Interpreting Robust Optimization via Adversarial Influence Functions

Zhun Deng [1] **Cynthia Dwork** [1] **Jialiang Wang** [1] **Linjun Zhang** [2]

Abstract

Robust optimization has been widely used in nowadays data science, especially in adversarial training. However, little research has been done to quantify how robust optimization changes the optimizers and the prediction losses comparing to standard training. In this paper, inspired by the influence function in robust statistics, we introduce the Adversarial Influence Function (AIF) as a tool to investigate the solution produced by robust optimization. The proposed AIF enjoys a closed-form and can be calculated efficiently. To illustrate the usage of AIF, we apply it to study model sensitivity — a quantity defined to capture the change of prediction losses on the natural data after implementing robust optimization. We use AIF to analyze how model complexity and randomized smoothing affect the model sensitivity with respect to specific models. We further derive AIF for kernel regressions, with a particular application to neural tangent kernels, and experimentally demonstrate the effectiveness of the proposed AIF. Lastly, the theories of AIF will be extended to distributional robust optimization.

1. Introduction

Robust optimization is a classic field of optimization theory that seeks to achieve a certain measure of robustness against uncertainty in the parameters or inputs involved (Ben-Tal et al., 2009; Beyer & Sendhoff, 2007). Recently, it has been used to address a concern in deep neural networks — the deep neural networks are vulnerable to adversarial perturbations (Goodfellow et al., 2014; Szegedy et al., 2013).

In supervised learning, given input x, output y and a certain loss function l, adversarial training through robust optimization for a model $\mathcal{M}$ is formulated as

$$\min_{\theta^{\mathcal{M}}\in\Theta} \mathbb{E}_{x,y} \max_{\delta\in\mathcal{R}(x)} l(\theta^{\mathcal{M}}, x+\delta, y, \mathcal{M}), \tag{1}$$

where $\mathcal{R}(x)$ is some constrained set, which is usually taken as a small neighborhood of x in robust optimization. For example, in image recognition (He et al., 2016), an adversarial attack should be small so that it is visually imperceptible.

Although adversarial training through robust optimization has achieved great success in defending against adversarial attacks (Madry et al., 2017), the influence of such adversarial training on predictions is under-explored, even for a simple model $\mathcal{M}$. In particular, let us define the regular optimizer and the robust optimizer respectively:

$$\begin{aligned} \theta^{\mathcal{M}}_{\min} &:= \arg\min_{\theta^{\mathcal{M}}\in\Theta} \mathbb{E}_{x,y} l(\theta^{\mathcal{M}}, x, y, \mathcal{M}), \\ \theta^{\mathcal{M}}_{\varepsilon,\min} &:= \arg\min_{\theta^{\mathcal{M}}\in\Theta} \mathbb{E}_{x,y} \max_{\delta\in\mathcal{R}(x,\varepsilon)} l(\theta^{\mathcal{M}}, x+\delta, y, \mathcal{M}). \end{aligned} \tag{2}$$

It is unclear how $\mathbb{E}_{x,y} l(\theta^{\mathcal{M}}_{\varepsilon,\min}, x, y, \mathcal{M})$ — the prediction loss on the original data with robust optimizer— performs compared to the optimal prediction loss $\mathbb{E}_{x,y} l(\theta^{\mathcal{M}}_{\min}, x, y, \mathcal{M})$. The difficulty for studying this questions is the underlying NP-hardness of solving robust optimization. Even for the simple models, say quadratic models, the robust optimization problem is NP-hard if the constraint set is polyhedral (Minoux, 2010).

To address this problem, drawing inspiration from the idea of influence function in robust statistics (Croux & Haesbroeck, 1999; Hampel; 1974; Huber & Ronchetti, 2009), which characterizes how the prediction loss changes when a small fraction of data points being contaminated, we propose the Adversarial Influence Function (AIF) to investigate the influence of robust optimization on prediction loss. Taking advantage of small perturbations, AIF has a closed-form expression and can be calculated efficiently. Moreover, AIF enables us to analyze the prediction error without implementing the robust optimization, which typically takes long time due to the computational burden of searching adversaries.

The rest of the paper is organized as follows. Section 2 lays out the setup and notation. Section 3 defines model sensitivity, which is used to understand how robust optimization affects the predictions. To efficiently approximate the

[1] John A. Paulson School of Engineering and Applied Sciences, Harvard University [2] Department of Statistics, Rutgers University. Correspondence to: Zhun Deng <zhundeng@g.harvard.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

model sensitivity, Section 4 introduces the AIF. Further, in Section 5, we show several case studies, by applying the proposed AIF to theoretically analyze the relationship between model sensitivity and model complexity and randomized smoothing. In Section 6, we extend the AIF theory to kernel regressions and distributional robust optimization.

1.1. Related work

Adversarial training and robust optimization Since (Goodfellow et al., 2014) proposed adversarial training, many innovative methods have been invented to improve the performance of adversarial training, such as (Agarwal et al., 2018; Liu & Hsieh, 2019; Shafahi et al., 2019; Yin et al., 2018). Earlier work only added adversarial examples in a few rounds during training, and many of them have been evaded by new attacks (Athalye et al., 2018). In (Madry et al., 2017), the authors proposed to use projected gradient ascent and obtain the state-of-art result. They further pointed out that the adversarial training can be formulated through the lens of robust optimization. Nevertheless, robust optimization has a very deep root in engineering (Taguchi & Phadke, 1989) , but many robust optimization problems are NP- hard(Minoux, 2010), and solving such problems heavily relies on high-speed computers and their exponentially increasing FLOPS-rates (Park et al., 2006). Our adversarial influence function may bridge the gap between theoretical analysis and engineering implementation of robust optimization to a certain degree, and improve our understanding of robust optimization.

Robust Statistics Robust statistics has been recently applied to machine learning and achieves impressive successes. (Koh & Liang, 2017) used the influence function to understand the prediction of a black-box model. (Debruyne et al., 2008; Liu et al., 2014) and (Christmann & Steinwart, 2004) used the influence function for model selections and cross-validations in kernel methods. Recently, (Bayaktar & Lai, 2018) extended the influence function to the adversarial setting, and investigated the adversarial robustness of multivariate M-Estimators. We remark here that their adversarial influence function is different from ours, where they focused on the influence on parameter inference, while ours focus on the influence of robust optimization on the prediction.

2. Setup and Notation

n this paper, we consider the task of mapping m-dimensional input $x \in \mathcal{X} \subseteq \mathbb{R}^m$ to a scalar output $y \in \mathcal{Y}$, with joint distribution $(x, y) \sim \mathbb{P}_{x,y}$ and marginal distributions $x \sim \mathbb{P}_x$, $y \sim \mathbb{P}_y$. We have training dataset $(X^t, Y^t) = \{(x_1^t, y_1^t), \cdots, (x_{n_t}^t, y_{n_t}^t)\}$ and evaluation dataset $(X^e, Y^e) = \{(x_1^e, y_1^e), \cdots, (x_{n_e}^e, y_{n_e}^e)\}$. For a given model architecture $\mathcal{M}$, the loss function is denoted as $l(\theta^{\mathcal{M}}, x, y, \mathcal{M})$ with parameter $\theta^{\mathcal{M}} \in \Theta \subseteq \mathbb{R}^d$ (we will omit $\mathcal{M}$ in l sometimes if not causing confusions). For robust optimization, we focus on studying the constraint set $\mathcal{R}(x, \varepsilon) = \{\omega \in \mathcal{X} : \|\omega - x\|_p \le \varepsilon \cdot \mathbb{E}_{x\sim\mathbb{P}_x}\|x\|_p\}$ **with small** ε, where $\|\cdot\|_p$ is the l_p norm. Such type of constraint set is also called l_p-attack in adversarial learning, which implies the adversaries are allowed to observe the whole dataset and are able to contaminate each data point x_i a little bit. This is commonly used in adversarial training for image classifications in machine learning and the constant factor $\mathbb{E}_{x\sim\mathbb{P}_x}\|x\|_p$ is for scale consideration.[1]

Further, we denote the empirical version of the minimizers for regular optimization and robust optimizers in Eq. (2):

$$\hat{\theta}_{\min}^{\mathcal{M}} := \arg\min_{\theta^{\mathcal{M}}\in\Theta} \frac{1}{n_t}\sum_{i=1}^{n_t} l(\theta^{\mathcal{M}}, x_i^t, y_i^t, \mathcal{M}),$$

$$\hat{\theta}_{\varepsilon,\min}^{\mathcal{M}} := \arg\min_{\theta^{\mathcal{M}}\in\Theta} \frac{1}{n_t}\sum_{i=1}^{n_t} \max_{\delta_i\in\hat{\mathcal{R}}(x_i^t,\varepsilon)} l(\theta^{\mathcal{M}}, x_i^t+\delta_i, y_i^t, \mathcal{M}),$$

where $\hat{\mathcal{R}}(x_i^t, \varepsilon) = \{u \in \mathcal{X} : \|u - x_i^t\|_p \le \varepsilon\hat{\mathbb{E}}_{x^t}\|x\|_p\}$, with $\hat{\mathbb{E}}_{x^t}$ being the expectation with respect to the empirical probability distribution of x^t.

We use $\text{sgn}(x)$ to denote the sign function: $\text{sgn}(x) = 1$ if $x > 0$, $\text{sgn}(x) = 0$ if $x = 0$, and -1 otherwise. We also use $[n]$ to denote the set $\{1, 2\cdots, n\}$. Further, we use the notion o_p and O_p, where for a sequence of random variables X_n, $X_n = o_p(a_n)$ means $X_n/a_n \to 0$ in probability, and $X_n = O_p(b_n)$ means that for any $\varepsilon > 0$, there is a constant K, such that $\mathbb{P}(|X_n| \le K \cdot b_n) \ge 1 - \varepsilon$.

3. Model Sensitivity

In order to quantify how robust optimization affects predictions, we first define the model sensitivity with respect to the robust optimization.

Definition 3.1 (ε-sensitivity/adversarial cost)**.** *For a given model $\mathcal{M}$, the ε-sensitivity/adversarial cost is defined as*

$$\mathcal{S}_\varepsilon(\mathcal{M}) := \mathbb{E}_{x,y} l(\theta_{\varepsilon,\min}^{\mathcal{M}}, x, y, \mathcal{M}) - \mathbb{E}_{x,y} l(\theta_{\min}^{\mathcal{M}}, x, y, \mathcal{M}).$$

The ε-sensitivity/adversarial cost quantifies how robust optimization increases the expected loss, and this loss also indicates the additional cost of *being adversarially robust*. Besides this straightforward interpretation, one can also interpret $\mathcal{S}_\varepsilon(\mathcal{M})$ as a trade-off between the prediction loss and robustness for model architecture $\mathcal{M}$ — the optimizer $\theta_{\varepsilon,\min}^{\mathcal{M}}$ is more adversarially robust but inflates the prediction loss comparing to $\theta_{\min}^{\mathcal{M}}$. For fixed ε, an architecture $\mathcal{M}$ with small ε-sensitivity implies that such an architecture

[1] For standard MNIST, the average l_2 norm of x is 9.21 with dimension $m = 28 \times 28$. The attack size does not have to be small, but ε, as the ratio of the magnitude of adversarial attacks and average magnitude of images, is small.

can achieve adversarial robustness by robust optimization without sacrificing the performance on the original data too much. We also say an architecture $\mathcal{M}$ with smaller ε-sensitivity is *more stable*.

Since $\theta^{\mathcal{M}}_{\min}$ is the minimizer of $\mathbb{E}_{x,y} l(\theta^{\mathcal{M}}, x, y, \mathcal{M})$ over $\theta^{\mathcal{M}}$, if we further have $\theta^{\mathcal{M}}_{\min} \in \Theta^{\circ}$, where Θ° denotes the interior of Θ and l is twice differentiable, by Taylor expansion, we would have

$$\mathcal{S}_{\varepsilon}(\mathcal{M}) = \frac{1}{2}(\Delta\theta^{\mathcal{M}}_{\varepsilon,\min})^T \mathbb{E}_{x,y} \nabla^2 l(\theta^{\mathcal{M}}_{\min}, x, y, \mathcal{M}) \Delta\theta^{\mathcal{M}}_{\varepsilon,\min} + o(\|\Delta\theta^{\mathcal{M}}_{\varepsilon,\min}\|_2^2),$$

where $\Delta\theta^{\mathcal{M}}_{\varepsilon,\min} = \theta^{\mathcal{M}}_{\varepsilon,\min} - \theta^{\mathcal{M}}_{\min}$, and the remainder is negligible if ε is small enough. Given the training set (X^t, Y^t) and the evaluation set (X^e, Y^e), we define the empirical ε-sensitivity:

$$\hat{\mathcal{S}}_{\varepsilon}(\mathcal{M}) \approx \frac{1}{2}(\Delta\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min})^T \mathbb{E}_{\hat{\mathbb{P}}_{x^e,y^e}} \nabla^2 l(\hat{\theta}^{\mathcal{M}}_{\min}, x, y, \mathcal{M}) \Delta\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min}, \tag{3}$$

by omitting the remainder $o(\|\Delta\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min}\|_2^2)$, where $\Delta\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min} = \hat{\theta}^{\mathcal{M}}_{\varepsilon,\min} - \hat{\theta}^{\mathcal{M}}_{\min}$. Notice that Eq. (3) involves $\Delta\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min}$, the solution of robust optimization, which, even for simple models with loss functions (such as linear regression with quadratic loss), does not have a closed-form expression and is computationally heavy to obtain. In the following sections, we will address this problem by introducing AIF, which provides an efficient way to approximate and analyze $\mathcal{S}_{\varepsilon}(\mathcal{M})$. For simplicity of illustration, we remove the superscripts t, e and use generic notation $(X, Y) = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ for general dataset in the following sections when there is no ambiguity.

4. Adversarial Influence Function

Unless explicitly stated, we mainly consider the case where the empirical risk $\sum_{i=1}^n l(\theta^{\mathcal{M}}, x_i^t, y_i^t; \mathcal{M})$ **is twice differentiable and strongly convex** in this paper. A relaxation of such conditions will be discussed in Section 4.1. In order to approximate $\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min} - \hat{\theta}^{\mathcal{M}}_{\min}$, for small ε, we use

$$\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min} - \hat{\theta}^{\mathcal{M}}_{\min} \approx \varepsilon^{\alpha} \cdot \frac{d(\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min} - \hat{\theta}^{\mathcal{M}}_{\min})}{d\varepsilon^{\alpha}}\Big|_{\varepsilon=0+} = \varepsilon^{\alpha} \cdot \frac{d\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min}}{d\varepsilon^{\alpha}}\Big|_{\varepsilon=0+}$$

for approximation, where $\alpha > 0$ is the smallest positive real number such that the limit $\lim_{\varepsilon \to 0+} (\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min} - \hat{\theta}^{\mathcal{M}}_{\min})/\varepsilon^{\alpha}$ is nonzero. Throughout this section, all the cases we consider later have $\alpha = 1$, while more general cases will be discussed in Section 6.2. Formally, we define the adversarial influence function as follows.

Definition 4.1 (Adversarial Influence Function)**.** *For a given model $\mathcal{M}$, the adversarial influence function (AIF) is defined as*

$$\mathcal{I}(\mathcal{M}) := \frac{d\theta^{\mathcal{M}}_{\varepsilon,\min}}{d\varepsilon}\Big|_{\varepsilon=0}. \tag{4}$$

The AIF measures the changing trend of the optimizer under robust optimization in the limiting sense. With the help of AIF, we then approximate $\mathcal{S}_{\varepsilon}(\mathcal{M})$ by

$$\mathcal{S}_{\varepsilon}(\mathcal{M}) \approx \frac{1}{2}\varepsilon^2 \mathcal{I}(\mathcal{M})^T \mathbb{E}_{x,y} \nabla^2 l(\theta^{\mathcal{M}}_{\min}, x, y, \mathcal{M}) \mathcal{I}(\mathcal{M})\big|_{\varepsilon=0}$$

when ε is small.

Next we provide a specific characterization of the empirical adversarial influence functions. We denote $\hat{I}(\mathcal{M}) = d\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min}/d\varepsilon|_{\varepsilon=0}$ as the empirical version of AIF. Besides, we denote the perturbation vector as $\Delta = (\delta_1^T, \cdots, \delta_n^T)^T$. Further, for given (X, Y) and $\mathcal{M}$, we define $g(\theta^{\mathcal{M}}, \Delta) = 1/n \sum_{i=1}^n l(\theta^{\mathcal{M}}, x_i + \delta_i, y_i; \mathcal{M})$ when we only consider the optimization over $(\theta^{\mathcal{M}}, \Delta)$.

Theorem 4.1. *Suppose $\mathcal{X}$, $\mathcal{Y}$ and Θ are compact spaces, the loss function $l(\theta, x, y)$ is three times continuously differentiable on $(\theta, x) \in \Theta \times \mathcal{X}$ for any given $y \in \mathcal{Y}$, and the empirical Hessian matrix $\hat{H}_{\hat{\theta}^{\mathcal{M}}_{\min}} = 1/n \sum_{i=1}^n \nabla_{\theta}^2 l(\hat{\theta}^{\mathcal{M}}_{\min}, x_i, y_i)$ is positive definite. Further, we assume the empirical risk $\sum_{i=1}^n l(\theta^{\mathcal{M}}, x_i^t, y_i^t; \mathcal{M})$ is twice differentiable and strongly convex and $g(\cdot, \Delta)$ is differentiable for every Δ, $\nabla_{\theta} g(\theta^{\mathcal{M}}, \Delta)$ is continuous on $\Theta \times \mathcal{X}$, $\hat{\theta}^{\mathcal{M}}_{\min}$ lies in the interior of Θ, and $\nabla_x l(\hat{\theta}^{\mathcal{M}}_{\min}, x_i, y_i, \mathcal{M}) \neq 0$ for all $i \in [n]$, then we have*

$$\hat{\mathcal{I}}(\mathcal{M}) = -\hat{H}^{-1}_{\hat{\theta}^{\mathcal{M}}_{\min}} \Phi, \tag{5}$$

where $\Phi = 1/n \sum_{i=1}^n \nabla_{x,\theta} l(\hat{\theta}^{\mathcal{M}}_{\min}, x_i, y_i) \mathbb{E}_{x \sim \hat{\mathbb{P}}_x} \|x\|_p \phi_i$ and $\phi_i = (\psi_1, \psi_2, \cdots, \psi_m)^T$, with

$$\psi_k = \frac{b_k^{q-1}}{(\sum_{k=1}^m b_k^q)^{\frac{1}{p}}} sgn\Big(\frac{\partial}{\partial x_{\cdot,k}} l(\hat{\theta}^{\mathcal{M}}_{\min}, x_i, y_i, \mathcal{M})\Big).$$

Here, we have $b_k = \big|\frac{\partial}{\partial x_{\cdot,k}} l(\hat{\theta}^{\mathcal{M}}_{\min}, x_i, y_i, \mathcal{M})\big|$, $x_{\cdot,k}$ is the k-th coordinate of vector x, for instance, $x_j = (x_{j,1}, x_{j,2}, \cdots, x_{j,m})^T$; $p \geq 0$ and $q \geq 0$ are conjugate such that $1/p + 1/q = 1$.

Remark 1. *The compactness condition is easy to satisfy. Since for any distributions $\mathcal{D}$ and integer n, we can take a sufficiently large constant $R > 0$, which is allowed to depend on n, such that all n samples are contained in the ball $\mathbb{B}(0, R)$ with high probability. Besides, if the input x is of high dimension, the computational bottleneck is mainly on inverting the empirical Hessian. We can use techniques such as conjugate gradients and stochastic estimation suggested in (Koh & Liang, 2017) to reduce the computational cost.*

The above theorem provides a closed-form expression for the first order AIF, and therefore a closed-form approximation of the model sensitivity $\mathcal{S}_{\varepsilon}(\mathcal{M})$. One nice property of such an approximation is that it does not depend on optimization algorithms, but only depends on the model $\mathcal{M}$ and the distribution of (x, y). This attribute makes model sensitivity an inherent property of model $\mathcal{M}$ and data distribution, making it a potential new rule for model selection. Model sensitivity can help us pick those models whose prediction result will not be greatly affected after robust optimization.

We show the effectiveness of approximation by AIF in Figure 1. We plot two error curves for $\Delta\hat{I}(n,\varepsilon) := \|(\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min} - \hat{\theta}^{\mathcal{M}}_{\min})/\varepsilon - \hat{\mathcal{I}}(\mathcal{M})\|_2$ and $\Delta\hat{S}(n,\varepsilon) := \|\hat{\mathcal{S}}_{\varepsilon}(\mathcal{M})/\varepsilon^2 - (\hat{\mathcal{I}}(\mathcal{M}))^T\mathbb{E}_{\hat{\mathbb{P}}_{x^e,y^e}}\nabla^2 l(\hat{\theta}^{\mathcal{M}}_{\min}, x, y, \mathcal{M})\hat{\mathcal{I}}(\mathcal{M})\|_2$, where the sample size is n. Theoretically, we expect $\Delta\hat{I}(n,\varepsilon)$ and $\Delta\hat{S}(n,\varepsilon)$ go to 0 as ε goes to 0. **In all the experiments in the paper, we use projected gradient descent (PGD) for robust optimization to obtain $\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min}$.** In Figure 1, we can see that as ε become smaller, $\Delta\hat{I}(n,\varepsilon)$ and $\Delta\hat{S}(n,\varepsilon)$ gradually go to 0. We remark here that we do not let ε be exactly 0 in our experiments, since PGD cannot obtain the exact optimal solutions for $\hat{\theta}^{\mathcal{M}}_{\min}$ and $\hat{\theta}^{\mathcal{M}}_{\varepsilon,\min}$. The existing system error will become dominating if ε is too small and return abnormally large value after divided by ε. This also motivates us to introduce the AIF to have an accurate approximation. The model we use is a linear regression model with 500 inputs drawn from a two-dimensional standard Gaussian, i.e. $x \sim \mathcal{N}(0, I)$. We fit y with $y = 2x_1 - 3.4x_2 + \eta$ and $\eta \sim 0.1 \cdot \mathcal{N}(0, I)$.

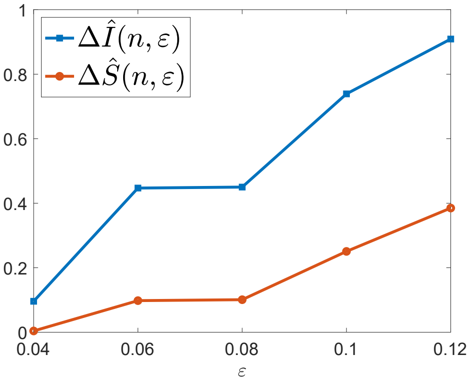


Figure 1. Effectiveness of AIF and model sensitivity for linear regression model. From the monotonicity relationship between ε and $\Delta\hat{I}(n,\varepsilon)$, $\Delta\hat{S}(n,\varepsilon)$, we verify the effectiveness of AIF and model sensitivity. Here, the sample size $n = 500$.

Remark 2. *It is straightforward to derive asymptotic normality for AIF by central limit theorem(Durrett, 2019), which can be used to construct confidence intervals for $\mathcal{I}(\mathcal{M})$. Specifically, if we denote $\zeta_i := -H^{-1}_{\hat{\theta}^{\mathcal{M}}_{\min}}\nabla_{\theta,x} l(\hat{\theta}^{\mathcal{M}}_{\min}, x_i, y_i)\mathbb{E}_{x\sim\hat{\mathbb{P}}_X}\|x\|_p\phi_i$, $\hat{\mu}_n := 1/n\sum_{i=1}^n \zeta_i$, and $\hat{\Sigma}_n := 1/n\sum_{i=1}^n(\zeta_i - \hat{\mu}_n)(\zeta_i - \hat{\mu}_n)^T$, then by classic statistical theory, we obtain*

$$\sqrt{n}\hat{\Sigma}_n^{-1/2}(\hat{\mathcal{I}}(\mathcal{M}) - \hat{\mu}_n) \overset{\mathcal{D}}{\to} \mathcal{N}(0, I_d),$$

as n goes to infinity, where $\mathcal{N}(0, I)$ denotes standard multivariate normal distribution and $\overset{\mathcal{D}}{\to}$ denotes convergence in distribution.

4.1. Non-convex, non-convergence cases

In the previous discussions, we talked about the case where the empirical loss is strongly convex. Now we briefly discuss about non-convex and non-convergence cases.

Well-separated condition. In the proof of Theorem 4.1, actually we only need $\hat{\theta}^{\mathcal{M}}_{\min}$ to be the global minimum and at the point $\hat{\theta}^{\mathcal{M}}_{\min}$, the empirical Hessian matrix is positive definite and the landscape are allowed to have many local minimums. The uniqueness assumption can also be formulated in a more elementary way: if we assume the smoothness of loss function l over $\mathcal{X} \times \Theta$, compactness of Θ and we only have one global minimum for $\mathbb{E}_{(x,y)\sim\mathbb{P}_{x,y}} l(\theta^{\mathcal{M}}, x, y, \mathcal{M})$ which lies in the interior of Θ, with positive definite Hessian matrix, and it is *well-separated*, which means that $\forall\omega > 0$, there exists $\kappa > 0$, such that $\forall\theta^{\mathcal{M}}$, if $\|\theta^{\mathcal{M}} - \theta^{\mathcal{M}}_{\min}\| > \omega$, we have

$$|\mathbb{E}_{x,y} l(\theta^{\mathcal{M}}, x, y, \mathcal{M}) - \mathbb{E}_{x,y} l(\theta^{\mathcal{M}}_{\min}, x, y, \mathcal{M})| > \kappa.$$

By classic statistical theory, $\hat{\theta}^{\mathcal{M}}_{\min}$ will be a global minimum if sample size is large enough.

The well-separated condition relaxes the convexity condition in Theorem 4.1. However, the validity of Theorem 4.1 still requires the condition that $\hat{\theta}^{\mathcal{M}}_{\min}$ is the global minimum of the empirical risk, which in practice, is hard to find. Another alternative relaxation is to use a surrogate loss.

Surrogate losses. In practice, we may obtain $\tilde{\theta}^{\mathcal{M}}_{\min}$ by running SGD with early stopping or on non-convex objectives, and get a solution $\hat{\theta}^{\mathcal{M}}_{\min}$ which is different from $\tilde{\theta}^{\mathcal{M}}_{\min}$. As in (Koh & Liang, 2017), we can form a convex quadratic approximation of the loss around $\tilde{\theta}^{\mathcal{M}}_{\min}$, i.e.,

$$\tilde{l}(\theta^{\mathcal{M}}, x, y) = l(\tilde{\theta}^{\mathcal{M}}_{\min}, x, y) + \nabla_\theta l(\tilde{\theta}^{\mathcal{M}}_{\min}, x, y)(\theta^{\mathcal{M}} - \tilde{\theta}^{\mathcal{M}}_{\min})$$
$$+ \frac{1}{2}(\theta^{\mathcal{M}} - \tilde{\theta}^{\mathcal{M}}_{\min})^T\left(\nabla^2_\theta l(\tilde{\theta}^{\mathcal{M}}_{\min}, x, y) + \lambda I\right)(\theta^{\mathcal{M}} - \tilde{\theta}^{\mathcal{M}}_{\min}),$$

where λ is a damping term to remove the negative eigenvalues of the Hessian. One can show the results of Theorem 4.1 hold with this surrogate loss.

5. Case studies of Adversarial Influence Functions

To illustrate the usage of adversarial influence functions, we use it to explore the relationship between model complexity, randomized smoothing and model sensitivity.

5.1. Model Complexity and Model Sensitivity

Throughout this paper, we use the term "model complexity" as a general term referring to 1) the number of features included in the predictive model, and 2) the model capacity, such as whether the model being linear, non-linear, and so on.

As observed in the prior literature (Fawzi et al., 2018; Kurakin et al., 2017; Madry et al., 2017), model complexity is closely related to adversarial robustness, that is, when the model capacity increases, the ε-sensitivity/adversarial cost will increase first and then decrease. However, such a phenomenon is only emporical and lack of theoretical justification. In this subsection, we will theoretically explore how the model complexity model affect the model sensitivity/adversarial cost by studying specific models with different model capacity and different number of features included in the predictive model.

5.1.1. Model Capacity and Model Sensitivity

We start with the relationship between model capacity and model sensitivity via two simple and commonly used models, with the dimension of inputs being fixed.

Linear regression models ($\mathcal{L}$) and quadratic models ($\mathcal{Q}$) We consider the class of linear models $\mathcal{L} = \{f_\beta(x) = \beta^T x : x, \beta \in \mathbb{R}^m\}$ and the class of quadratic models $\mathcal{Q} = \{f_{\beta,A}(x) = \beta^T x + x^T A x, x, \beta \in \mathbb{R}^m, A \in \mathbb{R}^{m \times m}\}$.

Apparently, the class of quadratic models has a larger model capacity and is more flexible than that of linear models. In the following theorem, we will show that larger model capacity does not necessarily lead to smaller sensitivity.

Theorem 5.1. *We fit the data (x_i, y_i) by $\mathcal{L}$ and $\mathcal{Q}$. For the simplicity of presentation, assume the sample sizes of both the training and testing sample are n. Suppose the underlying true generating process is $y = x^T\beta_1^* + (\beta_2^{*T}x)^2 + \xi$, where $x \sim \mathcal{N}(0, \sigma_x^2 I_m) \in \mathbb{R}^m$, $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$ and independent with x. For l_2 or l_∞ attack,*

I. when$(\|\beta_2^*\|_2^2\sigma_x^2 - \sqrt{\frac{2}{\pi}}\sigma_\xi)^2 > \frac{1+2m\sigma_x^2}{\max\{\sigma_x^2,1\}} \cdot \frac{2}{\pi}\sigma_\xi^2$, we have

$$\hat{\mathcal{S}}_\varepsilon(\mathcal{L}) > \hat{\mathcal{S}}_\varepsilon(\mathcal{Q}) + O_p(\varepsilon^2\sqrt{\frac{m^2}{n}});$$

II. when $(\|\beta_2^*\|_2^2\sigma_x^2 + \sqrt{\frac{2}{\pi}}\sigma_\xi)^2 < \frac{1}{\min\{1,\frac{3}{4}\sigma_x^2\}}(1 + m\sigma_x^2 - 2\sigma_x^2 \cdot \log m) \cdot \frac{3}{2\pi}\sigma_\varepsilon^2$, then

$$\hat{\mathcal{S}}_\varepsilon(\mathcal{L}) < \hat{\mathcal{S}}_\varepsilon(\mathcal{Q}) + O_p(\varepsilon^2\sqrt{\frac{m^2}{n}}).$$

From Theorem 5.1, unlike adversarial robustness, we can see that the model sensitivity does not have monotonic relationship with the model capacity. Such a monotonic relationship only holds when the model has high complexity (when $\|\beta_2^*\|$ is large). Therefore, when n is sufficiently large, the result implies that a larger model capacity does not necessarily lead to a model with smaller sensitivity.

5.1.2. Number of features and model sensitivity

Another important aspect of model complexity is the number of features included in the predictive model. There have been many model selection techniques, such as LASSO, AIC and BIC, developed over years. Given the newly introduced concept of model sensitivity, it is interesting to take model sensitivity into consideration during model selection. For example, if for a specific model, including more features results in a smaller model sensitivity, then for the sake of adversarial robustness, we should include more features even if it leads to feature redundancy.

For instance, the following results study when x_i follows some structures such as $Cov(x_i) = \sigma_x^2 I_m$ for some constant σ_x, the relationship between model sensitivity and number of features included in linear models.

Theorem 5.2. *Suppose that the data (x_i, y_i)'s are i.i.d. samples drawn from a joint distribution $P_{x,y}$. Denote the sample sizes of the training and testing sample by n_{train} and n_{test} respectively. Let m be the dimension of input x, and*

$$\beta_{\min}^{\mathcal{L}} = \arg\min_\beta \mathbb{E}_{P_{x,y}}(y - \beta^T x)^2.$$

Define $\eta_i^{\mathcal{L}} = y_i - \beta_{\min}^{\mathcal{L}\top} x_i$, and assume $\mathbb{E}[x_i \cdot \mathrm{sgn}(\eta_i^{\mathcal{L}})] = 0$ and $Cov(x_i) = \sigma_x^2 I_m$, then for ℓ_2 attack

$$\hat{S}_\varepsilon(\mathcal{L}) = \varepsilon^2(\mathbb{E}_{x \sim \hat{P}_x}\|x\|_2)^2 \cdot (\mathbb{E}|\eta_i^{\mathcal{L}}|)^2 \cdot \sigma_x^{-2} + O_p(\varepsilon^2 \cdot \sqrt{\frac{1}{n_{train}} + \frac{m}{n_{test}}}).$$

Given this theorem, we now consider a specific case where we apply this result to random effect model.

Corollary 5.1. *Consider the random effect model $y = \beta^\top x + \xi$, where $x \in \mathbb{R}^M$, $\beta_1, ..., \beta_M \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, and $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$. Further, we assume x is a random design with distribution $x_1, ..., x_n \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_x^2 I_M)$. Then when we only include m features in the linear predictive model, the resulting model sensitivity is*

$$\hat{S}_\varepsilon(\mathcal{L}) = \frac{4\varepsilon^2}{\pi\sigma_x^2}\frac{\Gamma^2(\frac{m+1}{2})}{\Gamma^2(\frac{m}{2})} \cdot ((M-m)\sigma_x^2 + \sigma_\xi^2) + O_p(\varepsilon^2 \cdot \sqrt{\frac{1}{n_{train}} + \frac{m}{n_{test}}}),$$

where $\Gamma(\cdot)$ is the Gamma function such that $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\,dt$.

Since $\frac{\Gamma^2(\frac{m+1}{2})}{\Gamma^2(\frac{m}{2})} \asymp \frac{1}{2}m$, there is a universal constant C, such that $\hat{S}_\varepsilon(\mathcal{L}) \asymp Cm((M-m)\sigma_x^2 + \sigma_\xi^2) = -C\sigma_x^2 m^2 + C(M + \sigma_\xi^2)m$. This also implies that a larger model capacity does not necessarily lead to a model with smaller sensitivity. Specifically, when m is small, including more features in the linear model results in larger model sensitivity. In contrast, when m is large, i.e. in the high-complexity regime, including more features leads to smaller model sensitivity.

Next, we consider a broader class of functions — general regression models.

General regression models ($\mathcal{GL}$) In general regression models, suppose we use a d-dimensional basis $v^{\mathcal{GL}}(x) = (v_1^{\mathcal{GL}}(x), ..., v_d^{\mathcal{GL}}(x))^T \in \mathbb{R}^d$ to approximate y (d can be a function of m), and get the coefficients by solving

$$\hat{\theta}^{\mathcal{GL}}_{\min} = \arg\min_{\theta} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \theta^T v^{\mathcal{GL}}(x_i))^2,$$

where the loss function is $l(\theta, x_i, y_i, \mathcal{GL}) = \frac{1}{2}(y_i - \theta^T v^{\mathcal{GL}}(x_i))^2$. By Theorem 4.1, it is straightforward to obtain

$$\hat{\mathcal{I}}(\mathcal{GL}) = -\hat{H}^{-1}_{\hat{\theta}^{\mathcal{GL}}_{\min}}\Phi = -Cov(v^{\mathcal{GL}}(x))^{-1}\Phi + O_P(\sqrt{\frac{d}{n}}),$$

where $Cov(v^{\mathcal{GL}}(x))$ is the covariance matrix of $v^{\mathcal{GL}}(x)$ and

$$\Phi = \sum_{i=1}^{n}\Big[\frac{|(\hat{\theta}^{\mathcal{GL}}_{\min})^T v^{\mathcal{GL}}(x_i) - y_i|}{n\|(\hat{\theta}^{\mathcal{GL}}_{\min})^T \frac{\partial v^{\mathcal{GL}}(x_i)}{\partial x}\|}\frac{\partial v^{\mathcal{GL}}(x_i)}{\partial x}(\frac{\partial v^{\mathcal{GL}}(x_i)}{\partial x})^T \hat{\theta}^{\mathcal{GL}}_{\min} + \frac{v^{\mathcal{GL}}(x_i)}{n}\|(\hat{\theta}^{\mathcal{GL}}_{\min})^T\frac{\partial v^{\mathcal{GL}}(x_i)}{\partial x}\| \operatorname{sgn}((\hat{\theta}^{\mathcal{GL}}_{\min})^T v^{\mathcal{GL}}(x_i) - y_i)\Big].$$

Thus,

$$\hat{\mathcal{S}}_\varepsilon(\mathcal{GL}) = \varepsilon^2 \cdot \Phi^\top Cov(v(x))^{-1}\Phi + O_P(\varepsilon^2\sqrt{\frac{d}{n}}). \quad (6)$$

Notice that the linear regression model is a special case if we take $v(x) = x$. However, the expression of model sensitivity for the general regression models is very complex and hard to analyze directly most of the time. Instead of directly studying Eq. (6), we further simplify the expression by providing an upper bound to shed some light.

Theorem 5.3. *Suppose that the data (x_i, y_i)'s are i.i.d. samples drawn from a joint distribution $P_{x,y}$. Let m be the dimension of input x, and*

$$\theta^{\mathcal{GL}}_{\min} = \arg\min_{\theta}\mathbb{E}_{P_{x,y}}(y - \theta^T v^{\mathcal{GL}}(x_i))^2.$$

Let $\eta_i^{\mathcal{GL}} = y_i - (\theta^{\mathcal{GL}}_{\min})^T v^{\mathcal{GL}}(x_i)$ and assume $\mathbb{E}[x_i \cdot \operatorname{sgn}(\eta_i^{\mathcal{GL}})] = 0$, then

$$\hat{\mathcal{S}}_\varepsilon(\mathcal{GL}) \le \varepsilon^2(\mathbb{E}_{x\sim\hat{P}_x}\|x\|_2)^2 \cdot \frac{1}{\lambda_{\min}(E[v(x_i)v(x_i)^\top])} \cdot \mathbb{E}[\|(\frac{\partial}{\partial x}v^{\mathcal{GL}}(x_i))^T\frac{\partial}{\partial x}v^{\mathcal{GL}}(x_i)\|_2] \cdot \mathbb{E}[|\eta_i^{\mathcal{GL}}|]^2 + O_p(\varepsilon^2\sqrt{\frac{d}{n}}).$$

The following example illustrates how our upper bound is used to demonstrate the trend of change between model sensitivity and number of features included.

Example 5.1. *Suppose $v(x) = (x^T, (\frac{x}{2}\odot\frac{x}{2})^T)^T$. If x consists of random features, such that each coordinate of x is i.i.d drawn from uniform distribution on $(-1, 1)$. $y = x^T\beta_1^* + \beta_2^{*T}\frac{x}{2}\odot\frac{x}{2} + \xi$, where $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$ and independent with x. As a result, the eigenvalue satisfies*

$$\lambda_{\min}\mathbb{E}[v^{\mathcal{GL}}(x_i)v^{\mathcal{GL}}(x_i)^\top] \ge \frac{1}{5};$$

$$\mathbb{E}[\|(\frac{\partial}{\partial x}v^{\mathcal{GL}}(x_i))^T\frac{\partial}{\partial x}v^{\mathcal{GL}}(x_i)\|_2] = 1,$$

regardless of the number of features m. Besides, $\mathbb{E}|\eta_i^{\mathcal{GL}}|$ decreases as m increases, and thus the upper bound will decrease as m increases.

In the experiments in Figure 2(a), we show the trend for $\hat{\mathcal{S}}_\varepsilon(\mathcal{GL})$ by taking sample size $n = 5000$, $\sigma_\xi = 0.1$. We take the average result for 1000 repetitions.

5.2. Randomized Smoothing and Model Sensitivity

As the last case study of AIF, we investigate the effect of randomized smoothing (Cohen et al., 2019), a technique inspired by differential privacy, in adversarial robustness. Randomized smoothing has achieved impressive empirical success as a defense mechanism of adversarial attacks for l_2 attack. The core techniques is adding isotropic noise $\vartheta \sim \mathcal{N}(0, \sigma_r^2 I)$ to the inputs so that for any output range O,

$$\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^{n} l(\theta^{\mathcal{M}}, x_i + \vartheta_i, y_i, \mathcal{M}) \in O\Big)$$

is close to

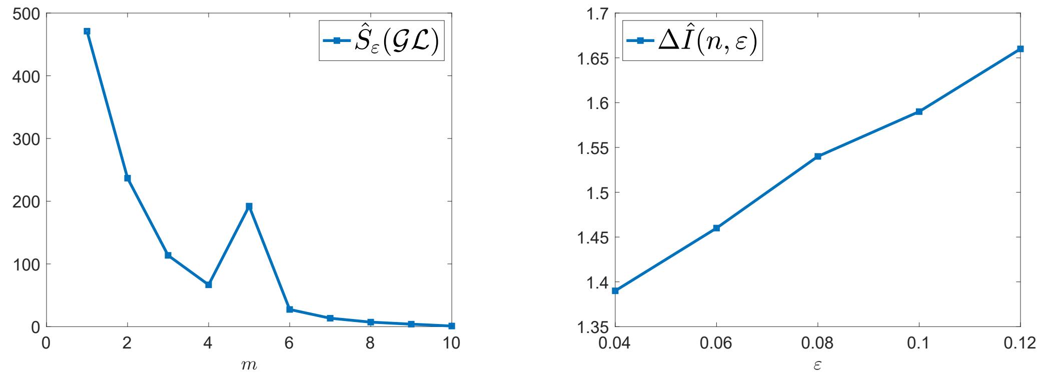
$$\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^{n} l(\theta^{\mathcal{M}}, x_i + \delta_i + \vartheta_i, y_i, \mathcal{M}) \in O\Big)$$

for constrained $\|\delta_i\|_2$.

The following theorem provides an insight into how randomized smoothing affects model sensitivity regarding linear regression models.

Theorem 5.4. *Use the same notation as that in Theorem 5.2. Suppose that the data (x_i, y_i)'s are i.i.d. samples drawn from a joint distribution $P_{x,y}$, and $\mathbb{E}[x_i \cdot \operatorname{sgn}(\eta_i^{\mathcal{L}})] = 0$, $Cov(x_i) = \sigma_x^2 I_m$, and $Var(\eta_i^{\mathcal{L}}) = \sigma_{\eta^2}$. When we fit y with $\tilde{x} = x + \vartheta$, where ϑ is distributed as $N(0, \sigma_r^2 I_m)$, then*

$$\frac{\hat{\mathcal{S}}_\varepsilon(\mathcal{L}_{noise})}{\hat{\mathcal{S}}_\varepsilon(\mathcal{L})} = \frac{\sigma_x^2/\sigma_\xi^2}{\sigma_x^2 + \sigma_r^2}\left(\frac{2\sigma_r^2\sigma_x^2}{\sigma_x^2 + \sigma_r^2}\|\beta^{\mathcal{L}}_{\min}\|_2^2 + \sigma_\xi^2\right) + O_p(\sqrt{\frac{m}{n}}).$$



(a) Illustration of the relationship between the feature number and model sensitivity for the model in Example 5.1.

(b) Effectiveness of AIF for kernel regression with NTK on MNIST.

Figure 2. a) Experimentally, the general trend for $\hat{\mathcal{S}}_\varepsilon(\mathcal{GL})$ with respect to m is decreasing (though not strict for every m) as the upper bound suggests. b) The monotonic trend of ε is still clearly observed, though thevalues are larger than the previous example in Figure 1 due to the high dimensionality of MNIST.

Here, $\mathcal{L}_{\text{noise}}$ denotes the linear model with randomized smoothing by adding input noise. This theorem informs us that when σ_r is large, we have $\hat{\mathcal{S}}_\varepsilon(\mathcal{L}_{\text{noise}}) \leq \hat{\mathcal{S}}_\varepsilon(\mathcal{L})$ asymptotically, and $\hat{\mathcal{S}}_\varepsilon(\mathcal{L}_{\text{noise}})$ becomes smaller with larger σ_r. In other words, the randomized smoothing helps reduce the sensitivity in this case.

6. Further Extensions

In this section, we extend the theories of IFA to kernel regressions and distributional robust optimization. First, we derive the AIF for kernel regressions in Section 6.1. In particular, we are interested in how well AIF characterizes the change of optimizers with neural tangent kernels (NTK), whose equivalence to infinitely wide neural networks has been well-established in recent literatures (Du et al., 2018; Jacot et al., 2018). In Section 6.2, we further extend our theory to compute the AIF for distributional robust optimization.

6.1. AIF of the kernel regressions

We consider the kernel regression model in the following form

$$\hat{L}_n(\theta, X, Y) := \frac{1}{n}\sum_{i=1}^n \big(y_i - \sum_{j=1}^n K(x_i, x_j)\theta_j\big)^2 + \lambda\|\theta\|_2^2. \tag{7}$$

where $\theta = (\theta_1, \cdots, \theta_n)^T$, and $\lambda > 0$. Now let us denote $g(\theta, \Delta) = \hat{L}_n(\theta, X + \Delta, Y)$, and we will calculate the empirical adversarial influence function $\hat{\mathcal{I}}(\mathcal{K})$ for kernel K.

Notice that in kernel regression, the loss function $\big(y_i - \sum_{j=1}^n K(x_i, x_j)\theta_j\big)^2$ includes all the data points in one single term, which is different from the summation-form of loss function in Theorem 4.1. Fortunately, the technique of proving Theorem 4.1 can still be used here with slight modification. We obtain the following corollary for the adversarial influence function $\hat{\mathcal{I}}(\mathcal{K})$ in kernel regression.

Corollary 6.1. *Suppose $\mathcal{X}$, $\mathcal{Y}$ and Θ are compact spaces, the kernel $\hat{L}_n$ is three times continuously differentiable on $\Theta\times\mathcal{X}$. $g(\cdot, \Delta)$ is differentiable for every Δ and $\nabla_\theta g(\theta, \Delta)$ s continuous on $\Theta \times \mathcal{X}$, the minimizer $\hat{\theta}_{\min}$ lies in the interior of Θ, with non-zero $\nabla_{x_i}\hat{L}_n(\hat{\theta}_{\min}, X, Y)$ for all $i \in [n]$, then we have*

$$\hat{\mathcal{I}}(\mathcal{K}) = -\big(\sum_{i=1}^n K(x_i)K(x_i)^T + n\lambda I\big)^{-1}$$

$$\Big(\sum_{k,i=1}^n \big(K(x_i)^T\hat{\theta}_{\min} + K(x_i)\hat{\theta}_{\min}^T - y_i\big)\mathcal{K}_{x_i,x_k}\beta_k\Big).$$

In the above formula,

$$K(x_i) = \big(K(x_i, x_1), K(x_i, x_2), \cdots, K(x_i, x_n)\big)^T,$$

$$\mathcal{K}_{x_i,x_k} = \Big(\frac{\partial K(x_i, x_1)}{\partial x_k}, \cdots, \frac{\partial K(x_i, x_n)}{\partial x_k}\Big)^T.$$

And the z-th coordinate of β_k is

$$\beta_{k,z} = \frac{c_z^{q-1}}{(\sum_{k=1}^m c_z^q)^{\frac{1}{p}}} sgn\Big(\nabla_{x_k}\hat{L}_n(\hat{\theta}_{\min}, z)\Big)\mathbb{E}_{x\sim\hat{\mathbb{P}}_x}\|x\|_p,$$

with $c_z = |\nabla_{x_k}\hat{L}_n(\hat{\theta}_{\min}, z)|$, where $\nabla_{x_k}\hat{L}_n(\hat{\theta}_{\min}, z)$ is short for the z-th coordinate of $\nabla_{x_k}\hat{L}_n(\hat{\theta}_{\min}, X, Y)$:

$$\nabla_{x_k}\hat{L}_n(\theta, X, Y) = \frac{2}{n}\sum_{i=1}^n \Big(K(x_i)^T\hat{\theta}_{\min} - y_i\Big)\mathcal{K}_{x_i,x_k}^T\hat{\theta}_{\min}.$$

Neural tangent kernels The intimate connection between kernel regression and overparametrized two-layer neural networks has been studied in the literature, see (Du et al., 2018; Jacot et al., 2018). In this section, we are going to apply Corollary 6.1 to the two-layer neural networks in the over-parametrized setting.

Specifically, we consider a two-layer ReLU activated neural network with b neurons in the hidden layer:

$$f_{W,a}(x) = \frac{1}{\sqrt{b}} \sum_{r=1}^{b} a_r \sigma(w_r^T x),$$

where $x \in \mathbb{R}^m$ denotes the input, $w_1, ..., w_b \in \mathbb{R}^m$ are weight vectors in the first layer, $a_1, ..., a_b \in \mathbb{R}$ are weights in the second layer. Further we denote $W = (w_1, ..., w_b) \in \mathbb{R}^{m \times b}$ and $a = (a_1, ..., a_m)^T \in \mathbb{R}^m$.

Suppose we have n samples $S = \{(x_i, y_i)\}_{i=1}^n$ and assume $\|x_i\|_2 = 1$ for simplicity. We train the neural network by randomly initialized gradient descent on the quadratic loss over data S. In particular, we initialize the parameters randomly: $w_r \sim N(0, \kappa^2 I)$, $a_r \sim U(-1, 1)$, for all $r \in [m]$, then Jacot et al. [2018] showed that, such a resulting network converges to the solution produced by the kernel regression with the so called Neural Tangent Kernel (NTK) matrix:

$$NTK = \left[\frac{x_i^\top x_j(\pi - \arccos(x_i^\top x_j))}{2\pi}\right]_{i,j\in[n]}.$$

In Figure 2(b), we experimentally demonstrate the effectiveness of the approximation of AIF in kernel regressions with neural tangent kernel on MNIST. The estimation is based on the average of randomly drawn 300 examples from MNIST for 10 times.

6.2. Distributional adversarial influence function

Another popular way to formulate adversarial attack is through distributional robust optimization (DRO), where instead of perturbing x with certain distance, one perturbs (x, y) in a distributional sense. For a model $\mathcal{M}$, the corresponding distributional robust optimization with respect to u-Wasserstein distance W_u for $u \in [1, \infty)$ regarding l_p-norm is formulated as:

$$\min_{\theta^{\mathcal{M}}} OPT(\varepsilon; \theta^{\mathcal{M}}),$$

where $OPT(\varepsilon; \theta^{\mathcal{M}})$ is defined as

$$OPT(\varepsilon; \theta^{\mathcal{M}}) := \max_{\tilde{\mathbb{P}}_{x,y}: W_u(\tilde{\mathbb{P}}_{x,y}, \mathbb{P}_{x,y}) \le \varepsilon} \mathbb{E}_{\tilde{\mathbb{P}}_{x,y}} l(\theta^{\mathcal{M}}, x, y; \mathcal{M}).$$

Here, $W_u(\mathcal{D}, \tilde{\mathcal{D}}) = \inf\{\int \|x - y\|_p^u d\gamma(x, y) : \gamma \in \Pi(\mathcal{D}, \tilde{\mathcal{D}})\}^{1/u}$ for two distributions $\mathcal{D}, \tilde{\mathcal{D}}$, and $\Pi(\mathcal{D}, \tilde{\mathcal{D}})$ are couplings of $\mathcal{D}, \tilde{\mathcal{D}}$. However, it is not clear whether

$$\theta^{\mathcal{M},DRO}_{\varepsilon,\min} := \arg\min_{\theta^{\mathcal{M}}} OPT(\varepsilon \mathbb{E}_{\hat{\mathbb{P}}_x} \|x\|_p; \theta^{\mathcal{M}}),$$

is well-defined since the optimizer may not be unique. Moreover, the corresponding sample version of the optimizer $\theta^{\mathcal{M},DRO}_{\varepsilon,\min}$ is not easy to obtain via regular optimization methods if we just replace the distribution $\mathbb{P}_{x,y}$ by its empirical distribution since it is hard to get the corresponding worst form of $\tilde{\mathbb{P}}_{x,y}$. As a result, we focus on defining empirical distributional adversarial influence function for a special approximation algorithm and state its limit. Interested readers are refered to the following result in (Staib & Jegelka, 2017) and (Gao & Kleywegt, 2016) to properly find an approximation for $\tilde{\mathbb{P}}_{x,y}$.

Lemma 6.1 (A variation of Corollary 2(iv) in (Gao & Kleywegt, 2016))**.** *Suppose for all y, $l(\theta^{\mathcal{M}}, x, y; \mathcal{M})$ is L-Lipschitz as a function of x. Define*

$$EMP(\varepsilon) := \max_{\delta_1, \cdots, \delta_n} \frac{1}{n} \sum_{i=1}^{n} l(\theta^{\mathcal{M}}, x_i + \delta_i, y_i, \mathcal{M}),$$

such that $(\sum_{i=1}^n \|\delta_i\|_p^u / n)^{1/u} \le \varepsilon$. Then, we have $EMP(\varepsilon) \ge OPT(\varepsilon; \theta^{\mathcal{M}}) - LD/n$ where D bounds the maximum deviation of a single point.

Lemma 6.1 provides a direction to define an **algorithm dependent** empirical DAIF $\hat{\mathcal{I}}^{DRO}(\mathcal{M})$. For a given model $\mathcal{M}$, the corresponding empirical distributional adversarial influence function is defined as

$$\hat{\mathcal{I}}^{DRO}(\mathcal{M}) := \frac{d\hat{\theta}^{\mathcal{M},DRO}_{\varepsilon,\min}}{d\varepsilon}\Big|_{\varepsilon=0+},$$

such that $\hat{\theta}^{\mathcal{M},DRO}_{\varepsilon,\min} \in \arg\min_{\theta^{\mathcal{M}} \in \Theta} EMP\Big(\varepsilon \mathbb{E}_{\hat{\mathbb{P}}_x} \|x\|_p\Big)$. We use $\in \arg\min$ here since there may not be a unique minimizer, but the limit $\hat{\mathcal{I}}^{DRO}(\mathcal{M})$ is still unique and well-defined. Similarly, we can provide a closed form of distributional adversarial influence function.

Theorem 6.1. *Under the settings of Theorem 4.1,*

$$\hat{\mathcal{I}}^{DRO}(\mathcal{M}) = -\hat{H}^{-1}_{\hat{\theta}^{\mathcal{M}}_{\min}} \varrho n^{\frac{1-u}{u}}, \tag{8}$$

where $\varrho = \nabla_{x,\theta} l(\hat{\theta}^{\mathcal{M}}_{\min}, x_J, y_J) \mathbb{E}_{\hat{\mathbb{P}}_x} \|x\|_p \phi_J$ and ϕ_i is defined as in Theorem 4.1, J is the index: $J = \arg\max_i \|\nabla_x L(\hat{\theta}^{\mathcal{M}}_{\min}, x_i, y_i)\|_q$.

We remark here from Eq. 8, we can see that if $u > 1$, more training data will result in a smaller norm of $\hat{\mathcal{I}}^{DRO}(\mathcal{M})$ since there is a factor $n^{(1-u)/u}$.

7. Conclusions and Future Work

To achieve adversarial robustness, robust optimization has been widely used in the training of deep neural networks, while their theoretical aspects are under-explored. In this work we first propose the AIF to quantify the influence of robust optimization theoretically. The proposed AIF is then used to efficiently approximate the model sensitivity, which is usually NP-hard to compute in practice. We then apply the AIF to study the relationship between model sensitivity and model complexity. Moreover, the AIF is applied to randomized smoothing and found that adding noise to the input during training would help reduce the model sensitivity. Further, the theories are extended to the kernel regression models and distributional robust optimization. Based on the newly introduced tool AIF, we suggest two main directions for future research.

First, we can study how to use AIF to select model with the greatest adversarial robustness. Due to the computational effectiveness of AIF, it is a natural idea to use AIF for model selection. Such an idea can be used for tuning parameter selection in statistical models such as high-dimensional regression and factor analysis, and further extended to the neural network depth and width selection.

Second, AIF can be extended to study more phenomena in adversarial training. For instance, the relationship between low-dimensional representations and adversarial robustness. Recently, Lezama et al. (2018); Sanyal et al. (2018) empirically observed that using learned low-dimensional representations as the input in neural networks is substantially more adversarially robust, but a theoretical exploration of this phenomenon is still lacking.

Acknowledgements

This work is in part supported by NSF award 1763665 and NSF DMS-2015378.

References

Agarwal, N., Gonen, A., and Hazan, E. Learning in non-convex games with an optimization oracle. *arXiv preprint arXiv:1810.07362*, 2018.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Bayaktar, E. and Lai, L. On the adversarial robustness of robust estimators. *arXiv preprint arXiv:1806.03801*, 2018.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. *Robust optimization*, volume 28. Princeton University Press, 2009.

Beyer, H.-G. and Sendhoff, B. Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33-34):3190–3218, 2007.

Christmann, A. and Steinwart, I. On robustness properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, 5(Aug):1007–1034, 2004.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

Croux, C. and Haesbroeck, G. Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. *Journal of Multivariate Analysis*, 71(2): 161–190, 1999.

Debruyne, M., Hubert, M., and Suykens, J. A. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9(Oct): 2377–2400, 2008.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Durrett, R. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Fawzi, A., Fawzi, O., and Frossard, P. Analysis of classifiers robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018.

Gao, R. and Kleywegt, A. J. Distributionally robust stochastic optimization with wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Hampel, F. R. Contributions to the theory of robust estimation. *Ph.D. Thesis*.

Hampel, F. R. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Huber, P. and Ronchetti, E. Robust statistics, john wiley & sons, inc, 2009.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1885–1894. JMLR. org, 2017.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. 2017.

Lezama, J., Qiu, Q., Musé, P., and Sapiro, G. Ole: Orthogonal low-rank embedding-a plug and play geometric loss for deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8109–8118, 2018.

Liu, X. and Hsieh, C.-J. Rob-gan: Generator, discriminator, and adversarial attacker. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11234–11243, 2019.

Liu, Y., Jiang, S., and Liao, S. Efficient approximation of cross-validation for kernel methods using bouligand influence function. In *International Conference on Machine Learning*, pp. 324–332, 2014.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Minoux, M. Robust network optimization under polyhedral demand uncertainty is np-hard. *Discrete Applied Mathematics*, 158(5):597–603, 2010.

Park, G.-J., Lee, T.-H., Lee, K. H., and Hwang, K.-H. Robust design: an overview. *AIAA journal*, 44(1):181–191, 2006.

Sanyal, A., Kanade, V., and Torr, P. H. Learning low-rank representations. *arXiv preprint arXiv:1804.07090*, 2018.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pp. 3353–3364, 2019.

Staib, M. and Jegelka, S. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS workshop on Machine Learning and Computer Security*, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Taguchi, G. and Phadke, M. S. Quality engineering through design optimization. In *Quality Control, Robust Design, and the Taguchi Method*, pp. 77–96. Springer, 1989.

Yin, D., Ramchandran, K., and Bartlett, P. Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*, 2018.