Submitted by: Oluwasegun Somefun, somefuno@oregonstate.edu

AI534 — Written Homework Assignment 2 (40 pts) — Due Oct 29th, 2021

1. (Subgradient) (5 pts) Consider the L_1 norm function for $\mathbf{x} \in R^d$: $f(\mathbf{x}) = |\mathbf{x}|_1 = \sum_{i=1}^{d} |x_i|$. Show that $\mathbf{g} = [g_1, g_2, ..., g_d]^T$ is a subgradient of $f(\mathbf{x})$ at $\mathbf{x} = \mathbf{0}$ if every $g_i \in [-1, 1]$. Hint: go back to the definition of subgradient: g is a subgradient of $f(x)$ at x_0 if $\forall x$, $f(x) \geq f(x_0) + g^T(x - x_0)$

   Given the $\mathcal{L}_1$ norm function $f(\mathbf{x}) = \|\mathbf{x}\|_1 = \sum_{i=1}^{d} |x_i|$, having each element as absolute values. Recall that, $\forall x$, the subgradient g of $f(\mathbf{x})$ is defined such that for a convex function:

   $$f(x) \geq f(x_0) + g^T(x - x_0)$$

   It is known that $\forall x$: when $x_0 > 0$, then $g = \nabla f(x) = 1$; when $x_0 < 0$, then $g = \nabla f(x) = -1$.

   At $x_0 = 0$, the subgradient expression of the convex function reduces to the inequality

   $$f(x) \geq g^T x \equiv |x| \geq g\, x$$

   such that the definition of the gradient of the absolute value function is satisfied if and only if g is 1 or -1, that is $g \in [-1, 1]$

   Therefore, $\mathbf{g} = [g_1, g_2, ..., g_d]^T$ is a subgradient of $f(\mathbf{x}) = \|\mathbf{x}\|_1$ at $\mathbf{x} = \mathbf{0}$ if every $g_i \in [-1, 1]$.

2. (Perceptron) (5 pts) Consider the following argument. We know that the number of steps for the perceptron algorithm to converge for linearly separable data is bounded by $(\frac{D}{\gamma})^2$. If we multiple the input $\mathbf{x}$ by a small constant α, which effectively reduces the bound on $|\mathbf{x}|$ to $D' = \alpha D$, we can reduce the upper bound to $(\alpha \frac{D}{\gamma})^2$. Is this argument correct? Why?

   The argument above is correct. The number of steps for the perceptron algorithm to converge for linearly separable data is bounded by $(\alpha \frac{D}{\gamma})^2$. This argument can be proved by showing that given the rescaled space scenario above, the direction of the weight parameter at the kth step converges to a unit vector $\|\omega^*\| = 1$, that is

   $$\frac{\omega^{*T}\omega_k}{\|\omega^*\|\|\omega_k\|} \leq 1$$

   and that for each training sample $\|\mathbf{x}_i\| \leq D \in \mathbb{R}$, such that the decision boundary is bounded by a margin (γ) described as $y\,\omega^*\,x_k \geq \gamma > 0$. Let k be the kth mistake step at which, given a small constant $\alpha > 0$ the update is

   $$\omega_k = \omega_{k-1} + \alpha\, y\, x_k$$

   Assuming $\omega_{k-1} = 0$, to prove this, we need to first show that

   (a) $\omega^{*T}\omega_k$ grows quickly as k increases

   (b) $\|\omega_k\|$ does not grow quickly, that is: $\|\omega_k\|$ is getting close to $\|\omega^*\|$

   For Part (a):

   $$\begin{aligned} \omega^{*T}\omega_k &= \omega^{*T}(\omega_{k-1} + \alpha\ y\, x_k) = \omega^{*T}\omega_{k-1} + \alpha\, y\, \omega^*\, x_k \\ &\geq \omega^{*T}\omega_{k-1} + \alpha\gamma \geq k\gamma \end{aligned}$$

   Similarly, for Part (b):

   $$\begin{aligned} \omega_k^T\omega_k &= (\omega_{k-1} + \alpha\, y\, x_k)^T(\omega_{k-1} + \alpha\, y\, x_k) \\ &= \omega_k^T\omega_{k-1} + \alpha\, 2\, y\, \omega_{k-1}\, x_k + (\alpha\, y)^2\, x_k^T x_k \\ &\leq \omega_k^T\omega_{k-1} + (\alpha\, D)^2 \leq k\,(\alpha\, D)^2 \\ \therefore \|\omega_k\| &= \sqrt{\omega_k^T\omega_k} = \sqrt{k}(\alpha\, D) \end{aligned}$$

$$\frac{\omega^{*T}\omega_k}{\|\omega^*\|\|\omega_k\|} = \frac{k\gamma}{\sqrt{k}(\alpha\, D)} \leq 1$$

Therefore, the upper bound number of steps it takes the perceptron algorithm to converge is reduced to:

$$k \leq \left(\alpha\, \frac{D}{\gamma}\right)^2$$

This concludes the proof.

3. (Cubic Kernels.) (10 pts) In class, we showed that the quadratic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^2$ was equivalent to mapping each $\mathbf{x} = (x_1, x_2) \in R^2$ into a higher dimensional space where

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

Now consider the cubic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$. What is the corresponding Φ function?

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$$

Let

$$\mathbf{x}_i = x_{i1}, x_{i2}, \mathbf{x}_j = x_{j1}, x_{j2}$$

and

$$a = x_{i1}x_{j1} + x_{i2}x_{j2} = a_1 + a_2$$

Then, we can rewrite

$$K(\mathbf{x}_i, \mathbf{x}_j) = (a+1)^3 = a^3 + 3a^2 + 3a + 1$$

where

$$a^2 = (a_1 + a_2)^2 = a_1^2 + 2a_1a_2 + a_2^2 \implies 3a^2 = 3a_1^2 + 6a_1a_2 + 3a_2^2$$
$$a^3 = (a_1 + a_2)^3 = a_1^3 + 3a_1^2a_2 + 3a_2^2a_1 + a_2^3$$

$\therefore$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(a_1^3 + 3a_1^2a_2 + 3a_2^2a_1 + a_2^3 + 3a_1^2 + 6a_1a_2 + 3a_2^2 + 3a_1 + 3a_2 + 1\right)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(a_1^3 + 3a_1^2 + 3a_1 + 3a_1^2a_2 + 6a_1a_2 + 3a_2^2a_1 + 3a_2 + 3a_2^2 + a_2^3 + 1\right)$$

Recall that $a_1 = x_{i1}x_{j1}$ and $a_2 = x_{i2}x_{j2}$. Therefore:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= \left(x_{i1}^3, \sqrt{3}x_{i1}^2, \sqrt{3}x_{i1}, \sqrt{3}x_{i1}^2x_{i2}, \sqrt{6}x_{i1}x_{i2}, \sqrt{3}x_{i2}^2x_{i1}, \sqrt{3}x_{i2}, \sqrt{3}x_{i2}^2, x_{i2}^3, 1\right) \cdot \\ &\quad \left(x_{j1}^3, \sqrt{3}x_{j1}^2, \sqrt{3}x_{j1}, \sqrt{3}x_{j1}^2x_{j2}, \sqrt{6}x_{j1}x_{j2}, \sqrt{3}x_{j2}^2x_{j1}, \sqrt{3}x_{j2}, \sqrt{3}x_{j2}^2, x_{j2}^3, 1\right) \\ K(\mathbf{x}_i, \mathbf{x}_j) &= \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \end{aligned}$$

which means that the kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$ in terms of explicit feature mapping is equivalent to $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$.

4. (Kernel or not). In the following problems, suppose that K, K_1 and K_2 are kernels with feature maps ϕ, ϕ_1 and ϕ_2. For the following functions $K'(x, z)$, state if they are kernels or not. If they are kernels, write down the corresponding feature map, in terms of ϕ, ϕ_1 and ϕ_2 and c, c_1, c_2. If they are not kernels, prove that they are not.

The necessary and sufficient conditions for a function to be a valid kernel function is that for any finite sample, its corresponding kernel matrix K' be **positive semi-definite (p.s.d)** and **symmetric** . This is also known as the **Mercer's theorem**

- (5 pts) $K'(\mathbf{x}, \mathbf{z}) = cK(\mathbf{x}, \mathbf{z})$ for $c > 0$.

  According to the properties of kernels: any positive rescaling of a kernel is also a kernel. Therefore K' is a kernel. The corresponding feature map is given as

  $$K'(\mathbf{x}, \mathbf{z}) = c\,\langle \phi(\mathbf{x}), \phi(\mathbf{z})\rangle = \langle \sqrt{c}\,\phi(\mathbf{x}), \sqrt{c}\,\phi(\mathbf{z})\rangle$$

- (5 pts) $K'(\mathbf{x}, \mathbf{z}) = cK(\mathbf{x}, \mathbf{z})$ for $c < 0$.

  K' is not a kernel. As a counter-example, this is proved as follows: since $c < 0$, $\exists$ an element in the transformed matrix K' that is negative definite, which violates the necessary p.s.d **Mercer's conditions** stated above.

  Therefore the scaling of K to K' is **not** a valid kernel.

- (5 pts) $K'(\mathbf{x}, \mathbf{z}) = c_1K_1(\mathbf{x}, \mathbf{z}) + c_2K_2(\mathbf{x}, \mathbf{z})$ for $c_1, c_2 > 0$.

  According to the properties of kernels: any positive linear combination of kernels is also a kernel. Therefore K' is a kernel. The corresponding feature map is given as

  $$K'(\mathbf{x}, \mathbf{z}) = \langle \sqrt{c_1}\,\phi_1(\mathbf{x}), \sqrt{c_1}\,\phi_1(\mathbf{z})\rangle + \langle \sqrt{c_2}\,\phi_2(\mathbf{x}), \sqrt{c_2}\,\phi_2(\mathbf{z})\rangle = \langle \phi'(\mathbf{x}), \phi'(\mathbf{z})\rangle$$

  where the number of features in ϕ' is a concatenation of the number features in both feature maps ϕ_1 and ϕ_2

- (5 pts) $K'(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z})K_2(\mathbf{x}, \mathbf{z})$.

  According to the properties of kernels: any product of two or more kernels is also a kernel. Therefore K' is a kernel. The corresponding feature map is given as

  $$K'(\mathbf{x}, \mathbf{z}) = \langle \phi_1(\mathbf{x}), \phi_1(\mathbf{z})\rangle\,\langle \phi_2(\mathbf{x}), \phi_2(\mathbf{z})\rangle = \langle \phi'(\mathbf{x}), \phi'(\mathbf{z})\rangle$$

  where the number of features in ϕ' is a linear product of the number features in both feature maps ϕ_1 and ϕ_2