Lecture 18

Modern NN results

NN ~ GP

SVM → RFF, DD

AIC, BIC, CV, SRM

Structural Risk Minimization

RKHS Reproducing Kernel Hilbert Spaces

$$\mathcal{H}_1 = \{ f(x) = \sum_{i=1}^{\infty} \alpha_i k(x^i, x),\ \alpha_i \in \mathbb{R},\ x^i \in \mathcal{X} \text{ for all } i \}$$

$f : \mathcal{X} \to \mathbb{R}$

$$k(x, x') = e^{-\frac{\|x - x'\|^2}{2\sigma^2}}$$

A1 $\mathcal{H}_1$ can approximate other f's closely

$\mathcal{H}$ is inner product space — $\dim \mathcal{H} < \infty$ Euclidean $\cong \mathbb{R}^d$

$= \infty$ Hilbert space (if complete) "spaces of functions"

$\langle , \rangle$ scalar prod : $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$

- linear in each argument
- $\langle x, x \rangle \geq 0$ for $x \neq 0$

$\|x\|^2 = \langle x, x \rangle$

SVM $\longrightarrow$ estimates an $\hat{f} \in \mathcal{H}_1$
fits $\hat{f}$ to $(x^i, y^i)_{i=1:}$

Thm (Mercer) iff $\iint_{\mathcal{X}\mathcal{X}} k(x, x') g(x) g(x') \, dx \, dx' \geq 0$ for any $g \in L^2(\mathcal{X})$

then $k(x, x') = \sum_{\ell=1}^{\infty} \varphi_\ell(x) \varphi_\ell(x') = \varphi(x)^T \varphi(x')$

linearly independent

$\exists$ feature space $x \longmapsto \varphi(x) \subset \mathcal{H}$

$$\varphi(x) = \begin{bmatrix} \varphi_1(x) \\ \vdots \\ \varphi_\ell(x) \\ \vdots \end{bmatrix}$$

$$\|x\|^2 = \varphi(x)^T\varphi(x) = k(x,x)$$

$$\|x-x'\|^2 = k(x,x) + k(x',x') - 2k(x,x')$$

linear functional on $\mathcal{H}$

$$\beta^T\varphi = \sum_{\ell=1}^{\infty} \beta_\ell \varphi_\ell$$

$\uparrow$ $\varphi_\ell(x)$

$$f(x) = \sum \alpha_i k(x^i, x)$$

$$= \sum_i \alpha_i \sum_\ell \varphi_\ell(x^i)\varphi_\ell(x) =$$

$$= \sum_\ell \varphi_\ell(x) \cdot \underbrace{\sum_i \alpha_i \varphi_\ell(x^i)}_{\beta_\ell \in \mathbb{R}}$$

Riesz: $\Rightarrow \exists f \in \mathcal{H}$,

$$f = \sum_{i=1}^{n} \alpha_i k(x^i, \cdot)$$

so that $\beta^T\varphi(x) = f(x)$ for all x

linear functional in $\mathcal{H}$

$x \in \mathcal{H}$

$$x \longmapsto k(x, \cdot)$$

SVM original name typically classifier $\subset$ kernel machines

non-linear functions (predictors) in RKHS over X

SV Classification + kernel trick

SV Multiclass

SV Regression ≠ NW kernel regression

SV 1-class classification = Estimating supp of distribution

Kernel PCA

data matrix $X = \begin{bmatrix} - x^1 - \\ \cdots \\ - x^n - \end{bmatrix} \in \mathbb{R}^{n \times d}$ assume $1^T X = 0$, $\sum x^i = 0$

$G = XX^T = U \Lambda^2 U^T \Rightarrow Y = U\Lambda$ can be obtained from G !! ∴ PCA

Gram matrix

$\Sigma = \frac{1}{n} X^T X$ → PCA → $\Sigma = \frac{1}{n} V \Lambda^2 V^T$ → $Y \leftarrow XV_{1:s}$ project X on principal space

$V_{1:s}$ s principal directions

SVD $X = U \Lambda V^T$ (ortho, diag, orthog)

$XV = U\Lambda V^T V = U\Lambda$

$XV_{1:s}$ → first s colums of $U\Lambda$

Kernel PCA algorithm

1. Compute $G = [k(x^i, x^j)]_{i,j=1:n}$ gram Matrix $n \times n$
2. Eigendecomposition $\Rightarrow U, \Lambda$
3. Select top d e-values
4. $x^i \longrightarrow$ row i of $U_{1:d} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix}$

Kernel machines
for Big n
(and φ(x) high dim)

Lecture VI-2: SVM with Random Fourier Features

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

November, 2021

Reading: Ali Rahimi and Ben Recht "Random features for large-scale Kernel Machine", NIPS 2007. Test of Time Award, NIPS 2017.

Problem: Kernel machines scale with sample size n

- Gram matrix $G = [k(x^i, x^j)]_{i,j=1}^n \in \mathbb{R}^{n\times n}$. Expensive/intractable for n large!
- Want to: benefit from infinite dimensional feature spaces, e.g. Gaussian kernel, AND have constant dimension D for any n
- **Idea** approximate $k(x, x')$ with finite sum.
- Equivalently, approximate feature space $\mathcal{H}$ with D-dimensional feature space. How? Pick D features at random!

Why is this possible? Bochner's Theorem

$\Delta = x - x'$

Let $K(x, x') = K(x - x')$ be a continuous shift invariant kernel.

Theorem [Bochner]

$K(x, x')$ is a positive definite kernel iff $K(\Delta)$ is the Fourier transform of some non-negative measure $p(\omega)$.

$$K(\Delta) = \int_{\mathbb{R}^d} p(\omega) e^{-i\omega^T \Delta} d\omega \approx \sum_{j=1}^{D} e^{-i w_j^T \Delta} \quad (1)$$

$w_j \sim p(w)$ iid

$K(\Delta)$	$p(\omega)$	
$e^{-\|\|\Delta\|\|^2/2}$	$(2\pi)^{-d/2} e^{-\|\|\omega\|\|^2/2}$	Gaussian (RBF) kernel
$e^{-\|\|\Delta\|\|_1}$	$(2\pi)^{-d} \prod_{j=1}^d \frac{1}{1+\omega_j^2}$	Laplace kernel
$\prod_{j=1}^d \frac{2\pi}{1+\omega_j^2}$	$e^{-\|\|\Delta\|\|_1}$	product kernel

From Bochner to RFF

- Note that $e^{-i\omega\Delta} = e^{-i\omega^T x}(e^{-i\omega^T x'})^*$ and let $\zeta_\omega(x) = e^{-i\omega^T x}$.
- Then $K(\Delta) = E_{p(\omega)}[\zeta_\omega(x)\zeta_\omega^*(x')] \approx \frac{1}{D}\sum_{j=1}^{D} \zeta_{\omega_j}(x)\zeta_{\omega_j}^*(x')$ with $\omega_{1:D} \sim$ i.i.d. $p(\omega)$
- D is the sample size, must be large enough for good approximation
- $\zeta_{\omega_{1:D}}$ form a **random feature space** of dimension D
- Feature map is $x \rightarrow \tilde{\phi}(x) = \frac{1}{\sqrt{D}}[\zeta_{\omega_1} \ldots \zeta_{\omega_D}]$

Fact Because $K()$ is real, the random complex features $\zeta_\omega \leftarrow \sqrt{2}cos(\omega^T x + b)$ with $b \sim uniform[0, 2\pi]$

- **Significance** Infinite dimensional feature vector $\phi(x)$ approximated by D-dimensional feature vector $\tilde{\phi}(x)$. Hence, primal problem of dimension D can be solved instead of dual of dimension n.
- Opens up SVM/kernel machines for large data

Approximation

Theorem [Rahimi and Recht 07]

Assume space $\mathcal{X}$ is compact of diameter $d_{\mathcal{X}}$ and let $\sigma_p^2 = E_p[\omega^T \omega]$ be the standard deviation of $p(\omega)$. Then,

1.
$$Pr\left[\sup_{x,x' \in \mathcal{X}} |\tilde{\phi}(x)^T \tilde{\phi}(x') - K(x,x')| \geq \epsilon\right] \leq e^{-\frac{D\epsilon^2}{4(d+2)}} \left(\frac{2^4 \sigma_p d_{\mathcal{X}}}{\epsilon}\right)^2 \tag{2}$$

2. For δ confidence level,
$$D = \Omega\left(\frac{d}{\epsilon^2} \ln \frac{\sigma_p d_{\mathcal{X}}}{\epsilon}\right) \tag{3}$$

Kernel machine with RFF algorithm

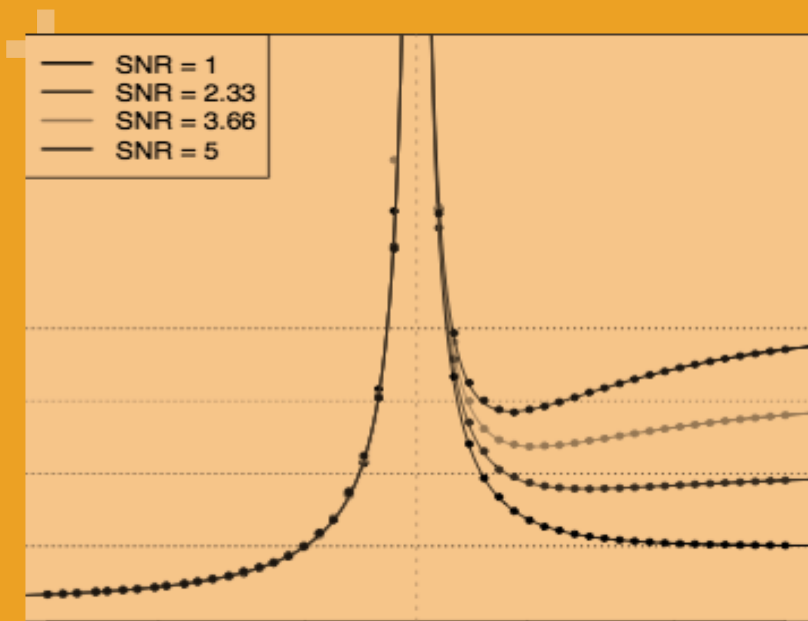
In Data $x^{1:n}, y^{1:n}$, kernel K

1. Fourier transform $p(\omega) = \frac{1}{2\pi} \int_{\mathbb{R}^d} e^{-i\omega^T \Delta} K(\Delta) d\Delta$.
2. Choose D.
3. Sample $\omega_{1:D}$ i.i.d. from p. Sample $b_{1:D}$ uniformly from $[0, 2\pi]$.
4. Map data to features $\tilde{\phi}(x^i) = \sqrt{\frac{2}{D}} [cos(\omega_j^T x^i + b_j)]_{j=1:D}$ for all $i = 1 : n$.
5. Solve SVM Primal problem; obtain $w \in \mathbb{R}^D$ and intercept $b \in \mathbb{R}$. (note that b is not one of $b_{1:D}$).

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$$

$$\|\varphi(x) - \varphi(x')\|^2 = \|\varphi(x)\|^2 + \|\varphi(x')\|^2 - 2\,\varphi(x)^T\varphi(x')$$

$$k(x,x) = e^0 = 1 \quad \Rightarrow \quad = 2 - 2\,k(x, x')$$



- What is it?
- Relation with RFF and GD
- appears in simple settings

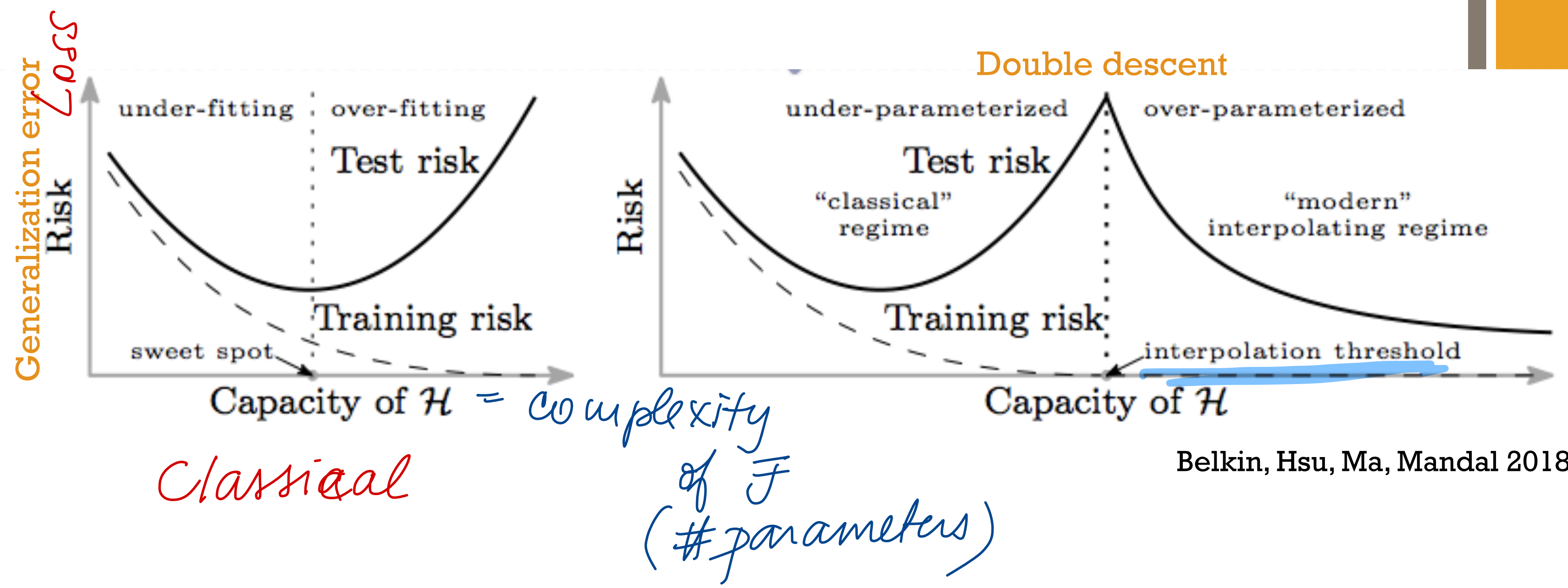
Double Descent

STAT 535+LPL2019

Beyond the Bias-Variance trade-off

Marina Meila
University of Washington

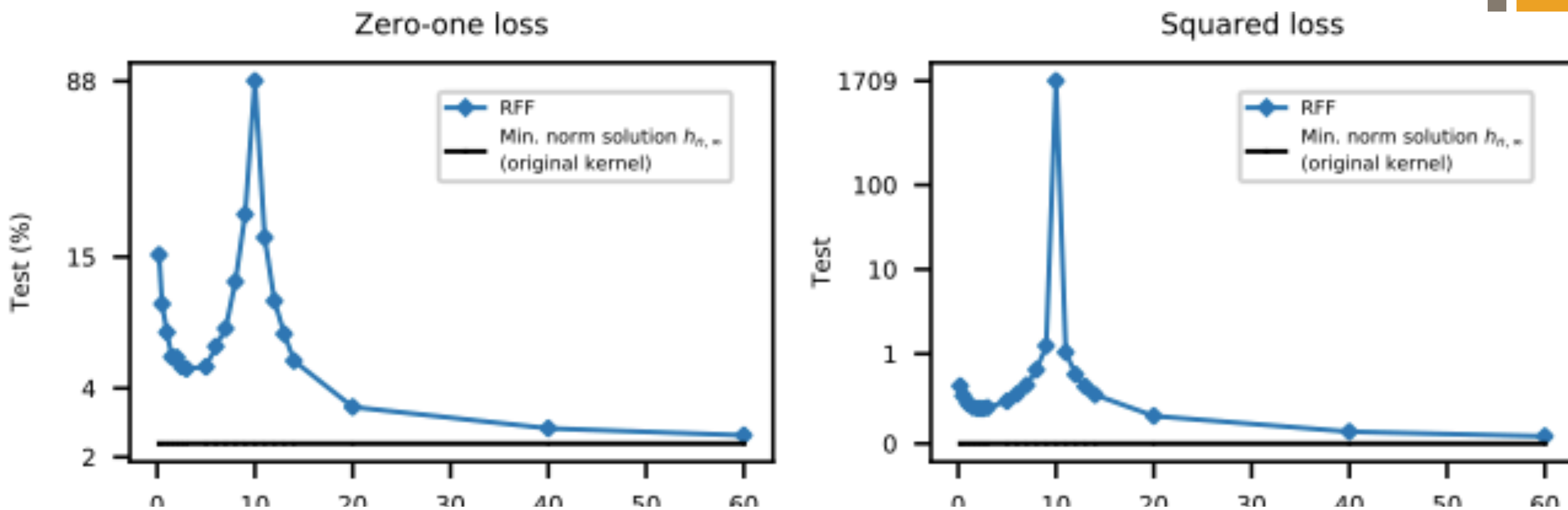
What is observed



Belkin, Hsu, Ma, Mandal 2018

- Classical regime p < N
- Modern/Deep Learning/High dimensional regime N > n
  - Think N fixed, p increases, gamma=p/N
  - Training error = 0 (interpolation)
  - Test error decreases with p (or gamma)

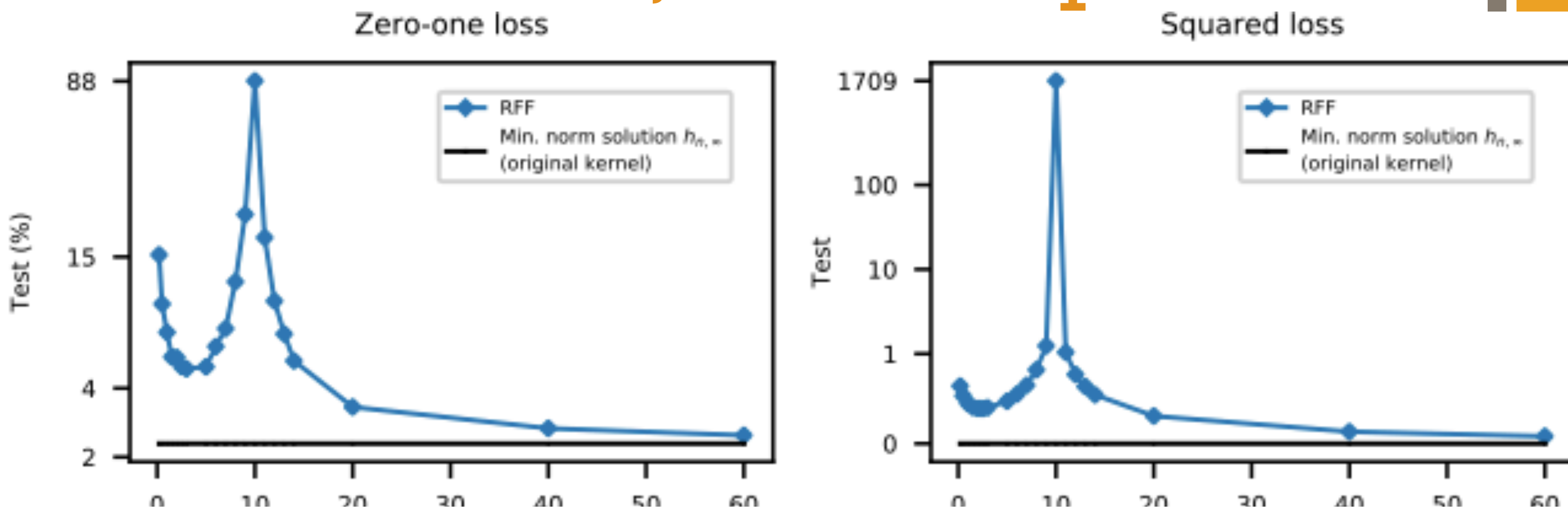
What is observed



Belkin, Hsu, Ma, Mandal 2018

- Double descent curves for the generalization error
  - Random Fourier Features (RFF)
  - ReLU 2 layer networks (with random first layer weights)
  - Random Forests, l2-Adaboost
  - Linear regression
- With and without noise

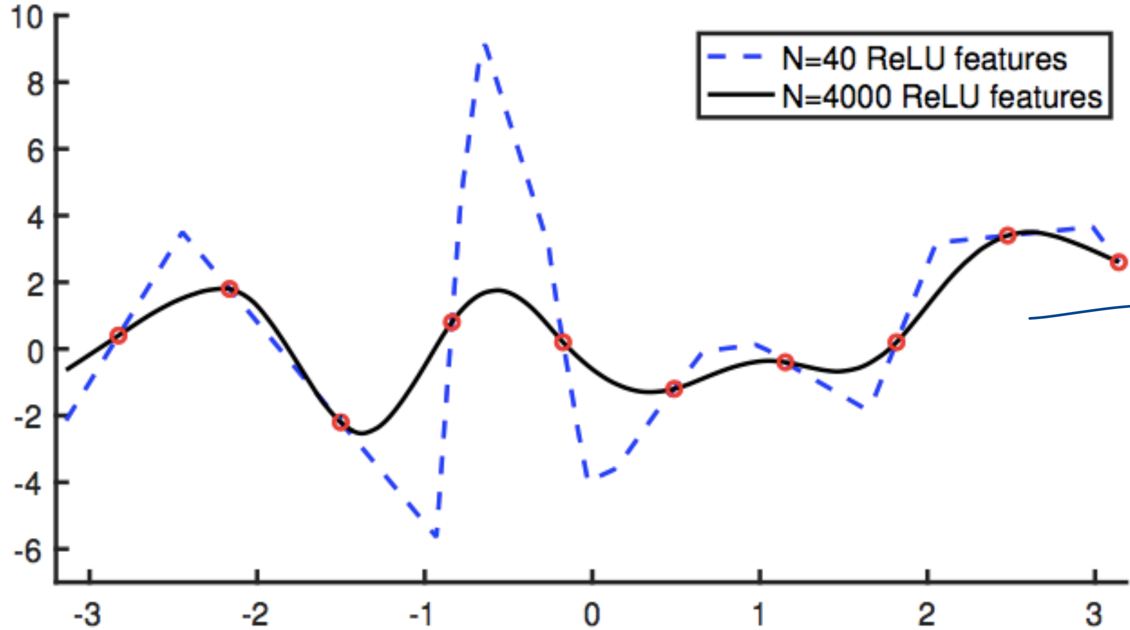
Double descent, the case p > N



Belkin, Hsu, Ma, Mandal 2018

- Model y = <phi(x), beta >
- Large N (cover a compact data domain)
- Features random
- Min-norm solution beta*

Main intuition [Belkin et al.]



Want $f(x^i) = y^i$ for $i = 1:n$

and f smoothest

$\mathcal{F} \supset \mathcal{F}_0 = \{ f : f(x^i) = y^i, \ i = 1:n \}$

$\hat{f}$ = smoothest in $\mathcal{F}_0$

- The target function h* is (mostly) smooth
  - i.e. $||h^*||_{RKHS}$ is small
- p > N, no noise, hence h_p interpolates data
- Train to minimize $||h_p||$ subject to 0 training error
- Then $||h_p||$ will decrease with p!

Random Fourier Features (RFF)

Random Fourier features. We first consider a popular class of non-linear parametric models called *Random Fourier Features* (*RFF*) [30], which can be viewed as a class of two-layer neural networks with fixed weights in the first layer. The RFF model family $\mathcal{H}_N$ with N (complex-valued) parameters consists of functions $h\colon \mathbb{R}^d \to \mathbb{C}$ of the form

$$h(x) = \sum_{k=1}^{N} a_k \phi(x; v_k) \quad \text{where} \quad \phi(x; v) := e^{\sqrt{-1}\langle v, x\rangle},$$

and the vectors $v_1, \ldots, v_N$ are sampled independently from the standard normal distribution in $\mathbb{R}^d$. (We consider $\mathcal{H}_N$ as a class of real-valued functions with $2N$ real-valued parameters by taking real and imaginary parts separately.) Note that $\mathcal{H}_N$ is a randomized function class, but as $N \to \infty$, the function class becomes a closer and closer approximation to the Reproducing Kernel Hilbert Space (RKHS) corresponding to the Gaussian kernel, denoted by $\mathcal{H}_\infty$.

- RFF → $\mathcal{H}_{\text{infinity}}$

+ Theorem

Theorem 1. *Fix any $h^* \in \mathcal{H}_\infty$. Let $(x_1, y_1), \ldots, (x_n, y_n)$ be independent and identically distributed random variables, where x_i is drawn uniformly at random from a compact cube*[2] *$\Omega \subset \mathbb{R}^d$, and $y_i = h^*(x_i)$ for all i. There exists absolute constants $A, B > 0$ such that, for any interpolating $h \in \mathcal{H}_\infty$ (i.e., $h(x_i) = y_i$ for all i), so that with high probability*

$$\sup_{x \in \Omega} |h(x) - h^*(x)| < A e^{-B(n/\log n)^{1/d}} \left(\|h^*\|_{\mathcal{H}_\infty} + \|h\|_{\mathcal{H}_\infty}\right).$$