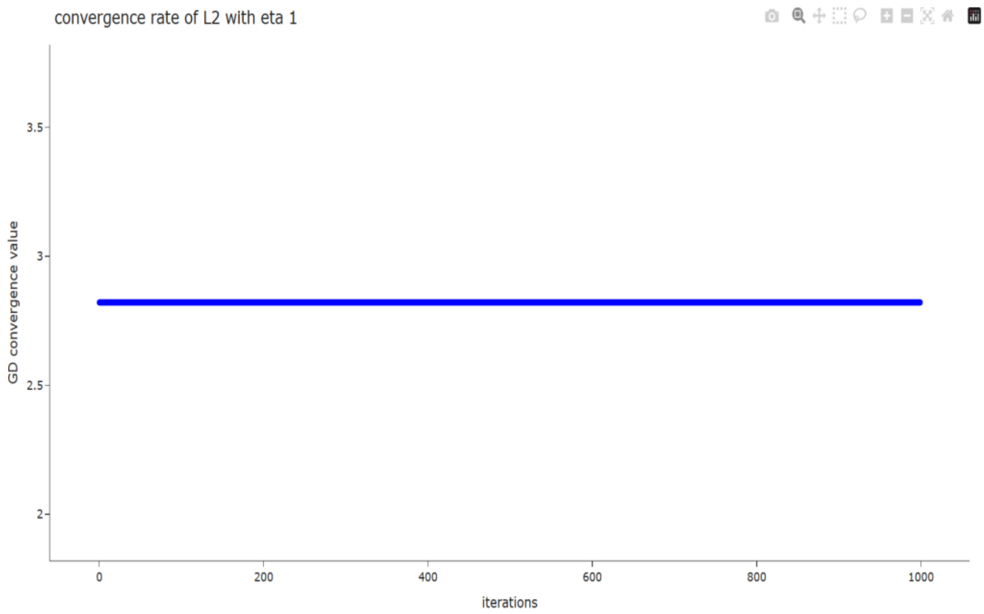



convergence rate of L1 with eta 0.01
GD convergence value
iterations
0
200
400
600
800
1000
0
0.5
1
1.5
2

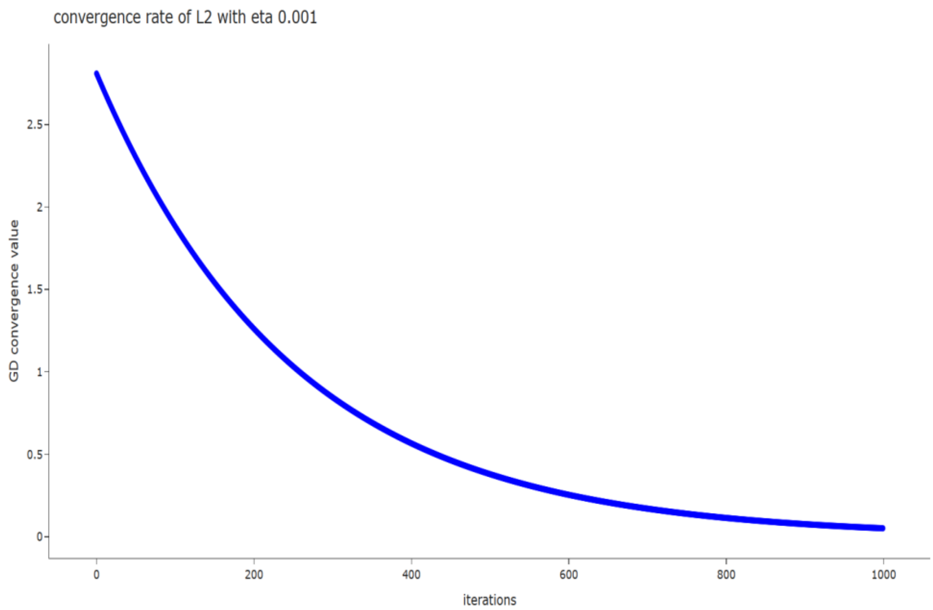

convergence rate of L1 with eta 0.001
GD convergence value
0
200
400
600
800
1000
0.5
1
1.5
2


convergence rate of L1 with eta 0.1
GD convergence value
iterations
0
200
400
600
800
1000
0
0.5
1
1.5
2

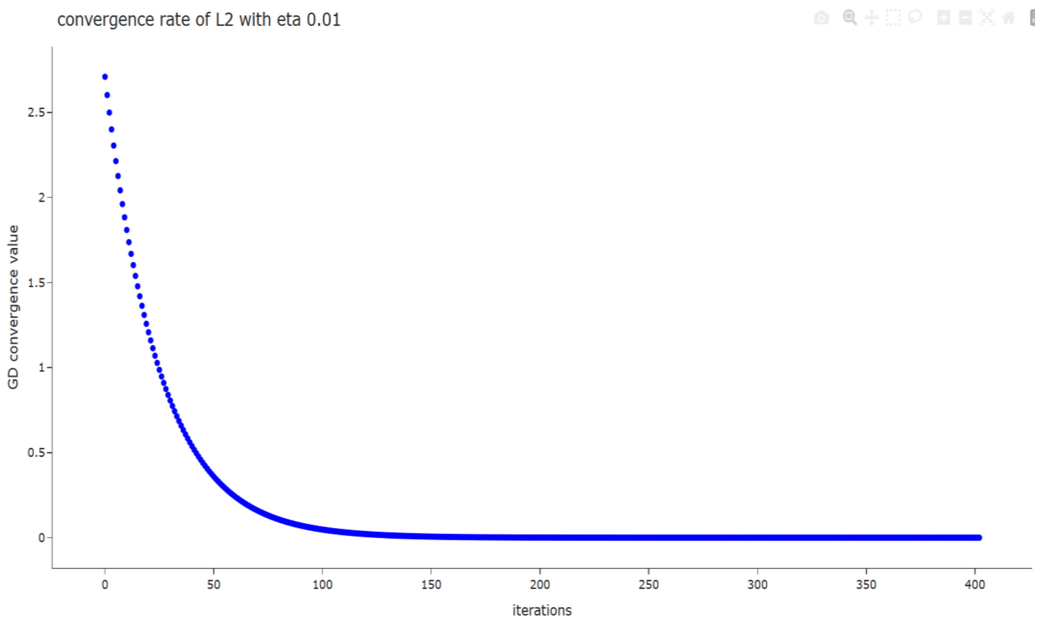
:L2

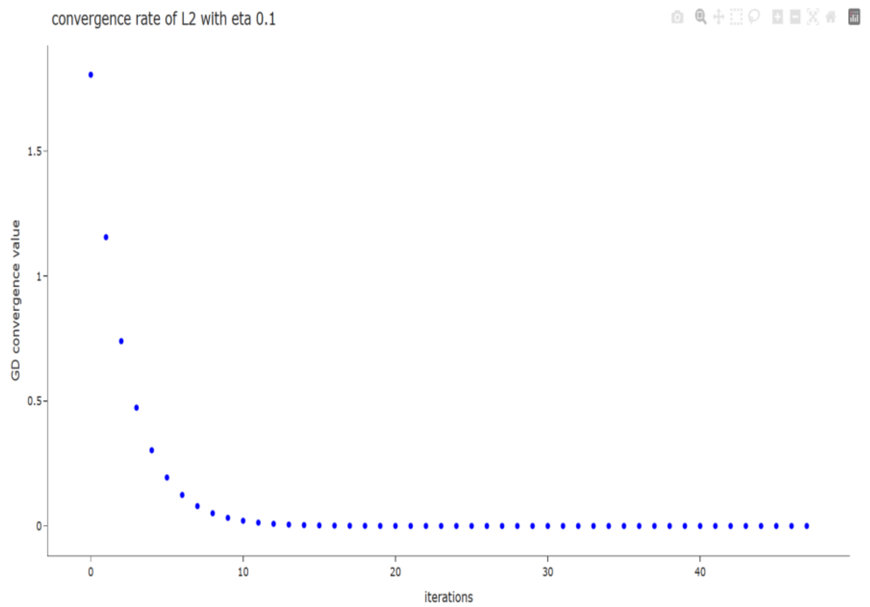


convergence rate of L2 with eta 1
GD convergence value
3.5
3
2.5
2
0
200
400
600
800
1000
iterations



convergence rate of L2 with eta 0.001
GD convergence value
2.5
2
1.5
1
0.5
0
0
200
400
600
800
1000
iterations





נסתכל על הגרפים, קל לראות כי מהגדרת אל 1 ואל 2 עבור אל1 נקבל פונקציה עם הרבה נקודות אי רציפות שמתאפס מהר לעומת אל2 שיותר רציפה ומתאפסת לאט יותר כלומר מתכווצת יחד לאט לאט

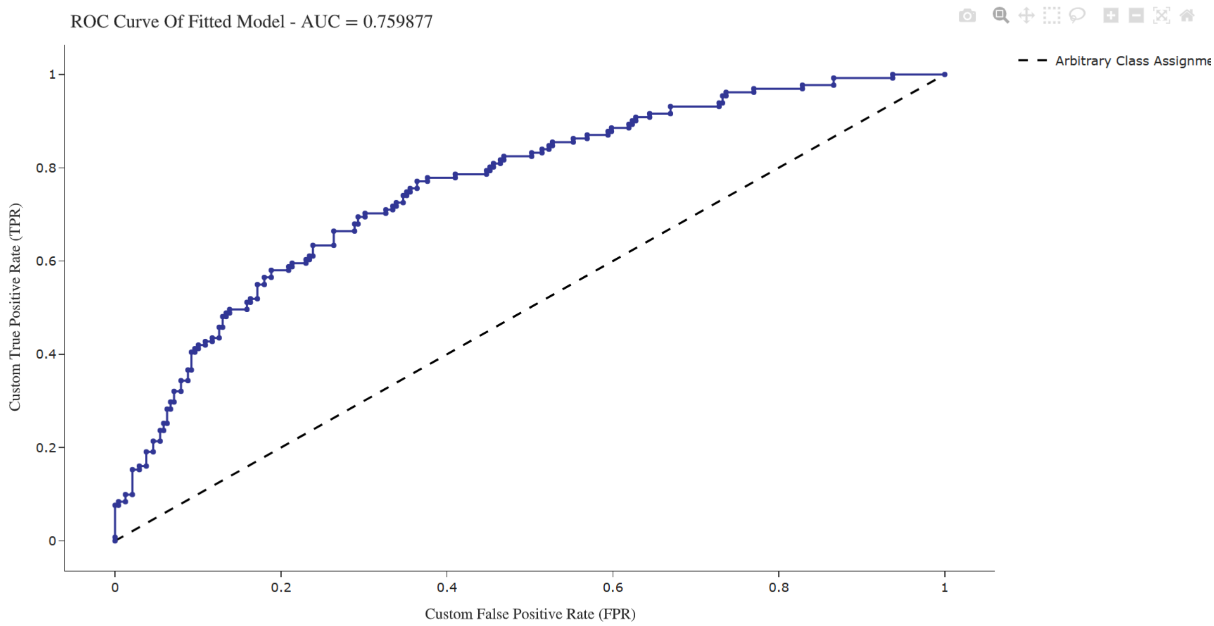
4. What is the lowest loss achieved when minimizing each of the modules? Explain the differences

```
best loss L1 module is:   0.011880380446586989 with eta 0.01
best loss L2 module is:   1.4029519498344255e-09 with eta 0.1
```

מכך שאל1 קופץ בין הערכים אנו מעט רחוקים יותר מ0 לעומת אל 2 שבכל צעד מקווץ את הפונקציה ויורד מונוטנית ל0 נקבל ערך ממש קטן

Then, load the South Africa Heart Disease dataset (SAheart.data), split it to train- and test sets (80% train) and answer the following questions:

8. Using your implementation, fit a logistic regression model over the data. Use the `predict_proba` to plot an ROC curve. You can use sklearn's `metrics.roc_curve` function and the code provided in Lab 04.



9. Which value of α achieves the optimal ROC value according to the criterion below. Using this value of α^* what is the model's test error?

$$\alpha^* = \text{argmax}_\alpha\{\text{TPR}_\alpha - \text{FPR}_\alpha\}$$

```
The best alpha is: 0.32
Model's test error: 0.33695652173913043
```

10. Fit an ℓ_1-regularized logistic regression by passing `penalty="l1"` when instantiating a logistic regression estimator
    - Set $\alpha = 0.5$
    - Use your previously implemented cross-validation procedure to choose λ
    - After selecting λ repeat fitting with the chosen λ and $\alpha = 0.5$ over the entire train portion.

For values of What value of λ was selected and what is the model's test error?

11. Repeat question 10 for ℓ_2 regularized logistic regression. What value of λ was selected and what is the model's test error?

Best lamda L1 is 0.02 and its loss is 0.27

Best lamda L2 is 0.01 and its loss is 0.29