Adversarial Training with Generated Data in High-Dimensional Regression: An Asymptotic Study

Yue Xing

Abstract
In recent years, studies such as (Carmon et al., 2019; Gowal et al., 2021; Xing et al., 2022) have demonstrated that incorporating additional real or generated data with pseudo-labels can enhance adversarial training through a two-stage training approach. In this paper, we perform a theoretical analysis of the asymptotic behavior of this method in high-dimensional linear regression. While a double-descent phenomenon can be observed in ridgeless training, with an appropriate $L_2$ regularization, the two-stage adversarial training achieves a better performance. Finally, we derive a shortcut cross-validation formula specifically tailored for the two-stage training method.

1. Introduction
The development of machine learning and deep learning methods has led to breakthrough performance in various applications. However, recent studies, e.g., (Goodfellow et al., 2014), observe that these models are vulnerable when the data are perturbed by adversaries. Attacked inputs can be imperceptibly different from clean inputs to humans but can cause the model to make incorrect predictions.

To defend against adversarial attacks, adversarial training is a popular and promising way to improve the adversarial robustness of modern machine learning models. Adversarial training first generates attacked samples, then calculates the gradient of the model based on these augmented data. Such a procedure can make the model less susceptible to adversarial attacks in real-world situations.

There are fruitful results in the theoretical justification and methodology development in adversarial training. Among various research directions, one interesting aspect is to improve adversarial training with extra unlabeled data. Recent works successfully demonstrate great improvements in the adversarial robustness with additional unlabeled data. For example, (Xing et al., 2021), show that additional external real data help improve adversarial robustness; (Gowal et al., 2021; Wang et al., 2023) use synthetic data to improve the adversarial robustness and achieve the highest 65% to 70% adversarial testing accuracy for CIFAR-10 dataset under AutoAttack (AA) in (Croce et al., 2020)$^1$.

A recent study (Xing et al., 2022) reveals that adversarial training gains greater benefits from unlabeled data than clean (natural) training. The key observation is that adversarially robust models rely on the conditional distribution of the response given the features ($Y|X$) and the marginal distribution of the features ($X$). In contrast, clean training only depends on $Y|X$ in their study. As a result, adversarial training can benefit more than clean training from unlabeled data.

Besides adversarial training, high dimensional statistics is another important field of traditional machine learning to solve real-world problems from genomics, neuroscience to image processing. While many studies focus on obtaining a better performance via regularization, one surprising phenomenon in this field is the double descent phenomenon (Belkin et al., 2019; Hastie et al., 2019), which refers to a U-shaped curve in the test error as a function of the model complexity, together with a second descent phase occurring in the over-parameterized regime. This phenomenon challenges the conventional wisdom that increasing model complexity always leads to over-fitting. It provides significant implications for designing and analyzing machine learning algorithms in high-dimensional settings.

Given the substantial achievements in high-dimensional statistics, this paper aims to extend the analysis of (Xing et al., 2022) to a high-dimensional regression setup, in which both the data dimension $d$ and the sample size of the labeled data $n_1$ increase and $d/n_1 \to \gamma$ asymptotically. Although (Xing et al., 2022) provides a theoretical explanation for the benefits of unlabeled data in the large sample regime ($n_1 \gg d$), the asymptotic behavior of the two-stage method

---

$^1$Department of Statistics and Probability, Michigan State University, United States. Correspondence to: Yue Xing <xingyue1@msu.edu>.


$^1$https://robustbench.github.io
in other scenarios remains unclear.

Our contributions are summarized as follows:

- We derived the asymptotic convergence of the two-stage adversarial training when $d/n_1 \to \gamma$ for some constant $\gamma > 0$. (Section 3.1).

- It is observed that a proper ridge penalty in the clean training stage benefits the two-stage method. However, the optimal ridge penalty for the clean estimate in the first stage of (Xing et al., 2022) differs from the one yielding the best clean performance. We conjecture that this discrepancy arises from the change in the error decomposition from clean training to two-stage adversarial training. To facilitate more efficient hyperparameter tuning, we propose adaptations to existing cross validation (CV) methods, improving the time-consuming vanilla CV approach (Sections 3.2 and 3.3).

1.1. Related Works

Below is a summary of related works in adversarial training, high-dimensional statistics, and cross validation.

**Adversarial Training.** There are many studies in the area of adversarial training. Some studies, e.g., (Goodfellow et al., 2014; Zhang et al., 2019; Wang et al., 2019b; Cai et al., 2018; Zhang et al., 2020a; Carmon et al., 2019; Gao et al., 2019; Zhang et al., 2020b; Zhang and Li, 2023; Mianjy and Arora, 2019), work in methodology. Theoretical investigations have also been conducted from different perspectives. For instance, Chen et al. (2020); Javanmard et al. (2020); Taheri et al. (2021); Yin et al. (2018); Raghunathan et al. (2019); Najafi et al. (2019); Min et al. (2020); Hendrycks et al. (2019); Dan et al. (2020); Wu et al. (2020); Deng et al. (2021) study the statistical properties of adversarial training; Sinha et al. (2018); Wang et al. (2019a); Xiao et al. (2022) study the optimization perspective; Gao et al. (2019); Zhang et al. (2020b); Zhang and Li (2023); Mianjy and Arora (2022); Lv and Zhu (2021); Xiao et al. (2021) work on deep learning.

**Double Descent and High-Dimensional Statistics.** Double descent phenomenon is an observation in the learning curves of machine learning models. It describes the behavior of the generalization gap, i.e., the difference between the model performance on the training data and testing data. In a typical learning curve, the generalization error decreases and then increases with larger model complexity. However, in the double descent phenomenon, after the first decrease-increase pattern, the error decreases again when further enlarging the model complexity in the over-fitting regime. This non-monotonic behavior of the learning curve has been observed in various machine learning settings. Comprehensive investigations into the double descent phenomenon can be found in (Belkin et al., 2019; Hastie et al., 2019; Ba et al., 2020; d’Ascoli et al., 2020; Adlam and Pennington, 2020; Liu et al., 2021; Rocks and Mehta, 2022).

**Cross Validation.** Cross validation (CV) is a resampling procedure used to evaluate the performance of machine learning models. This paper mainly considers leave-one-out CV. For leave-one-out CV, it trains the model using all-but-one samples and repeats this process so that every sample is left in the estimation once. The final model performance is then averaged across all the models. The model can generalize better to new data by optimizing the hyperparameters in the model, e.g., regularization, through CV.

However, although a leave-one-out CV is an effective method for selecting hyperparameters, it is time-consuming by its design. Consequently, some studies propose shortcut formulas for the leave-one-out CV to reuse some terms when estimating the model using different data. Studies related to CV can be found in (Stone, 1978; Picard and Cook, 1984; Shao, 1993; Browne, 2000; Berrar, 2019).

2. Model Setup

In this section, we present the data generation model and the two-stage adversarial training framework.

**Data generation model.** We assume that the attributes $X \sim N(0, \Sigma)$ with covariance matrix $\Sigma = I_d$, and the response $Y$ satisfies $Y = X^\top \theta_0 + \varepsilon$ for $\| \theta_0 \| = r = O(1)$ and a Gaussian noise $\varepsilon$ with $\text{Var}(\varepsilon) = \sigma^2$.

**Two-stage adversarial training.** There are two stages in this training framework. In the first stage, we utilize $n_1$ i.i.d. labeled samples, i.e., $(x_i, y_i)$ for $i = 1, \ldots, n_1$. We consider the scenario where $d \asymp n_1$. The first stage solves the following clean training problem

$$\frac{1}{n_1} \sum_{i=1}^{n_1} (x_i^\top \theta - y_i)^2 + \lambda \| \theta \|^2$$

and obtain the clean estimate $\hat{\theta}_0(\lambda)$.

In the second stage, we use the trained model $\hat{\theta}_0(\lambda)$ to generate a pseudo response for a set of unlabeled data, i.e.,

$$\tilde{y}_i = x_i^\top \hat{\theta}_0(\lambda) + \varepsilon_i$$

for $i = n_1 + 1, \ldots, n_1 + n_2$. In this paper, we consider the scenario where $n_2 = \infty$. We also assume $\sigma^2$ is known and $\varepsilon_i$ are generated from $N(0, \sigma^2)$. Finally we use the extra data with pseudo response to do adversarial training and minimize the following loss w.r.t $\theta$:

$$\frac{1}{n_2} \sum_{i=n_1+1}^{n_1+n_2} \sup_{z \in B_2(x_i, \varepsilon)} (z^\top \theta - \tilde{y}_i)^2.$$
Denote the final solution as $\hat{\theta}_c(\lambda)$.

Remark 1. The two-stage method in this paper is slightly different from the original one in (Gowal et al., 2021; Xing et al., 2022). We only utilize the generated data in the second stage. This simplifies the theoretical analysis. In addition, when $d/n_1 = \gamma$ is a large constant, we empirically observe that the two-stage method is better than an adversarial training with only labeled data, i.e., the right of Figure 1.

Remark 2. Our initial trial indicates that adding additional regularization in equation (2.2) does not help much. Thus, we only inject a penalty in the clean training stage.

Expected Adversarial Risk

Under the model assumption of $(X, Y)$, the population adversarial risk for any given estimate $\theta$ becomes

$$R_e(\theta, \theta_0) = \|\theta - \theta_0\|^2_S + 2\sigma_0\epsilon \|\theta\| \sqrt{\|\theta - \theta_0\|^2_S + \sigma^2 + \epsilon^2\|\theta\|^2},$$

where $\|\cdot\|$ is the $L_2$ norm, and $\sigma_0 = \sqrt{2/\pi}$ is derived from the exact distribution of $(X, Y)$. We rewrite $R_e(\theta, \theta_0)$ as $R_e(\theta)$ for simplicity when no confusion arises.

Remark 3. One can denote $\theta_\epsilon = \arg\min_{\theta} R_e(\theta, \theta_0)$ as the best robust model. However, from $R_e(\theta, \theta_0)$, we are interested in $\|\theta - \theta_0\|^2_S$ and $\|\theta\|$ rather than $\|\theta - \theta_0\|$.

Based on (Xing et al., 2021), when an estimate $\theta \to \theta_\epsilon$, the excess adversarial risk $R_e(\theta, \theta_0) - R_e(\theta_\epsilon, \theta_0)$ can be approximated by a function of $\theta - \theta_\epsilon$. However, when $\theta - \theta_\epsilon$ diverges in the high-dimensional setup, such an approximation leads to a large error.

3. Analyzing the Two-Stage Adversarial Training Framework

This section presents the main theoretical results and simulation studies. We first demonstrate the main theory of the convergence of the two-stage method in Section 3.1, take different $\lambda$ under different attack strength $\epsilon$ in Section 3.2, and finally introduce a CV method in Section 3.3.

3.1. Convergence Result

For the two-stage adversarial framework, to study $\tilde{\theta}_\epsilon(\lambda)$, we denote the following function

$$m_\gamma(-\lambda) = -(1 - \gamma + \lambda) + \sqrt{(1 - \gamma + \lambda)^2 + 4\lambda \gamma},$$

which is used to describe the asymptotic behavior of $tr((\sum_{i=1}^{n_1} x_i x_i^\top + \lambda \mathbf{I}_d)^{-1})$ as in (Hastie et al., 2019).

After defining $m_\gamma$, one can obtain the convergence of $\tilde{\theta}_0(\lambda)$, and further figure out the asymptotic behavior of $\tilde{\theta}_\epsilon(\lambda)$. The convergence of the two-stage adversarial training framework is as follows:

Theorem 1 (Convergence of Two-Stage Adversarial Training). With probability tending to 1, $\tilde{\theta}_0(\lambda)$ satisfies

$$\|\tilde{\theta}_0(\lambda) - \theta_0\|^2 \to \lambda^2 r^2 m'_\gamma(-\lambda) + \sigma^2 \gamma (m_\gamma(-\lambda) - \lambda m'_\gamma(-\lambda)), \|\tilde{\theta}_0(\lambda)\|^2 \to r^2[1 - 2\lambda m_\gamma(-\lambda) + \lambda^2 m'_\gamma(-\lambda)] + \sigma^2 \gamma (m_\gamma(-\lambda) - \lambda m'_\gamma(-\lambda)).$$

For the two-stage adversarial estimate $\tilde{\theta}_\epsilon(\lambda)$, assuming $n_2 = \infty$, $\tilde{\theta}_\epsilon(\lambda)$ satisfies

$$\|\tilde{\theta}_\epsilon(\lambda) - \theta_\epsilon\|^2 \to \frac{1}{(1 + \alpha_\epsilon(\lambda))^2} \|\tilde{\theta}_0(\lambda)\|^2 + \frac{r^2 - \frac{2}{(1 + \alpha_\epsilon(\lambda))}}{(1 + \alpha_\epsilon(\lambda))^2} \|\tilde{\theta}_0(\lambda)\|^2, \|\tilde{\theta}_\epsilon(\lambda)\|^2 \to \frac{1}{(1 + \alpha_\epsilon(\lambda))^2} \|\tilde{\theta}_0(\lambda)\|^2,$$

where $2\tilde{\theta}_0(\lambda)\top \theta_0$ can be calculated via

$$2\tilde{\theta}_0(\lambda)\top \theta_0 = \|\theta_0\|^2 + \|\tilde{\theta}_0(\lambda)\|^2 - \|\tilde{\theta}_0(\lambda) - \theta_0\|^2,$$

and $\alpha_\epsilon(\lambda)$ is the solution of $\alpha$ in

$$\alpha + \epsilon \sigma_0 \frac{\alpha \|\tilde{\theta}_0(\lambda)\|}{\sqrt{\|\tilde{\theta}_0(\lambda)\|^2 \alpha^2 + \sigma^2 (1 + \alpha)^2}} = \epsilon \sigma_0 \sqrt{\|\tilde{\theta}_0(\lambda)\|^2 \alpha^2 + \sigma^2 (1 + \alpha)^2} + \epsilon^2.$$

The proof of Theorem 1 is in the appendix. We first study the convergence of $\tilde{\theta}_0(\lambda)$, and then evaluate $\tilde{\theta}_\epsilon(\lambda)$.

From Theorem 1, similar to $\tilde{\theta}_0$, one can see that $\|\tilde{\theta}_\epsilon(\lambda) - \theta_\epsilon\|^2$ and $\|\tilde{\theta}_\epsilon(\lambda)\|^2$ converges to some value as a function of $(\gamma, \lambda, \epsilon, \sigma^2)$ asymptotically.

We conduct a simulation to verify Theorem 1 and study the risk of the two-stage adversarial training. In the experiment, we take $n_1 = 100$ and $n_2 = \infty$, i.e., we directly use the population adversarial risk in the second stage. We change the data dimension $d$ to obtain different $\gamma = d/n_1$. The data follows $X \sim N(0, I_d)$, $Y = X\top \theta_0 + \epsilon$ with $\theta_0 \sim N(0, I_d/d)$ and $\epsilon \sim N(0, 1)$. The adversarial attack is taken as $\epsilon = 0.3$. We repeat the experiment 100 times to obtain the average performance. We use the excess adversarial risk, i.e., $R_e(\theta) - R_e(\theta_\epsilon)$ for $\theta \in \{\tilde{\theta}_0(\lambda), \tilde{\theta}_\epsilon(\lambda), \hat{\theta}_\epsilon(\lambda)\}$, to evaluate the performance of the three methods. The model $\hat{\theta}_\epsilon(\lambda)$ refers to the vanilla adversarial training as an additional benchmark, i.e., we conduct adversarial training using
Adversarial Training with Generated Data in High-Dimensional Regression: An Asymptotic Study

Figure 1. Simulation: Excess adversarial risk of clean training, vanilla adversarial training, and the two-stage adversarial training, without ridge penalty.

Figure 2. Theoretical value corresponding to Figure 1.

Figure 3. Simulation: Ridgeless regression and ridge regression with the best penalty in clean training and the two-stage adversarial training respectively. Adversarial training benefits more from a proper penalty.

Figure 4. Theoretical value corresponding to Figure 3.

In Figure 1, we take $\lambda \to 0$ to align with the experiments in the double descent literature. There are several observations from Figure 1. First, if we compare the performance of the two-stage adversarial training and the clean training, the two-stage adversarial training is better than clean training. Second, when $d/n_1$ gets larger, the performance of the two-stage adversarial training is better than the vanilla adversarial training, indicating that the information of the additional extra data matters. Finally, for all the three training methods, they all observe a double-descent phenomenon.

In addition, we plot the theoretical curves for the excess adversarial risk associating with the two-stage adversarial training. From Figure 2, the theoretical curve and the simulation result match with each other.

Finally, we examine how the ridge penalty affects the performance. In the simulation in Figure 3, we take $\epsilon = 0, 0.3$ and compare the performance when $\lambda = 0$ and $\lambda$ is taken to minimize the risk. In Figure 3, the y-axis is the corresponding excess adversarial risk, i.e., $\epsilon = 0, 0.3$ for the corresponding groups respectively. The corresponding theoretical curves can be found in Figure 4.

From Figure 3, one can see that the excess risk for the ridgeless regression is similar, while the two-stage adversarial training ($\epsilon = 0.3$) benefits more than clean training ($\epsilon = 0$) when taking a proper ridge penalty, which motivates us to further investigate in the penalty term in the following sections. In addition, the theoretical curves in Figure 4 align with the simulation results in 3 as well.

3.2. A Better Clean Estimate May Not Be Preferred

Different from ridgeless regression in the large-sample regime, with high-dimensional data, it is essential to utilize ridge penalty or other regularization to improve the testing performance. While one can adjust the penalty to control the performance of the clean estimate, we would like to ask:
To investigate how the optimal λ changes in the two-stage method, a simulation study is conducted in Figure 5. We take $n_1 = 50$. The data $X \sim N(0, I_d)$ and $d = 200$. The response $Y = \theta_0^T X + \varepsilon$ with $\theta_0 = 1/\sqrt{d}$ and $\varepsilon \sim N(0, 0.1^2)$. Besides the $n_1$ labeled data, we take $n_2 = \infty$. We repeat 30 times to get the average result and check the best $\lambda$ under different attack strength $\epsilon$.

From Figure 5, one can see that the optimal $\lambda$ gets larger when the attack strength gets larger. When $\epsilon = 0$, the optimal $\lambda$ is closed to zero. When $\epsilon = 0.3$, the best $\lambda$ is around 1, and 3 when $\epsilon = 0.5$, both of which are much larger than the case for $\epsilon = 0$.

### 3.3. Cross Validation

Observing that the optimal $\lambda$ for clean training is not the best for the two-stage adversarial training, we next investigate how to better select a proper $\lambda$.

We can always use the leave-one-out procedure for any estimate, it is time-consuming. As a result, existing literature, e.g. (Hastie et al., 2019), utilizes ways to approximate the leave-one-out CV procedure.

Recall that when $n_2 = \infty$, the second stage of the two-stage method minimizes

$$R_e(\theta, \hat{\theta}_0(\lambda)) = \|\theta - \hat{\theta}_0(\lambda)\|^2_\Sigma + \sigma^2 + \|\theta\|^2,$$

and the solution is

$$\hat{\theta}_e(\lambda) = (\Sigma + \alpha(\lambda) I_d)^{-1} \Sigma \hat{\theta}_0(\lambda),$$

for some $\alpha(\lambda) \geq 0$. One needs to rerun the CV procedure for $n_1$ times and obtain different $\hat{\theta}_e(\lambda)^{-1}$, the leave-one-out estimate of $\hat{\theta}_e(\lambda)$ leaving the $j$th labeled sample.

Given that the above formula $\hat{\theta}_e(\lambda)$ is a transformation of $\hat{\theta}_0(\lambda)$, one can borrow the idea of approximating CV in clean training to the two-stage adversarial training. To be specific, since both the $\alpha(\lambda)$ and $\hat{\theta}_0(\lambda)$ relate to each labeled sample, assuming the $j$th sample is discarded, the estimate of the two-stage method will be

$$\hat{\theta}_e(\lambda) = (\Sigma + \alpha^{-1} I_d) \Sigma \hat{\theta}_0^{-1}(\lambda),$$

and we approximate both $\alpha^{-1}$ and $\hat{\theta}_0^{-1}(\lambda)$.

The following lemma shows how to approximate $\alpha(\lambda)$ in the leave-one-out CV:

**Lemma 1.** Rewrite $\hat{\theta}_e(\lambda)$ as $\hat{\theta}_e(\hat{\theta}_0(\lambda), \hat{\theta}_0, \alpha = \alpha(\lambda)$ for simplicity. Denote $\Delta_1 = \hat{\theta}_0^{-1} - \hat{\theta}_0$, and

$$A_1 = \frac{1}{\|\theta - \hat{\theta}_0\|_\Sigma \|\theta\|} \langle \theta - \hat{\theta}_0 \rangle^T (\Sigma + \alpha I_d)^{-2} \Sigma \hat{\theta}_0,$$

$$- \frac{\|\hat{\theta}_0\|}{\|\theta - \hat{\theta}_0\|_\Sigma} \langle \theta - \hat{\theta}_0 \rangle^T \Sigma (\Sigma + \alpha I_d)^{-2} \Sigma \hat{\theta}_0,$$

$$A_2 = \frac{1}{\|\theta - \hat{\theta}_0\|_\Sigma \|\theta\|} \langle \theta - \hat{\theta}_0 \rangle^T \Sigma (\Sigma + \alpha I_d)^{-2} \Sigma \hat{\theta}_0,$$

$$- \frac{\|\theta - \hat{\theta}_0\|_\Sigma}{\|\theta\|^3} \langle \theta - \hat{\theta}_0 \rangle^T \Sigma (\Sigma + \alpha I_d)^{-2} \Sigma \hat{\theta}_0$$
The proof of Lemma 1 can be found in the appendix. Based on the result in Lemma 1, we can use

\[
\hat{\alpha}^{-j} - \alpha = \left( \frac{\epsilon_0 \alpha A_1 \Sigma (\hat{\theta} - \hat{\theta}_0) + \epsilon_0 A_2 \hat{\theta} + A_3}{\|\epsilon_0 A_1 \Sigma (\hat{\theta} - \hat{\theta}_0) + \epsilon_0 A_2 \hat{\theta} + A_3\|^2} \right) \times \left( \epsilon_0 A_4 \Delta_j \Sigma (\hat{\theta} - \hat{\theta}_0) + \epsilon_0 A_5 \Delta_j \hat{\theta} \right)
\]

to approximate \( \alpha^{-j} \).

In terms of the leave-out estimate of \( \hat{\theta}_0(\lambda) \), i.e., \( \hat{\theta}_0^{-j}(\lambda) \), one can use the Kailath Variant formual (from 3.1.2 of Petersen and Pedersen, 2008) and obtain

\[
\hat{\theta}_0(\lambda) - \hat{\theta}_0^{-j}(\lambda) = \frac{y_j - \hat{y}_j(\lambda)}{1 - S_j(\lambda)} (X^\top X + n\lambda I_d)^{-1} x_j,
\]

where \( X \in \mathbb{R}^{n \times d} \) denotes the labeled data matrix, and \( \hat{y}_j(\lambda) = \hat{\theta}_0(\lambda)^\top x_j \) as the fitted value of the \( j \)th observation.

After obtaining the estimate \( \hat{\alpha}^{-j} \) and \( \hat{\theta}_0^{-j}(\lambda) \), one can put them into (3.1) to obtain the leave-one-out estimate of \( \hat{\theta}_0(\lambda) \). The following theorem justifies the correctness of the above procedure:

**Theorem 2.** Denote

\[
CV(\lambda, \epsilon) = \frac{1}{n_1} \sum \left( |x_i^\top \hat{\theta}_0^{-j}(\lambda) - y_i| + \epsilon \| \hat{\theta}_0^{-j}(\lambda) \| \right)^2,
\]

and \( \hat{\theta}_0^{-j}(\lambda) = (\Sigma + \hat{\alpha}^{-j} I_d) \Sigma \hat{\theta}_0^{-j}(\lambda) \) as the approximation of the leave-one-out estimate using Lemma 1. Then under the Gaussian model assumption of \((X, Y)\), the approximated CV converges to the actual CV result, i.e.,

\[
\frac{1}{n_1} \sum \left( |x_i^\top \hat{\theta}_0^{-j}(\lambda) - y_i| + \epsilon \| \hat{\theta}_0^{-j}(\lambda) \| \right)^2 \overset{P}{\to} CV(\lambda, \epsilon).
\]

We use the simulation setting in Figure 5 to examine the performance of the above cross validation method. The results are summarized in Table 1.

From Table 1, there are two observations. First, one can see that using the cross validation, the CV loss in training is closed to the corresponding population risk.

In addition, the performance of the proposed algorithm is closed to the optimal \( \lambda \), and using clean regression in cross validation leads to a worse performance.

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross validation (CV loss in training)</td>
<td>0.8750</td>
<td>0.9663</td>
<td>1.0300</td>
</tr>
<tr>
<td>Cross validation (corresponding population risk)</td>
<td>0.8871</td>
<td>0.9751</td>
<td>1.0270</td>
</tr>
<tr>
<td>Cross validation for clean regression (corresponding population risk)</td>
<td>0.8873</td>
<td>1.0076</td>
<td>1.1140</td>
</tr>
<tr>
<td>Best ( \lambda ) (corresponding population risk)</td>
<td>0.8741</td>
<td>0.9648</td>
<td>1.0185</td>
</tr>
</tbody>
</table>

**Table 1.** Adversarial risks using cross validation and the best \( \lambda \).

4. Conclusion and Future Directions

This paper studies the asymptotics of the two-stage adversarial training in a high-dimensional linear regression setup. Double descent is observed for the ridge-less regression case, and a better performance can be achieved via \( \ell_2 \) regularization. We also derive the shortcut cross validation formula for this two-stage method to simplify the computation for cross validation.

The results in this paper can be extended in some directions. First, in literature, e.g., (Ba et al., 2020), the double descent phenomenon is also related to two-layer neural networks. An interesting future direction is to extend the analysis in this paper to the neural network setup. Second, since the shortcut formula for cross validation is distribution specific and assumes \( n_2 = \infty \), one may investigate in a more general cross validation procedure or relax to the scenario with a finite \( n_2 \).
References


A. Proofs

A.1. Theorem 1

Proof of Theorem 1. We first analyze \( \|\hat{\theta}_0(\lambda) - \theta_0\|^2 \) and \( \|\hat{\theta}_0(\lambda)\|^2 \).

For \( \|\hat{\theta}_0(\lambda)\|^2 \), denoting \( y \) and \( \varepsilon \) as the vector of response and noise, we have

\[
\|\hat{\theta}_0(\lambda)\|^2 = y^\top X (X^\top X + \lambda n I_d)^{-2} X^\top y
\]

\[
= \theta_0^\top X^\top X (X^\top X + \lambda n I_d)^{-2} X^\top \theta_0 + \varepsilon^\top X (X^\top X + \lambda n I_d)^{-2} X^\top \varepsilon + 2\theta_0^\top X^\top X (X^\top X + \lambda n I_d)^{-2} X^\top \varepsilon.
\]

We look at each term respectively. In probability, we have

\[
\theta_0^\top X^\top X (X^\top X + \lambda n I_d)^{-2} X^\top \theta_0 = r^2 - 2\lambda n \theta_0^\top (X^\top X + \lambda n I_d)^{-1} \theta_0 + \lambda^2 n^2 \theta_0^\top (X^\top X + \lambda n I_d)^{-2} \theta_0
\]

\[\to r^2 [1 - 2\lambda m_\gamma(-\lambda) + \lambda^2 m'_\gamma(-\lambda)],\]

and

\[
\varepsilon^\top X (X^\top X + \lambda n I_d)^{-2} X^\top \varepsilon \to \sigma^2 \left[ \frac{1}{n^2} tr((\Sigma + \lambda I_d)^{-1}) - \frac{1}{n^2} tr((\tilde{\Sigma} + \lambda I_d)^{-2}) \right]
\]

\[\to \sigma^2 \left[ \gamma m_\gamma(-\lambda) - \lambda \gamma m'_\gamma(-\lambda) \right],\]

where the function \( m_\gamma \) is obtained from (Hastie et al., 2019). For the cross term, we also have

\[
[\theta_0^\top X^\top X (X^\top X + \lambda n I_d)^{-2} X^\top \varepsilon]^2
\]

\[\to \sigma^2 tr [X^\top X (X^\top X + \lambda n I_d)^{-2} X^\top X (X^\top X + \lambda n I_d)^{-2} X^\top X \theta_0 \theta_0^\top]
\]

\[\overset{P}{\to} 0.\]

As a result,

\[
\|\hat{\theta}_0(\lambda)\|^2 \overset{P}{\to} r^2 [1 - 2\lambda m_\gamma(-\lambda) + \lambda^2 m'_\gamma(-\lambda)] + \sigma^2 \gamma \left[ m_\gamma(-\lambda) - \lambda m'_\gamma(-\lambda) \right].
\]

For \( \|\hat{\theta}_0(\lambda) - \theta_0\|^2 \), we have

\[
\|\hat{\theta}_0(\lambda) - \theta_0\|^2 = \|\hat{\theta}_0(\lambda)\|^2 + \|\theta_0\|^2 - 2\hat{\theta}_0(\lambda)^\top \theta_0,
\]

where in probability,

\[
[\varepsilon X (X^\top X + \lambda n I_d)^{-1} \theta_0]^2 \to 0,
\]

and

\[
\hat{\theta}_0(\lambda)^\top \theta_0 = \theta_0^\top X^\top X (X^\top X + \lambda n I_d)^{-1} \theta_0 + \varepsilon^\top X (X^\top X + \lambda n I_d)^{-1} \theta_0
\]

\[\to r^2 - \lambda n \theta_0^\top (X^\top X + \lambda n I_d)^{-1} \theta_0
\]

\[\to r^2 - \lambda r^2 m_\gamma(-\lambda).
\]

Consequently, in probability,

\[
\|\hat{\theta}_0(\lambda) - \theta_0\|^2 \to r^2 \lambda^2 m'_\gamma(-\lambda) + \sigma^2 \gamma [m_\gamma(-\lambda) - \lambda m'_\gamma(-\lambda)].
\]

For adversarial training, from (Javanmard et al., 2020; Xing et al., 2021) we know that the minimizer of \( R_e(\theta, \hat{\theta}_0(\lambda)) \) is

\[
\tilde{\theta}_e(\lambda) = (\Sigma + \alpha I_d)^{-1} \Sigma \hat{\theta}_0(\lambda),
\]
where \( c_0 = \sqrt{2/\pi} \) and \( \alpha \) satisfies

\[
\alpha \left( 1 + \epsilon c_0 \frac{\|\theta\|}{\sqrt{\|\theta - \hat{\theta}_0\|^2 + \sigma^2}} \right) = \left( c_0 \frac{\sqrt{\|\theta - \hat{\theta}_0\|^2 + \sigma^2}}{\|\theta\|} + \epsilon^2 \right).
\]

When \( \Sigma = I_d \), the above is reduced to

\[
\alpha + \epsilon c_0 \frac{\alpha \|\hat{\theta}_0\|}{\sqrt{\|\hat{\theta}_0\|^2 + \sigma^2(1 + \alpha)^2}} = \epsilon c_0 \frac{\sqrt{\|\theta - \hat{\theta}_0\|^2 + \sigma^2(1 + \alpha)^2}}{\|\theta\|} + \epsilon^2.
\]

Since \( \|\hat{\theta}_0(\lambda)\|^2 \) asymptotically converges to some fixed value, the solution of \( \alpha \) also asymptotically converges. \( \square \)

### A.2. Cross Validation

We present the proof of Lemma 1 and Theorem 2 in this section.

**Proof of Lemma 1.** To do cross validation, we know that \( \alpha \) satisfies

\[
\alpha \left( 1 + \epsilon c_0 \frac{\|\theta\|}{\sqrt{\|\theta - \hat{\theta}_0\|^2 + \sigma^2}} \right) = \left( c_0 \frac{\sqrt{\|\theta - \hat{\theta}_0\|^2 + \sigma^2}}{\|\theta\|} + \epsilon^2 \right).
\]

For the optimal solution in the adversarial training stage, we have

\[
0 = \nabla R_c(\theta, \hat{\theta}_0) = 2 \left[ \Sigma(\theta - \hat{\theta}_0) + \epsilon c_0 \frac{\|\theta - \hat{\theta}_0\|}{\|\theta\|} \Sigma \theta + \epsilon c_0 \frac{\|\theta\|}{\|\theta - \hat{\theta}_0\|} \Sigma(\theta - \hat{\theta}_0) + (\epsilon^2) \theta \right]
\]

\[
= 2 \left[ (I_d + \epsilon c_0 \frac{\|\theta\|}{\|\theta - \hat{\theta}_0\|} \Sigma(\theta - \hat{\theta}_0)) + \left( \epsilon c_0 \frac{\|\theta - \hat{\theta}_0\|}{\|\theta\|} + \epsilon^2 \right) \theta \right].
\]

For leave-one-out CV, we have

\[
0 = \nabla R_c(\theta, \hat{\theta}_0^{-j}) = 2 \left[ \Sigma(\theta - \hat{\theta}_0^{-j}) + \epsilon c_0 \frac{\|\theta - \hat{\theta}_0^{-j}\|}{\|\theta\|} \Sigma \theta + \epsilon c_0 \frac{\|\theta\|}{\|\theta - \hat{\theta}_0^{-j}\|} \Sigma(\theta - \hat{\theta}_0^{-j}) + (\epsilon^2) \theta \right]
\]

\[
= 2 \left[ (I_d + \epsilon c_0 \frac{\|\theta\|}{\|\theta - \hat{\theta}_0^{-j}\|} \Sigma(\theta - \hat{\theta}_0^{-j})) + \left( \epsilon c_0 \frac{\|\theta - \hat{\theta}_0^{-j}\|}{\|\theta\|} + \epsilon^2 \right) \theta \right].
\]

Consequently,

\[
(I_d + \epsilon c_0 \frac{\|\theta\|}{\|\theta - \hat{\theta}_0\|} \Sigma(\theta - \hat{\theta}_0)) + \left( \epsilon c_0 \frac{\|\theta - \hat{\theta}_0\|}{\|\theta\|} + \epsilon^2 \right) \theta
\]

\[
= (I_d + \epsilon c_0 \frac{\|\theta - \hat{\theta}_0\|}{\|\theta - \hat{\theta}_0^{-j}\|} \Sigma(\theta - \hat{\theta}_0^{-j})) + \left( \epsilon c_0 \frac{\|\theta - \hat{\theta}_0^{-j}\|}{\|\theta - \hat{\theta}_0^{-j}\|} + \epsilon^2 \right) \theta^{-j}.
\]

Denote

\[
\Delta_j = \hat{\theta}_0^{-j} - \hat{\theta}_0,
\]

and denote \( \alpha^{-j} \) as the best \( \alpha \) without \( j \)th sample. Then

\[
\tilde{\theta}^{-j} - \tilde{\theta} = (\Sigma + \alpha^{-j} I_d)^{-1} \Sigma \hat{\theta}_0^{-j} - (\Sigma + \alpha I_d)^{-1} \Sigma \hat{\theta}_0
\]

\[
= (\Sigma + \alpha I_d)^{-1} \Sigma \Delta_j - (\alpha^{-j} - \alpha) (\Sigma + \alpha I_d)^{-1} \tilde{\theta} + R_0.
\]
When \( \| \hat{\theta}_0 \| \) and \( \| \tilde{\theta} \| \) are away from zero,

\[
\| R_0 \| = O((\alpha^{-j} - \alpha)\|\Delta_j\|).
\]

As a result,

\[
\left( I_d + \epsilon \epsilon_0 \frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} \right) \Sigma (\tilde{\theta} - \hat{\theta}_0) + \left( \epsilon \epsilon_0 \frac{\| \tilde{\theta} - \hat{\theta}_0 \| \Sigma}{\| \tilde{\theta} \|} + \epsilon^2 \right) \tilde{\theta} - \left( I_d + \epsilon \epsilon_0 \frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} \right) \Sigma (\tilde{\theta} - \hat{\theta}_0) + \left( \epsilon \epsilon_0 \frac{\| \tilde{\theta} - \hat{\theta}_0 \| \Sigma}{\| \tilde{\theta} \|} + \epsilon^2 \right) \tilde{\theta} - \left( I_d + \epsilon \epsilon_0 \frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} \right) \Sigma (\tilde{\theta} - \hat{\theta}_0)
\]

and changing the order of the terms in the above, we have

\[
\left( I_d + \epsilon \epsilon_0 \frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} \right) \Sigma (\tilde{\theta} - \hat{\theta}_0) + \left( \epsilon \epsilon_0 \frac{\| \tilde{\theta} - \hat{\theta}_0 \| \Sigma}{\| \tilde{\theta} \|} + \epsilon^2 \right) \tilde{\theta} - \nabla R_0(\theta, \hat{\theta}_0)
\]

\[
= \left( I_d + \epsilon \epsilon_0 \frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} \right) \Sigma (\tilde{\theta} - \hat{\theta}_0) - \epsilon \epsilon_0 \frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} \Sigma (\tilde{\theta} - \hat{\theta}_0) - \epsilon \epsilon_0 \frac{\| \tilde{\theta} - \hat{\theta}_0 \| \Sigma}{\| \tilde{\theta} \|} + \epsilon^2 \left( \tilde{\theta} - \hat{\theta}_0 \right)
\]

for

\[
\| R_1 \| = O(\| \tilde{\theta} - \hat{\theta}_0 \| \| \Delta_j \|) = O(\| \Delta_j \|^2 + |\alpha^{-j} - \alpha| \| \Delta_j \|).
\]

We know that

\[
\frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} = \frac{\| \tilde{\theta} \|}{\| \theta - \theta_0 \| \Sigma} = \frac{1}{\| \theta - \theta_0 \| \Sigma} \| \tilde{\theta} \| \| \theta - \theta_0 \| \Sigma - (\tilde{\theta} - \hat{\theta}_0) \Sigma (\tilde{\theta} - \hat{\theta}_0) \| \tilde{\theta} \| + O(\| R_0 \|),
\]
Similarly, the first-order terms can be represented as

\[
\frac{1}{\|\theta - \theta_0\|\Sigma} \frac{\theta^\top (\theta_0^j - \theta)}{\|\theta\|} - \frac{(\theta - \theta_0)^\top \Sigma(\theta_0^j - \theta - \Delta_j)}{\|\theta - \theta_0\|\Sigma^3} \|\theta\|
\]

\[
= \frac{1}{\|\theta - \theta_0\|\Sigma} \frac{\theta^\top \left( \Sigma + \alpha I_d \right)^{-1} \Sigma \Delta_j - (\alpha^j - \alpha)(\Sigma + \alpha I_d)^{-2} \Sigma \theta_0}{\|\theta\|} - \frac{(\theta - \theta_0)^\top \Sigma (\Sigma + \alpha I_d)^{-1} \Sigma \Delta_j - (\alpha^j - \alpha)(\theta - \theta_0)^\top \Sigma (\Sigma + \alpha I_d)^{-2} \Sigma \theta_0 - (\theta - \theta_0)^\top \Sigma \Delta_j}{\|\theta - \theta_0\|\Sigma^3} \|\theta\| + O(\|R_0\|)
\]

\[
= \frac{1}{\|\theta - \theta_0\|\Sigma} \frac{\alpha \theta^\top (\Sigma + \alpha I_d)^{-2} \Sigma \theta_0}{\|\theta\|} - \frac{\|\theta\|}{\|\theta - \theta_0\|\Sigma^3} \alpha (\theta - \theta_0)^\top \Sigma (\Sigma + \alpha I_d)^{-2} \Sigma \theta_0
\]

\[
= A_1\alpha + \left( \frac{1}{\|\theta - \theta_0\|\Sigma} \frac{\theta^\top (\Sigma + \alpha I_d)^{-1} \Sigma}{\|\theta\|} + \frac{\|\theta\|}{\|\theta - \theta_0\|\Sigma^3} \alpha (\theta - \theta_0)^\top \Sigma (\Sigma + \alpha I_d)^{-1} \right) \Delta_j
\]

\[
- \alpha^j \left( \frac{1}{\|\theta - \theta_0\|\Sigma} \frac{\theta^\top (\Sigma + \alpha I_d)^{-2} \Sigma \theta_0}{\|\theta\|} - \frac{\|\theta\|}{\|\theta - \theta_0\|\Sigma^3} (\theta - \theta_0)^\top \Sigma (\Sigma + \alpha I_d)^{-2} \Sigma \theta_0 \right) + O(\|R_0\|).
\]
As a result,

\[
\begin{align*}
& \left( I_d + \epsilon c_0 \frac{\| \theta \|}{\| \theta - \theta_0 \|} \Sigma \right) (\theta^{-j} - \tilde{\theta} - \Delta_j) + \left( \epsilon c_0 \frac{\| \theta - \theta_0 \|}{\| \theta \|} \right) (\theta^{-j} - \tilde{\theta}) \\
& = -\frac{1}{2} \left( \epsilon c_0 \left( A_1 + A_2 \Delta_j - \alpha^{-j} A_3 \right) \Sigma(\tilde{\theta} - \theta_0) + \epsilon c_0 \left( A_4 + A_5 \Delta_j - \alpha^{-j} A_6 \right) \tilde{\theta} \right) + O(\|R_0\| + \|R_0\|) \\
& = \left( I_d + \epsilon c_0 \frac{\| \theta \|}{\| \theta - \theta_0 \|} \Sigma \right) \left( (\Sigma + \alpha I_d)^{-1} \Sigma \Delta_j - (\alpha^{-j} - \alpha)(\Sigma + \alpha I_d)^{-2} \Sigma \theta_0 - \Delta_j \right) \\
& + \left( \epsilon c_0 \frac{\| \theta - \theta_0 \|}{\| \theta \|} + \epsilon^2 \right) \left( (\Sigma + \alpha I_d)^{-1} \Sigma \Delta_j - (\alpha^{-j} - \alpha)(\Sigma + \alpha I_d)^{-2} \Sigma \theta_0 \right) \\
& = \left( -\alpha \left( I_d + \epsilon c_0 \frac{\| \theta \|}{\| \theta - \theta_0 \|} \Sigma \right) + \left( \epsilon c_0 \frac{\| \theta - \theta_0 \|}{\| \theta \|} + \epsilon^2 \right) \right) (\Sigma + \alpha I_d)^{-1} \Sigma \Delta_j \\
& + \alpha \left( I_d + \epsilon c_0 \frac{\| \theta \|}{\| \theta - \theta_0 \|} \Sigma + \left( \epsilon c_0 \frac{\| \theta - \theta_0 \|}{\| \theta \|} + \epsilon^2 \right) \right) (\Sigma + \alpha I_d)^{-1} \tilde{\theta} \\
& = -\alpha^{-j} \left( I_d + \epsilon c_0 \frac{\| \theta \|}{\| \theta - \theta_0 \|} \Sigma + \left( \epsilon c_0 \frac{\| \theta - \theta_0 \|}{\| \theta \|} + \epsilon^2 \right) \right) (\Sigma + \alpha I_d)^{-1} \tilde{\theta},
\end{align*}
\]

that is,

\[-\epsilon c_0 A_1 \Delta_j \Sigma (\tilde{\theta} - \theta_0) - \epsilon c_0 \alpha A_3 \Delta_j \tilde{\theta} = (\alpha^{-j} - \alpha) \left( \epsilon c_0 A_2 \Sigma (\tilde{\theta} - \theta_0) + \epsilon c_0 A_4 \tilde{\theta} - A_5 \right) + O(\|R_0\| + \|R_0\|),\]

and

\[
\alpha^{-j} - \alpha \approx \frac{\epsilon c_0 A_2 \Sigma (\tilde{\theta} - \theta_0) + \epsilon c_0 A_4 \tilde{\theta} + A_5}{\| \epsilon c_0 A_2 \Sigma (\tilde{\theta} - \theta_0) + \epsilon c_0 A_4 \tilde{\theta} + A_5 \|^2} \left( \epsilon c_0 A_1 \Delta_j \Sigma (\tilde{\theta} - \theta_0) + \epsilon c_0 A_3 \Delta_j \tilde{\theta} \right).
\]

\[\Box\]

**Proof of Theorem 2.** From Lemma 1, we know that when \(\| \Delta_j \| = o(1), \alpha^{-j} - \alpha = o(1)\). In this proof, we check whether \(\| \Delta_j \| \rightarrow 0\) for all \(j = 1, \ldots, n_1\).

One can use the Kailath Variant formular (from 3.1.2 of (Petersen and Pedersen, 2008)) to obtain

\[
\begin{align*}
\hat{\theta}_0(\lambda) - \hat{\theta}_0^{-j}(\lambda) \\
& = (X^\top X + n \lambda I_d)^{-1} \hat{X}^\top y \\
& = \left( X^\top X + n \lambda I_d \right)^{-1} \left( \frac{X^\top X + n \lambda I_d}{1 - x_j^\top (X^\top X + n \lambda I_d)^{-1} x_j} - x_j^\top \right) y_j \\
& = y_j (X^\top X + n \lambda I_d)^{-1} x_j - \frac{\hat{y}_j (X^\top X + n \lambda I_d)^{-1} x_j + y_j S_j(\lambda)(X^\top X + n \lambda I_d)^{-1} x_j}{1 - S_j(\lambda)} \\
& = \frac{y_j - \hat{y}_j(\lambda)}{1 - S_j(\lambda)} (X^\top X + n \lambda I_d)^{-1} x_j,
\end{align*}
\]

where \(\hat{y}_j(\lambda) = \hat{\theta}_0(\lambda)^\top x_j\).
Based on (Hastie et al., 2019), almost surely, denote \( A_i = n_1 (X_{-i} X_{-i} + \lambda n_1 I_d)^{-1} \), and \( \delta_i = \frac{x_i}{\sqrt{n_1}} \), then

\[
x_i^\top (X^\top X + \lambda n_1 I_d)^{-2} x_i = \frac{1}{n_1} \delta_i \left( A_i - \frac{A_i \delta_i \delta_i^\top A_i}{1 + \delta_i^\top A_i \delta_i} \right)^2 \delta_i
\]

\[
= \frac{1}{n_1} \delta_i \left( A_i^2 - 2 \frac{A_i^2 \delta_i \delta_i^\top A_i}{1 + \delta_i^\top A_i \delta_i} + \frac{A_i \delta_i \delta_i^\top A_i^2 \delta_i \delta_i^\top A_i}{(1 + \delta_i^\top A_i \delta_i)^2} \right) \delta_i
\]

\[
= \frac{1}{n_1} \left( \delta_i A_i^2 \delta_i - 2 \frac{\delta_i^\top A_i^2 \delta_i \delta_i^\top A_i^2 \delta_i}{1 + \delta_i^\top A_i \delta_i} + \frac{(\delta_i^\top A_i \delta_i)^2 \delta_i^\top A_i^2 \delta_i}{(1 + \delta_i^\top A_i \delta_i)^2} \right)
\]

\[
= \frac{1}{n_1} \frac{\delta_i^2 A_i^2 \delta_i}{(1 + \delta_i^\top A_i \delta_i)^2} \quad \text{a.s.}
\]

Finally, for \( y_i - \hat{y}_i \), we have

\[
y_i - \hat{y}_i = y_i - x_i^\top (X^\top X + \lambda n_1 I_d)^{-1} X^\top X \theta_0 + \epsilon
\]

\[
= \epsilon_i - x_i^\top (X^\top X + \lambda n_1 I_d)^{-1} X^\top \epsilon + \lambda n_1 x_i^\top (X^\top X + \lambda n_1 I_d)^{-1} \theta_0.
\]

Using Sherman–Morrison formula, we have

\[
x_i^\top (X^\top X + \lambda n_1 I_d)^{-1} \theta_0 = x_i^\top \left[ \frac{1}{n_1} A_i - A_i x_i^\top A_i / n_1^2 \right] \theta_0,
\]

thus

\[
(x_i^\top (X^\top X + \lambda n_1 I_d)^{-1} \theta_0)^2
\]

\[
= x_i^\top \frac{1}{n_1^2} A_i \theta_0 \theta_0^\top A_i x_i - \frac{2}{n_1^2} x_i^\top A_i \theta_0 \theta_0^\top A_i x_i^\top A_i / n_1^2 x_i + x_i^\top A_i x_i^\top A_i / n_1^2 \theta_0 \theta_0^\top A_i x_i^\top A_i / n_1^2 x_i
\]

\[
= O_p (\text{tr} (A_i^2 \theta_0 \theta_0^\top) / n_1^2)
\]

\[
= O_p (m_{\gamma}(-\lambda) / n_1^2).
\]

In addition,

\[
(x_i^\top (X^\top X + \lambda n_1 I_d)^{-1} X^\top \epsilon)^2 \rightarrow \sigma^2 x_i^\top (X^\top X + \lambda n_1 I_d)^{-1} X^\top X (X^\top X + \lambda n_1 I_d)^{-1} x_i,
\]

which converges to a constant.

Finally, given the distribution of \( \epsilon \), we have with probability tending to 1,

\[
\sup_j \| \theta_0(\lambda) - \hat{\theta}_0^{-1}(\lambda) \|^2 = o((\log n_1) / n_1).
\]