Derivatives for Kernel method

1 Binary Classification

1.1 Problems and Models

Let $g_j(x) := \partial^j f(x)$. Assume $g_j \in \mathcal{H}$ which is true for the kernels we consider. If $\sup_{x\in\mathcal{X}} k(x,x) = 1$, then

$$\begin{aligned}\sup_{x\in\mathcal{X}} \|g(x)\|_2^2 = \sup_{x\in\mathcal{X}} \sum_{j=1}^d g_j(x)^2 &= \sup_{x\in\mathcal{X}} \sum_{j=1}^d \langle g_j, k(x,\cdot)\rangle_{\mathcal{H}}^2 \\ &\le \sup_{x\in\mathcal{X}} \sum_{j=1}^d \|g_j\|_{\mathcal{H}}^2 \|k(x,\cdot)\|_{\mathcal{H}}^2 = \sum_{j=1}^d \|g_j\|_{\mathcal{H}}^2 . \end{aligned} \tag{1}$$

$$\sup_{x\in\mathcal{X}} \|g(x)\|_1 = \sup_{x\in\mathcal{X}} \sum_{j=1}^d |g_j(x)| \le \sup_{x\in\mathcal{X}} \sum_{j=1}^d \|g_j\|_{\mathcal{H}} \|k(x,\cdot)\|_{\mathcal{H}} = \sum_{j=1}^d \|g_j\|_{\mathcal{H}} . \tag{2}$$

The exact evaluation of $\|g_j\|_{\mathcal{H}}$ still remains challenging, but it is amenable to the Nyström approximation. Intuitively, $\|g_j\|_{\mathcal{H}} \approx \|\tilde{g}_j\|_2$, where $\tilde{g}_j \in R^n$ is the Nyström approximation computed by

$$\tilde{g}_j := K_W^{-\frac{1}{2}} \begin{pmatrix} g_j(w^1) \\ \vdots \\ g_j(w^n) \end{pmatrix} = K_W^{-\frac{1}{2}} \begin{pmatrix} \langle g_j, k(w^1,\cdot)\rangle_{\mathcal{H}} \\ \vdots \\ \langle g_j, k(w^n,\cdot)\rangle_{\mathcal{H}} \end{pmatrix}, \quad \text{where} \quad K_W = [k(w^i, w^{i'})]_{i,i'} \tag{3}$$

$$= (Z^\top Z)^{-\frac{1}{2}} Z^\top g_j, \quad \text{where} \quad Z = (k(w^1,\cdot), k(w^2,\cdot), \ldots, k(w^n,\cdot)). \tag{4}$$

By Adding (1) and (2) to regular kernel machine as constraits, we obtain our defense models for ℓ_2 or ℓ_∞ attacks, respectively.

(A). Defense ℓ_2 attacks:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^n \ell(\langle k(x_i,\cdot), f\rangle_{\mathcal{H}}, y_i), \quad \text{where } f = \sum_{i=1}^n \alpha_i k(x_i,\cdot) \\ s.t. \quad & \sum_{j=1}^d \|g_j\|_{\mathcal{H}}^2 \approx \sum_{j=1}^d \left\|(Z^\top Z)^{-\frac{1}{2}} Z^\top g_j\right\|_2^2 \le L^2. \end{aligned}$$

(B). Defense ℓ_∞ attacks:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^n \ell(\langle k(x_i,\cdot), f\rangle_{\mathcal{H}}, y_i), \quad \text{where } f = \sum_{i=1}^n \alpha_i k(x_i,\cdot) \\ s.t. \quad & \sum_{j=1}^d \|g_j\|_{\mathcal{H}} \approx \sum_{j=1}^d \left\|(Z^\top Z)^{-\frac{1}{2}} Z^\top g_j\right\|_2 \le L, \end{aligned}$$

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

When the size of samples is large, the space complexity and computational complexity becomes the bottleneck. We can use Nyström approximation to approximate $k(x, \cdot)$ so that

$$f(w) = \langle k(w, \cdot), f\rangle_{\mathcal{H}} \approx \left\langle K_B^{-\frac{1}{2}} \begin{pmatrix} k(b^1, w) \\ \vdots \\ k(b^p, w) \end{pmatrix}, \boldsymbol{\alpha} \right\rangle = \left\langle \begin{pmatrix} k(b^1, w) \\ \vdots \\ k(b^p, w) \end{pmatrix}, K_B^{-\frac{1}{2}} \boldsymbol{\alpha} \right\rangle \tag{5}$$

where $K_B = [k(b^i, b^{i'})]_{i,i'} \in R^{p\times p}$ and the landmarks $B := \{b_i\}_{i=1}^p$ are either uniformly sampled from training data or through kmean.

1.2 Derivatives for solving (A) and (B)

① $\frac{\partial \ell}{\partial \boldsymbol{\alpha}}$ is trivial based on (5).

② Based on (5), the constraints in problem **(A)** is actually a quadratic constraint, i.e., $\boldsymbol{\alpha}^\top Q \boldsymbol{\alpha} \leq L^2$.

$$\begin{aligned} \sum_{j=1}^d \left\| (Z^\top Z)^{-\frac{1}{2}} Z^\top g_j \right\|_2^2 &= \sum_{j=1}^d g_j^\top Z K_W^{-1} Z^\top g_j \\ &= \sum_{j=1}^d \left(\partial^j f(w^1), \cdots, \partial^j f(w^n) \right) K_W^{-1} \begin{pmatrix} \partial^j f(w^1) \\ \vdots \\ \partial^j f(w^n) \end{pmatrix} \\ &= \sum_{j=1}^d \boldsymbol{\alpha}^\top k_B^{-\frac{1}{2}} S_j K_W^{-1} S_j' k_B^{-\frac{1}{2}} \boldsymbol{\alpha} \end{aligned}$$

where

$$S_j := \left(\partial^{0,j} \begin{pmatrix} k(b^1, w^1) \\ \vdots \\ k(b^p, w^1) \end{pmatrix}, \cdots, \partial^{0,j} \begin{pmatrix} k(b^1, w^n) \\ \vdots \\ k(b^p, w^n) \end{pmatrix} \right) \in R^{p\times n}$$

If k is Gaussian kernel, i.e.,

$$k(b, w) = \exp - \frac{\|b - w\|^2}{2\sigma^2} \quad and \quad \partial^{0,j} k(b, w) = \exp\left(- \frac{\|b - w\|^2}{2\sigma^2} \right) \frac{b_j - w_j}{\sigma^2}$$

then

$$\begin{aligned} S_j &= B_j K_{BW} - K_{BW} W_j, \\ \text{where } K_{BW} &:= \left[\exp\left(- \frac{\|b^i - w^j\|^2}{2\sigma^2} \right) / \sigma^2 \right]_{i,j} \in R^{p\times n}, \\ B_j &:= \operatorname{diag}(b_j^1, \cdots, b_j^p), \qquad W_j := \operatorname{diag}(w_j^1, \cdots, w_j^n) \end{aligned}$$

therefore, the quadratic matrix Q can be computed as

$$\begin{aligned} Q &= k_B^{-\frac{1}{2}} \sum_{j=1}^d (S_j K_W^{-1} S_j') k_B^{-\frac{1}{2}} \\ &= k_B^{-\frac{1}{2}} \sum_{j=1}^d (B_j K_{BW} K_W^{-1} K_{BW}' B_j' - B_j K_{BW} K_W^{-1} W_j' K_{BW}' \\ &\qquad - K_{BW} W_j K_W^{-1} K_{BW}' B_j' + K_{BW} W_j K_W^{-1} W_j' K_{BW}'] k_B^{-\frac{1}{2}} \\ &= k_B^{-\frac{1}{2}} [(K_{BW} K_W^{-1} K_{BW}') .* (B^\top B) - (K_{BW} K_W^{-1}) .* (B^\top W) K_{BW}' \\ &\qquad - K_{BW} (K_W^{-1} K_{BW}') .* (W^\top B) + K_{BW} (K_W^{-1} .* (W^\top W)) K_{BW}'] k_B^{-\frac{1}{2}} \end{aligned}$$

the last equality is because $\sum_i^d \text{diag}(X_i)A\text{diag}(Y_i) = A.*(X^\top Y)$

③ There are two ways to solve problem **(B)**: a) frank-wolfe algorithm; b) fmincon by directly taking derivative of the constraints function $\sum_j^d ||\tilde{g}_j||_2$ w.r.t. $\boldsymbol{\alpha}$, which is differentiable except when $\tilde{g}_j = 0$.

$$
\begin{aligned}
\frac{\partial \sum_j^d ||\tilde{g}_j||_2}{\partial \boldsymbol{\alpha}} &= \sum_{j=1}^{d} \frac{\tilde{g}_j}{\|\tilde{g}_j\|} K_W^{-\frac{1}{2}} S_j' k_B^{-\frac{1}{2}} \\
&= \sum_{j=1}^{d} \frac{\tilde{g}_j}{\|\tilde{g}_j\|} K_W^{-\frac{1}{2}} (B_j K_{BW} - K_{BW} W_j)' k_B^{-\frac{1}{2}} \\
&= \sum_{j=1}^{d} \left(\frac{\tilde{g}_j}{\|\tilde{g}_j\|} K_W^{-\frac{1}{2}} K_{BW}' B_j' - \frac{\tilde{g}_j}{\|\tilde{g}_j\|} K_W^{-\frac{1}{2}} W_j' K_{BW}' \right) k_B^{-\frac{1}{2}} \\
&= \left(sum \left(K_W^{-\frac{1}{2}} K_{BW}' .* (\frac{\tilde{g}}{\|\tilde{g}\|}^\top B) \right) - sum \left(K_W^{-\frac{1}{2}} .* (\frac{\tilde{g}}{\|\tilde{g}\|}^\top W) \right) K_{BW}' \right) k_B^{-\frac{1}{2}}
\end{aligned}
$$

the last equality is because $\mathbf{x}^\top A\text{diag}(\mathbf{y}) = sum(\text{diag}(\mathbf{x})A\text{diag}(\mathbf{y}))$

1.3 Derivatives for finding largest Lipschitz constant

We need to find the point with largest Lipschitz constant (BFGS), and add it to constraints set. This can be formulated as follows,

(a) $\max_x \|\nabla f(x)\|_2^2$: the derivative is $2 * \nabla f(x) * \nabla^2 f(x)$

(b) $\max_x \|\nabla f(x)\|_1$: the derivative is $\text{sign}(\nabla f(x)) * \nabla^2 f(x)$

$$
\begin{aligned}
\nabla f(x) &= \nabla_x \begin{pmatrix} k(b^1, x) \\ \vdots \\ k(b^p, x) \end{pmatrix} \times K_B^{-\frac{1}{2}} \boldsymbol{\alpha} = \left(K_{b^1 w} \frac{b^1 - x}{\sigma^2}, \cdots, K_{b^p w} \frac{b^p - x}{\sigma^2} \right) K_B^{-\frac{1}{2}} \boldsymbol{\alpha} \\
&= \left(\frac{b^1 - x}{\sigma^2}, \cdots, \frac{b^p - x}{\sigma^2} \right) (K_{Bw} .* K_B^{-\frac{1}{2}} \boldsymbol{\alpha})
\end{aligned}
$$

$$
\begin{aligned}
\nabla^2 f(x) &= \nabla_x \begin{pmatrix} k(b^1, x) \\ \vdots \\ k(b^p, x) \end{pmatrix} \times K_B^{-\frac{1}{2}} \boldsymbol{\alpha} = \left(K_{b^1 w} \frac{b^1 - x}{\sigma^2}, \cdots, K_{b^p w} \frac{b^p - x}{\sigma^2} \right) K_B^{-\frac{1}{2}} \boldsymbol{\alpha} \\
&= \left(\frac{b^1 - x}{\sigma^2}, \cdots, \frac{b^p - x}{\sigma^2} \right) (K_{Bw} .* K_B^{-\frac{1}{2}} \boldsymbol{\alpha})
\end{aligned}
$$

1.4 Tigher relaxation in (1) and (2)

In fact, we can have a tighter relaxation than (1):

$$
\sup_{x \in \mathcal{X}} \|g(x)\|_2^2 = \sup_{x \in \mathcal{X}} \sum_{j=1}^{d} g_j(x)^2 = \sup_{x \in \mathcal{X}} \sum_{j=1}^{d} \langle g_j, k(x, \cdot) \rangle_{\mathcal{H}}^2 \leq \sup_{\|u\| \leq 1} u^T (\sum_j g_j g_j^T) u. \tag{6}
$$

where the last inequality is because representer elements $k(x, \cdot)$ are just a proper subset of the unit sphere of RKHS. Hence, model (A) becomes

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n} \ell(\langle k(x_i, \cdot), f \rangle_{\mathcal{H}}, y_i), \quad \text{where } f = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) \\
s.t. \quad & \lambda_{\max}(\sum_j g_j g_j^T) \approx \lambda_{\max}(\sum_j \tilde{g}_j \tilde{g}_j^T) \leq L^2.
\end{aligned}
$$

Note the gradient of constraint function $\lambda_{\max}(\sum_j \tilde{g}_j \tilde{g}_j^T)$ is equivalent to $u_*^T(\sum_{j=1}^d \tilde{g}_j \tilde{g}_j^T)u_* = \sum_{j=1}^d (u_*^T \tilde{g}_j)^2$ where u_* is the leading eigenvector of matrix $\sum_j \tilde{g}_j \tilde{g}_j^T$. Then the gradient of constraint function is

$$
\begin{aligned}
\frac{\partial \sum_{j=1}^d (u_*^T \tilde{g}_j)^2}{\partial \boldsymbol{\alpha}} &= \sum_{j=1}^d 2*(u_*^T \tilde{g}_j) u_*^T \frac{\partial \tilde{g}_j}{\partial \boldsymbol{\alpha}} \\
&= \sum_{j=1}^d 2*(u_*^T \tilde{g}_j) u_*^T \frac{\partial K_W^{-1/2} S_j' k_B^{-\frac{1}{2}} \boldsymbol{\alpha}}{\partial \boldsymbol{\alpha}} \\
&= \sum_{j=1}^d 2*(u_*^T \tilde{g}_j) u_*^T K_W^{-1/2} S_j' k_B^{-\frac{1}{2}} \\
&= 2*u_*^T K_W^{-1/2} \sum_{j=1}^d \left((u_*^T \tilde{g}_j) * S_j' \right) k_B^{-\frac{1}{2}}
\end{aligned}
$$

On the other hand, a tighter relaxation than (2) is as follows:

$$
\sup_{x \in \mathcal{X}} \|g(x)\|_1 = \sup_{x \in \mathcal{X}} \sup u^\top g(x) \le \sup_{\|\phi\|_2 \le 1, \|u\|_\infty \le 1} u^\top \tilde{G}^\top \phi \tag{7}
$$

where $\tilde{G}_c = [\tilde{g}_1^c, \ldots, \tilde{g}_d^c] \in R^{n \times d}$. Alternatively updating u and ϕ gives the tighter bound.

2 Multiclass Classification

2.1 Problems and Models

2.1.1 Defense ℓ_2 attacks

For multiclass classification (e.g. 10 classes in MNIST), we will using multiclass loss. Let $F(x)$ be the logits from classifier and κ be the margin, then we could use following loss function:

$$
\begin{aligned}
&\text{Cramer-Singer:} && \max_{i \ne y} \left(0, \kappa + (F_i(x) - F_y(x))\right), \\
&\text{Weston-Watkins:} && \sum_{i \ne y} \max \left(0, \kappa + (F_i(x) - F_y(x))\right), \\
&\text{Cross-entropy:} && -\mathbf{y}^\top \log(\text{softmax}(F(x) - \kappa e_y)).
\end{aligned}
$$

To enforce the smoothness of the multiclass classifier, we would like to enforce spectral norm of its Jacobian matrix:

$$
\begin{aligned}
& \sup_{x \in \mathcal{X}} \left\| \left[g^1(x), \ldots, g^{10}(x) \right] \right\|_{sp}^2 \\
= & \sup_{x \in \mathcal{X}} \lambda_{\max}(\left[g^1(x), \ldots, g^{10}(x) \right] \left[g^1(x), \ldots, g^{10}(x) \right]^\top) \\
= & \sup_{x \in \mathcal{X}} \lambda_{\max}(\sum_{c=1}^{10} G_c^\top k(x, \cdot) k(x, \cdot)^\top G_c), \quad \text{where} \quad G_c := [g_1^c, \ldots, g_d^c] \\
\le & \sup_{\|v\|_{\mathcal{H}} \le 1} \lambda_{\max}(\sum_{c=1}^{10} G_c^\top v v^\top G_c) \le L^2
\end{aligned}
$$

Therefore, the overall learning model is

(A). Defense ℓ_2 attacks:

$$\min_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} \ell(F(x), \mathbf{y}), \quad \text{where } F = \left[\sum_{i=1}^{n} \boldsymbol{\alpha}_i^1 k(x_i, \cdot); \ldots ; \sum_{i=1}^{n} \boldsymbol{\alpha}_i^{10} k(x_i, \cdot)\right]$$

$$s.t. \qquad \sup_{\|v\|_{\mathcal{H}} \le 1} \lambda_{\max}\Big(\sum_{c=1}^{10} G_c^\top v v^\top G_c\Big) \approx \sup_{\|v\|_2 \le 1} \lambda_{\max}\Big(\sum_{c=1}^{10} \tilde{G}_c^\top v v^\top \tilde{G}_c\Big) \le L^2.$$

To solve this problem, we use fmincon solver which requires the gradients of objective and constraints w.r.t. $\boldsymbol{\alpha}$ as well as the hessian of the Lagrangian.

Note that the constraint itself is a supremum problem, so we need to solve the constraint at first. The constraint is a supremum problem: $\sup_{\|v\|_2 \le 1} \lambda_{\max}(\sum_{c=1}^{10} \tilde{G}_c^\top v v^\top \tilde{G}_c) = \sup_{\|v\|_2 \le 1, \|u\|_2 \le 1} u^\top \sum_{c=1}^{10} \tilde{G}_c^\top v v^\top \tilde{G}_c) u$. To solve it, we can alternatively update v and u and obtain the optimal v_* and u_*.

We know that

$$R^{n \times d} \ni \tilde{G}_c = [\tilde{g}_1^c, \ldots, \tilde{g}_d^c] = K_W^{-\frac{1}{2}} \left[\begin{pmatrix} \partial^1 f^c(w^1) \\ \vdots \\ \partial^1 f^c(w^n) \end{pmatrix}, \ldots, \begin{pmatrix} \partial^d f^c(w^1) \\ \vdots \\ \partial^d f^c(w^n) \end{pmatrix} \right]$$

and recall (3) that

$$f^c(w) = \left\langle \begin{pmatrix} k(b^1, w) \\ \vdots \\ k(b^p, w) \end{pmatrix}, K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c \right\rangle$$

$$\text{and for Gaussian kernel} \quad \nabla_w f^c(w) = \left[K_{b^1 w} \frac{b^1 - w}{\sigma^2}, \ldots, K_{b^p w} \frac{b^p - w}{\sigma^2} \right] K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c$$
$$= \Big(B * (K_{Bw} .* K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c) - w * (K_{Bw}^\top K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c) \Big) / \sigma^2$$

combining above two equations we have

$$\tilde{G}_c = K_W^{-\frac{1}{2}} \left[\begin{pmatrix} \partial^1 f^c(w^1) \\ \vdots \\ \partial^1 f^c(w^n) \end{pmatrix}, \ldots, \begin{pmatrix} \partial^d f^c(w^1) \\ \vdots \\ \partial^d f^c(w^n) \end{pmatrix} \right]$$
$$= K_W^{-\frac{1}{2}} \left[\frac{B \mathrm{diag}(K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c) K_{BW} - W \mathrm{diag}(K_{BW}^\top K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c)}{\sigma^2} \right]'$$

for Gaussian kernel. For l-layer inverse kernel,

$$\nabla_w f^c(w) = \left[\frac{1}{(l + 1 - l b^1 w)^2} b^1, \ldots, \frac{1}{(l + 1 - l b^p w)^2} b^p \right] K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c$$
$$= B * \left(\frac{1}{(l + 1 - l B w)^2} .* K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c \right)$$

Then

$$\tilde{G}_c = K_W^{-\frac{1}{2}} \left[\begin{pmatrix} \partial^1 f^c(w^1) \\ \vdots \\ \partial^1 f^c(w^n) \end{pmatrix}, \ldots, \begin{pmatrix} \partial^d f^c(w^1) \\ \vdots \\ \partial^d f^c(w^n) \end{pmatrix} \right] = K_W^{-\frac{1}{2}} \left[B \mathrm{diag}(K_B^{-\frac{1}{2}} \boldsymbol{\alpha}^c) \frac{1}{(l + 1 - l B W)^2} \right]'$$

Finally, the original spectral norm constraint becomes

$$C(\{\boldsymbol{\alpha}^c\}) = \sum_{c=1}^{10} (u_*^\top \tilde{G}_c^\top v_*)^2 \le L^2.$$

Its partial derivative (Gaussian kernel) to each $\boldsymbol{\alpha}^c$ is

$$
\begin{aligned}
&\frac{\partial C(\{\boldsymbol{\alpha}^c\})}{\partial \boldsymbol{\alpha}^c} = \frac{\partial (u_*^\top \tilde{G}_c^\top v_*)^2}{\partial \boldsymbol{\alpha}^c} \\
=&2 * (u_*^\top \tilde{G}_c^\top v_*) * \frac{\partial u_*^\top \left[\frac{B\mathrm{diag}(K_B^{-\frac{1}{2}}\boldsymbol{\alpha}^c)K_{BW} - W\mathrm{diag}(K_{BW}^\top K_B^{-\frac{1}{2}}\boldsymbol{\alpha}^c)}{\sigma^2} \right] K_W^{-\frac{1}{2}} v_*}{\partial \boldsymbol{\alpha}^c} \\
=&2 * (u_*^\top \tilde{G}_c^\top v_*) * \frac{\partial \left(u_*^\top B\mathrm{diag}(K_B^{-\frac{1}{2}}\boldsymbol{\alpha}^c)K_{BW}K_W^{-\frac{1}{2}} v_* - u_*^\top W\mathrm{diag}(K_{BW}^\top K_B^{-\frac{1}{2}}\boldsymbol{\alpha}^c)K_W^{-\frac{1}{2}} v_* \right) / \sigma^2}{\partial \boldsymbol{\alpha}^c} \\
=&2 * (u_*^\top \tilde{G}_c^\top v_*) * \left[\left((u_*^\top B)^\top . * K_{BW}K_W^{-\frac{1}{2}} v_* \right) - \left((u_*^\top W)^\top . * K_W^{-\frac{1}{2}} v_* \right) * K_{BW}^\top \right] * K_B^{-\frac{1}{2}} / \sigma^2
\end{aligned}
$$

where the last equation is because $\frac{\partial x^\top \mathrm{diag}(\mathbf{a}) y}{\partial \mathbf{a}} = \frac{\partial (x.*y)^\top \mathbf{a}}{\partial \mathbf{a}} = x.*y$.

For inverse kernel, its partial derivative is

$$
\begin{aligned}
&\frac{\partial C(\{\boldsymbol{\alpha}^c\})}{\partial \boldsymbol{\alpha}^c} = \frac{\partial (u_*^\top \tilde{G}_c^\top v_*)^2}{\partial \boldsymbol{\alpha}^c} \\
=&2 * (u_*^\top \tilde{G}_c^\top v_*) * \frac{\partial u_*^\top \left[B\mathrm{diag}(K_B^{-\frac{1}{2}}\boldsymbol{\alpha}^c) \frac{1}{(l+1-lBW)^2} \right] K_W^{-\frac{1}{2}} v_*}{\partial \boldsymbol{\alpha}^c} \\
=&2 * (u_*^\top \tilde{G}_c^\top v_*) * \frac{\partial \left(u_*^\top B\mathrm{diag}(K_B^{-\frac{1}{2}}\boldsymbol{\alpha}^c) \frac{1}{(l+1-lBW)^2} K_W^{-\frac{1}{2}} v_* \right)}{\partial \boldsymbol{\alpha}^c} \\
=&2 * (u_*^\top \tilde{G}_c^\top v_*) * \left[(u_*^\top B)^\top . * \frac{1}{(l+1-lBW)^2} K_W^{-\frac{1}{2}} v_* \right] * K_B^{-\frac{1}{2}}
\end{aligned}
$$

2.1.2 Defense ℓ_∞ attacks

To defense ℓ_∞ attacks, we need to enforce ℓ_∞ norm of the Jacobian matrix:

$$
\begin{aligned}
&\sup_{x \in \mathcal{X}} \left\| \left[g^1(x), \ldots, g^{10}(x) \right]^\top \right\|_\infty \\
=&\sup_{x \in \mathcal{X}} \max_{1 \le c \le 10} \|g^c(x)\|_1 = \max_{1 \le c \le 10} \sup_{x \in \mathcal{X}} \|g^c(x)\|_1 \\
\le& \max_{1 \le c \le 10} \sup_{\|\phi\|_2 \le 1, \|u\|_\infty \le 1} u^\top \tilde{G}_c^\top \phi
\end{aligned}
$$

where the last inequality is due to (7).

Therefore, the overall learning model is

(B). Defense ℓ_∞ attacks:

$$
\begin{aligned}
&\min_{\boldsymbol{\alpha}} \quad \sum_{i=1}^n \ell(F(x), \mathbf{y}), \quad \text{where } F = \left[\sum_{i=1}^n \boldsymbol{\alpha}_i^1 k(x_i, \cdot); \ldots; \sum_{i=1}^n \boldsymbol{\alpha}_i^{10} k(x_i, \cdot) \right] \\
&s.t. \quad \sup_{\|\phi\|_2 \le 1, \|u\|_\infty \le 1} u^\top \tilde{G}_c^\top \phi \le L, \qquad \forall c \in \{1, \ldots, 10\}
\end{aligned}
$$