On the Saturation Effects of Spectral Algorithms in Large Dimensions

Weihao Lu, Haobo Zhang, Yicheng Li, Qian Lin

Tsinghua University

November 11, 2024

Settings

$(x_i, y_i) \in \mathbb{S}^d \times \mathbb{R}, i \in [n]$ are i.i.d. samples $\quad y_i = f_\star(x_i) + \epsilon_i \quad \mathbb{E}[\epsilon_i \mid x_i] \le \sigma^2$
Large dimensional framework: $n \asymp d^\gamma$

The goal is to find an estimator $\widehat{f}$ with small loss:

$$\mathcal{E} := \left\| \widehat{f} - f_\star \right\|_{L^2}^2 .$$

Inner product kernel function: $K : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}$, $K(x, x') = \Phi(\langle x, x' \rangle)$

Assume $f_\star \in [\mathcal{H}]^s$, $s > 0$, where $\mathcal{H}$ is the Reproducing Kernel Hilbert Space (RKHS) induced by K.

Settings

$(x_i, y_i) \in \mathbb{S}^d \times \mathbb{R}, i \in [n]$ are i.i.d. samples $\quad y_i = f_\star(x_i) + \epsilon_i \quad \mathbb{E}[\epsilon_i \mid x_i] \leq \sigma^2$
Large dimensional framework: $n \asymp d^\gamma$

The goal is to find an estimator $\widehat{f}$ with small loss:

$$\mathcal{E} := \left\| \widehat{f} - f_\star \right\|_{L^2}^2 .$$

Inner product kernel function: $K : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}$, $K(x, x') = \Phi(\langle x, x' \rangle)$

Assume $f_\star \in [\mathcal{H}]^s$, $s > 0$, where $\mathcal{H}$ is the Reproducing Kernel Hilbert Space (RKHS) induced by K.

Note.

- Mercer Decomposition: $K(x, x') = \sum_i \lambda_i e_i(x) e_i(x')$
- $\{\lambda_i\}$ are the eigenvalues in descending order, and $\{e_i(\cdot)\}$ are the eigenfunctions
- $\mathcal{H} = \left\{ \sum_i a_i \lambda_i^{1/2} e_i : (a_i)_i \in \ell_2 \right\}$, norm $\| \sum_i a_i \lambda_i^{1/2} e_i \|_{\mathcal{H}}^2 := \sum_i a_i^2$
- $[\mathcal{H}]^s = \left\{ \sum_i a_i \lambda_i^{s/2} e_i : (a_i)_i \in \ell_2 \right\}$, norm $\| \sum_i a_i \lambda_i^{s/2} e_i \|_{[\mathcal{H}]^s}^2 := \sum_i a_i^2$

Analytic Filtering Functions and Spectral Algorithms

Definitions Review

- $\varphi_\lambda(\cdot)$: an analytic filter function of order $\tau \geq 1$
- Let $K_x : \mathbb{R} \to \mathcal{H}$ be defined by $K_x(y) = y \cdot K(x, \cdot)$
- Define $T_x = K_x K_x^*$ and $T_X = \frac{1}{n} \sum_{i=1}^{n} T_{x_i}$
- Define $\hat{g}_Z = \frac{1}{n} \sum_{i=1}^{n} y_i \cdot K(x_i, \cdot)$
- The estimator for the analytic spectral algorithm is defined as

$$\hat{f}_\lambda = \varphi_\lambda(T_X)\hat{g}_Z. \tag{1}$$

Summary:

Analytic filtering function of order τ $\quad\rightarrow\quad$ Analytic spectral algorithm of order τ

Spectral Algorithms with Different τ

Example 1 (Kernel Gradient Flow)

$$\varphi_{\lambda}^{\mathrm{GF}}(z) = \frac{1 - e^{-\lambda^{-1} z}}{z}, \quad \tau = \infty.$$

Example 2 (Kernel Ridge Regression)

$$\varphi_{\lambda}^{\mathrm{KRR}}(z) = \frac{1}{z + \lambda}, \quad \tau = 1.$$

Example 3 (Iterated Ridge Regression, $q = 1, 2, \cdots$)

$$\varphi_{\lambda}^{\mathrm{IT},q}(z) = \frac{1}{z}\left[1 - \frac{\lambda^{q}}{(z + \lambda)^{q}}\right], \quad \tau = q.$$

Assume there exists a constant $\beta > 1$ such that the eigenvalues satisfy $\lambda_j \asymp j^{-\beta}$.

Review of Saturation Effects in Fixed Dimensions

Minimax rate: $n^{-s\beta/(s\beta+1)}$

Optimal convergence rate of algorithms:

- Kernel Gradient Flow ($\tau = \infty$): $n^{-s\beta/(s\beta+1)}$
- Kernel Ridge Regression ($\tau = 1$): $s \leq 2$, $n^{-s\beta/(s\beta+1)}$; $s > 2$, $n^{-2\beta/(2\beta+1)}$
- Analytic spectral algorithm of order τ:
  $s \leq 2\tau$, $n^{-s\beta/(s\beta+1)}$; $s > 2\tau$, $n^{-2\tau\beta/(2\tau\beta+1)}$

When $s > 2$, Kernel Ridge Regression performs worse than certain spectral algorithms (e.g., Kernel Gradient Flow).

- Inspired by the uniform convergence concepts of neural networks and Kernel Gradient Flow, large-dimensional spectral algorithms have garnered renewed attention.
- Previous research mainly focuses on Kernel Ridge Regression (KRR).

- Inspired by the uniform convergence concepts of neural networks and Kernel Gradient Flow, large-dimensional spectral algorithms have garnered renewed attention.
- Previous research mainly focuses on Kernel Ridge Regression (KRR).

If large-dimensional KRR exhibits saturation effects, then KRR underperforms compared to Kernel Gradient Flow.

Hence, the above results cannot be directly extended to large-dimensional neural networks.

- Inspired by the uniform convergence concepts of neural networks and Kernel Gradient Flow, large-dimensional spectral algorithms have garnered renewed attention.
- Previous research mainly focuses on Kernel Ridge Regression (KRR).

If large-dimensional KRR exhibits saturation effects, then KRR underperforms compared to Kernel Gradient Flow.

Hence, the above results cannot be directly extended to large-dimensional neural networks.

Motivations: Does the saturation effect exist in large dimensions?

Main Result

When d is sufficiently large:

Analytic Spectral Algorithm with $\tau \geq s$

Suppose one of the following conditions holds:

$$\text{(i) } \tau = \infty, \quad \text{(ii) } s > 1/(2\tau), \quad \text{(iii) } \gamma > ((2\tau+1)s)/(2\tau(1+s));$$

Then there exists a penalty coefficient $\lambda^\star > 0$ such that

$$\mathbb{E}(\mathcal{E}(\hat{f}_{\lambda^\star}) \mid X) = \Theta_{d,\mathbb{P}}\left(d^{-\min\{\gamma-p, s(p+1)\}}\right) \cdot \operatorname{poly}(\ln(d)).$$

Analytic Spectral Algorithm with $\tau < s$

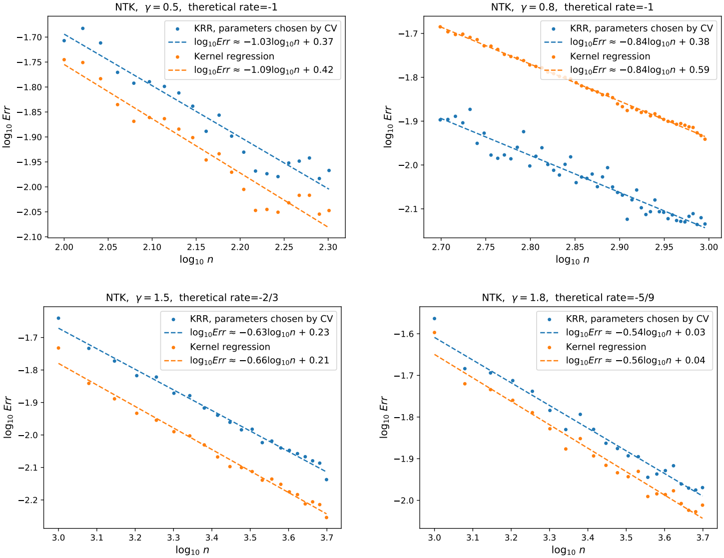
$$\mathbb{E}(\mathcal{E}(\hat{f}_{\lambda^\star}) \mid X) = \Theta_{d,\mathbb{P}}\left(d^{-\min\left\{\gamma-p, \frac{\tau(\gamma-p+1)+p\tilde{s}}{\tau+1}, \tilde{s}(p+1)\right\}}\right) \cdot \operatorname{poly}(\ln(d)),$$

where $\tilde{s} = \min\{s, 2\tau\}$.

Minimax Lower Bound

$$\inf_{\hat{f}} \sup_{f_\star \in R_\gamma[\mathcal{B}]^s} \mathbb{E}_{(X,Y)\sim\rho^{\otimes n}}[\mathcal{E}] = \Omega_d\left(d^{-\min\{\gamma-p, s(p+1)\}}\right) \Big/ \operatorname{poly}(\ln(d)).$$

Numerical Experiment I: Convergence Rate of Kernel Gradient Flow and KRR Loss when $s = 1$



Numerical Experiment II: Performance Comparison of KRR and Kernel Gradient Flow for $s = 1.9 > 1$

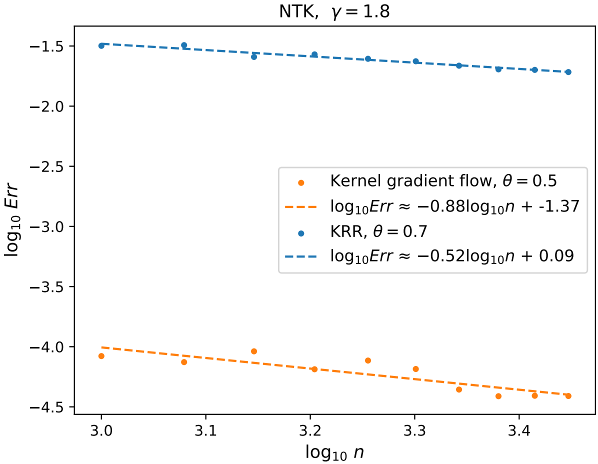


Figure: Comparison of KRR and Kernel Gradient Flow loss. The penalty coefficients for both algorithms are set as $\lambda = cd^{\theta}$, with θ chosen as the theoretically optimal value.