



# A random matrix analysis of random fourier features

beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent

Kailong Wang<sup>1</sup>

<sup>1</sup>Ph.D. of ECE  
Rutgers University

ECE 539 HDP, May 8, 2023

# Table of Contents

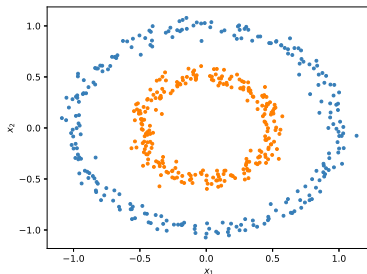
- 1 Motivation
- 2 Random Fourier Features
- 3 An analysis of RFF



# Linear Classification with Non-linear Input

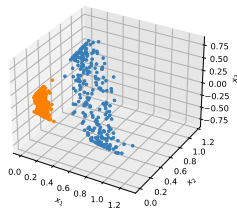
Consider a binary classification problem with non-linear (e.g. polynomial) samples. This is not separable with linear function.

$$\text{(e.g. } \mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ \vdots & \vdots \\ x_{N,1} & x_{N,2} \end{bmatrix} \in \mathbb{R}^{N \times 2}.)$$



# Lifting

One idea is to **LIFT** the samples into a higher dimensional space in which the samples are linearly separable.



The Lifting function in this case is  $\phi(\mathbf{X}) = \begin{bmatrix} x_{1,1}^2 & x_{1,2}^2 & \sqrt{2}x_{1,1}x_{1,2} \\ x_{2,1}^2 & x_{2,2}^2 & \sqrt{2}x_{2,1}x_{2,2} \\ \vdots & \vdots & \vdots \\ x_{N,1}^2 & x_{N,2}^2 & \sqrt{2}x_{N,1}x_{N,2} \end{bmatrix}$ .

# Curse of Dimensionality

Consider solving the above problem with *support vector machine* (SVM).

$$\mathcal{L}(\mathbf{w}, \alpha) = \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_n^N \sum_m^N \alpha_n \alpha_m y_n y_m (\mathbf{x}_n^T \mathbf{x}_m).$$

The  $\mathbf{w}$  is the linear decision boundary and  $\alpha$  is a vector of Lagrange multipliers.

# Curse of Dimensionality

Consider solving the above problem with *support vector machine* (SVM).

$$\mathcal{L}(\mathbf{w}, \alpha) = \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_n^N \sum_m^N \alpha_n \alpha_m y_n y_m (\mathbf{x}_n^\top \mathbf{x}_m).$$

The  $\mathbf{w}$  is the linear decision boundary and  $\alpha$  is a vector of Lagrange multipliers.

We need to use lifting function  $\phi(X)$  to make the samples linearly separable. Specifically, we replace  $(\mathbf{x}_n^\top \mathbf{x}_m)$  with  $(\phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m))$ .

$$\begin{aligned} \phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m) &= \begin{bmatrix} x_{n,1}^2 & x_{n,2}^2 & \sqrt{2}x_{n,1}x_{n,2} \end{bmatrix} \begin{bmatrix} x_{m,1}^2 & x_{m,2}^2 & \sqrt{2}x_{m,1}x_{m,2} \end{bmatrix}^\top \\ &= x_{n,1}^2 x_{m,1}^2 + x_{n,2}^2 x_{m,2}^2 + 2x_{n,1}x_{n,2}x_{m,1}x_{m,2} \end{aligned}$$



# Curse of Dimensionality

Consider solving the above problem with *support vector machine* (SVM).

$$\mathcal{L}(\mathbf{w}, \alpha) = \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_n^N \sum_m^N \alpha_n \alpha_m y_n y_m (\mathbf{x}_n^\top \mathbf{x}_m).$$

The  $\mathbf{w}$  is the linear decision boundary and  $\alpha$  is a vector of Lagrange multipliers.

We need to use lifting function  $\phi(X)$  to make the samples linearly separable. Specifically, we replace  $(\mathbf{x}_n^\top \mathbf{x}_m)$  with  $(\phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m))$ .

$$\begin{aligned} \phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m) &= \begin{bmatrix} x_{n,1}^2 & x_{n,2}^2 & \sqrt{2}x_{n,1}x_{n,2} \end{bmatrix} \begin{bmatrix} x_{m,1}^2 & x_{m,2}^2 & \sqrt{2}x_{m,1}x_{m,2} \end{bmatrix}^\top \\ &= x_{n,1}^2 x_{m,1}^2 + x_{n,2}^2 x_{m,2}^2 + 2x_{n,1}x_{n,2}x_{m,1}x_{m,2} \end{aligned}$$

Calculate the inner product in the  $\mathbb{R}^3$  across all  $N$  pairs of samples is acceptable. However, the lifting function  $\phi(X)$  is usually very high dimensional.

# Kernel Trick

Consider the following derivation,

$$\begin{aligned}(\mathbf{x}_n^\top \mathbf{x}_m)^2 &= ([x_{n,1} \ x_{n,2}][x_{m,1} \ x_{m,2}]^\top)^2 \\&= (x_{n,1}x_{m,1} + x_{n,2}x_{m,2})^2 \\&= x_{n,1}^2x_{m,1}^2 + x_{n,2}^2x_{m,2}^2 + 2x_{n,1}x_{n,2}x_{m,1}x_{m,2} \\&= \phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)\end{aligned}$$



# Kernel Trick

Consider the following derivation,

$$\begin{aligned}(\mathbf{x}_n^\top \mathbf{x}_m)^2 &= ([x_{n,1} \ x_{n,2}][x_{m,1} \ x_{m,2}]^\top)^2 \\&= (x_{n,1}x_{m,1} + x_{n,2}x_{m,2})^2 \\&= x_{n,1}^2x_{m,1}^2 + x_{n,2}^2x_{m,2}^2 + 2x_{n,1}x_{n,2}x_{m,1}x_{m,2} \\&= \phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)\end{aligned}$$

Instead of computing inner product in the high dimensional space, we compute the inner product in the original space.

# Kernel Trick

Consider the following derivation,

$$\begin{aligned}(\mathbf{x}_n^\top \mathbf{x}_m)^2 &= ([x_{n,1} \ x_{n,2}][x_{m,1} \ x_{m,2}]^\top)^2 \\&= (x_{n,1}x_{m,1} + x_{n,2}x_{m,2})^2 \\&= x_{n,1}^2x_{m,1}^2 + x_{n,2}^2x_{m,2}^2 + 2x_{n,1}x_{n,2}x_{m,1}x_{m,2} \\&= \phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)\end{aligned}$$

Instead of computing inner product in the high dimensional space, we compute the inner product in the original space.

The function

$$K(\mathbf{x}_n, \mathbf{x}_m) = (\mathbf{x}_n^\top \mathbf{x}_m)^2 = \phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)$$

is called a **kernel function**.

## There must be disadvantages...

Given training data  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \in \mathcal{X} \times \mathcal{Y}$ , where  $\mathcal{X} \subseteq \mathbb{R}^d$  and  $\mathcal{Y} \subseteq \mathbb{R}$ . Consider *Kernel Ridge Regression* (KRR), with  $\phi(\mathcal{X}) \subseteq \mathbb{R}^k$ , where  $k \rightarrow \infty$

$$\mathcal{L}(\mathbf{w}, \lambda) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_n^N (y_n - \mathbf{w}^\top \phi(\mathbf{x}_n))^2 + \lambda \mathbf{w}^\top \mathbf{w}.$$

Solving it with Lagrange multipliers  $\alpha$ , which is the solution of

$$(\mathbf{K} + \lambda \mathbf{I}_k) \alpha = \mathbf{y},$$

requires  $\Theta(k^3)$  time and  $\Theta(k^2)$  memory. Here  $\mathbf{K} \in \mathbb{R}^{k \times k}$  is the kernel matrix or Gram matrix defined by  $\mathbf{K}_{nm} \equiv K(\mathbf{x}_n, \mathbf{x}_m)$ .



## There must be disadvantages...

Given training data  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \in \mathcal{X} \times \mathcal{Y}$ , where  $\mathcal{X} \subseteq \mathbb{R}^d$  and  $\mathcal{Y} \subseteq \mathbb{R}$ . Consider *Kernel Ridge Regression* (KRR), with  $\phi(\mathcal{X}) \subseteq \mathbb{R}^k$ , where  $k \rightarrow \infty$

$$\mathcal{L}(\mathbf{w}, \lambda) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_n^N (y_n - \mathbf{w}^\top \phi(\mathbf{x}_n))^2 + \lambda \mathbf{w}^\top \mathbf{w}.$$

Solving it with Lagrange multipliers  $\alpha$ , which is the solution of

$$(\mathbf{K} + \lambda \mathbf{I}_k) \alpha = \mathbf{y},$$

requires  $\Theta(k^3)$  time and  $\Theta(k^2)$  memory. Here  $\mathbf{K} \in \mathbb{R}^{k \times k}$  is the kernel matrix or Gram matrix defined by  $\mathbf{K}_{nm} \equiv K(\mathbf{x}_n, \mathbf{x}_m)$ .

**Intuition:** Can we find a kernel function which lifts  $\mathcal{X}$  to  $\mathbb{R}^s$ , where  $d < s \ll k$ , while not sacrifices model performance?

## Some Prerequisites

### Shift Invariant Kernel (Radial Basis Function (RBF))

A kernel function  $K(\mathbf{x}_n, \mathbf{x}_m)$  is called **shift invariant** if it can be written as  $K(\mathbf{x}_n, \mathbf{x}_m) = g(\mathbf{x}_n - \mathbf{x}_m)$  for some function  $g(\cdot)$  (e.g.  $K_{Gaussian}(\mathbf{x}_n, \mathbf{x}_m) = \exp(-\gamma \|\mathbf{x}_n - \mathbf{x}_m\|_2^2)$ ).

### Mercer's Theorem

A continuous function  $K(\mathbf{x}_n, \mathbf{x}_m)$  is a valid kernel function if and only if the kernel matrix  $\mathbf{K}$  is **positive semi-definite**.

### Bochner's Theorem

A continuous function  $g(\cdot)$  is **positive semi-definite** if and only if it is the Fourier transform of a non-negative measure.



# Random Fourier Features

## Conclusion

A continuous **shift invariant** kernel  $K(\mathbf{x}_n, \mathbf{x}_m)$ , which is **positive semi-definite** (Mercer's Theorem), is the Fourier transform of a non-negative measure  $p(\cdot)$ .

$$\phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m) = K(\mathbf{x}_n, \mathbf{x}_m) = K(\mathbf{x}_n - \mathbf{x}_m) \quad (1)$$

$$= \int_{\mathbb{R}^d} p(\omega) \exp(i\omega^\top (\mathbf{x}_n - \mathbf{x}_m)) d\omega \quad (2)$$

$$= \mathbb{E}_\omega [\xi_\omega(\mathbf{x}_n)^* \xi_\omega(\mathbf{x}_m)] \quad (3)$$

Here  $\xi_\omega(\mathbf{x}) = \exp(i\omega^\top \mathbf{x}) = \begin{bmatrix} \cos(\omega^\top \mathbf{x}) \\ \sin(\omega^\top \mathbf{x}) \end{bmatrix}$  and hence  $\xi_\omega(\mathbf{x}_n)^* \xi_\omega(\mathbf{x}_m)$  is an unbiased estimator of  $K(\mathbf{x}_n, \mathbf{x}_m)$  when  $\omega$  is drawn from  $p(\cdot)$ .



# Random Fourier Features

Since both the  $p(\cdot)$  and  $K(\Delta)$  are real-valued, we can replace  $\xi_\omega(\mathbf{x})$  with  $z_\omega(\mathbf{x}) = [\sqrt{2} \cos(\omega^\top \mathbf{x} + b)]$  where  $\omega$  is drawn from  $p(\omega)$  and  $b$  is uniformly drawn from  $[0, 2\pi]$ . Then eq. (3) becomes  $\mathbb{E}_\omega [z_\omega(\mathbf{x}_n)^\top z_\omega(\mathbf{x}_m)]$



# Random Fourier Features

Since both the  $p(\cdot)$  and  $K(\Delta)$  are real-valued, we can replace  $\xi_\omega(\mathbf{x})$  with  $z_\omega(\mathbf{x}) = [\sqrt{2} \cos(\omega^\top \mathbf{x} + b)]$  where  $\omega$  is drawn from  $p(\omega)$  and  $b$  is uniformly drawn from  $[0, 2\pi]$ . Then eq. (3) becomes  $\mathbb{E}_\omega[z_\omega(\mathbf{x}_n)^\top z_\omega(\mathbf{x}_m)]$

**Note:**  $z_\omega(\mathbf{x}_n)^\top z_\omega(\mathbf{x}_m)$  is an unbiased estimator of  $\phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)$ .  
The  $z_\omega(\mathbf{x})$  is not a lifting function.





# Random Fourier Features

Since both the  $p(\cdot)$  and  $K(\Delta)$  are real-valued, we can replace  $\xi_\omega(\mathbf{x})$  with  $z_\omega(\mathbf{x}) = [\sqrt{2} \cos(\omega^\top \mathbf{x} + b)]$  where  $\omega$  is drawn from  $p(\omega)$  and  $b$  is uniformly drawn from  $[0, 2\pi]$ . Then eq. (3) becomes  $\mathbb{E}_\omega[z_\omega(\mathbf{x}_n)^\top z_\omega(\mathbf{x}_m)]$

**Note:**  $z_\omega(\mathbf{x}_n)^\top z_\omega(\mathbf{x}_m)$  is an unbiased estimator of  $\phi(\mathbf{x}_n)^\top \phi(\mathbf{x}_m)$ .  
The  $z_\omega(\mathbf{x})$  is not a lifting function.

**Note:** To further reduce the variance of the estimator, we can randomly draw  $s$  samples of  $\omega$  and normalize each corresponding  $z_\omega(\mathbf{x})$  by  $\sqrt{s}$ . Then the inner product  $z(\mathbf{x}_n)^\top z(\mathbf{x}_m) = \frac{1}{s} \sum_{j=1}^s z_{\omega_j}(\mathbf{x}_n)^\top z_{\omega_j}(\mathbf{x}_m)$

# Algorithm

---

## Algorithm 1 Random Fourier Features

---

**Require:** A shift invariant kernel  $K(\mathbf{x}_n, \mathbf{x}_m) = K(\mathbf{x}_n - \mathbf{x}_m)$ .

**Ensure:** A randomized feature map  $z(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^s$  so that

$$z(\mathbf{x}_n)^\top z(\mathbf{x}_m) \approx K(\mathbf{x}_n, \mathbf{x}_m).$$

Compute the Fourier transform  $p(\cdot)$  of the kernel  $K : p(\omega) = \frac{1}{2\pi} \int \exp(-i\omega^\top \Delta) K(\Delta) d\Delta$

Draw  $s$  i.i.d. samples  $\omega_1, \omega_2, \dots, \omega_s \in \mathbb{R}^d$  from  $p(\cdot)$  and  $s$  i.i.d. samples  $b_1, b_2, \dots, b_s \in [0, 2\pi]$ .

Let  $z(\mathbf{x}) \equiv \sqrt{\frac{2}{s}} [\cos(\omega_1^\top \mathbf{x} + b_1) \ \cos(\omega_2^\top \mathbf{x} + b_2) \ \dots \ \cos(\omega_s^\top \mathbf{x} + b_s)]^\top$

---

# Convergence

Bound for a *fixed* pair of samples  $\mathbf{x}_n$  and  $\mathbf{x}_m$

Given  $z_\omega$  is bounded random variable between  $[-\sqrt{2}, \sqrt{2}]$ , with Hoeffding's Inequality, we have

$$\mathbb{P}(|z(\mathbf{x}_n)^\top z(\mathbf{x}_m) - K(\mathbf{x}_n, \mathbf{x}_m)| \geq \epsilon) \leq 2 \exp\left(-\frac{s\epsilon^2}{4}\right).$$

# Convergence



Bound for *all* pair of samples  $\mathbf{x}_n$  and  $\mathbf{x}_m$

Let  $\mathcal{M}$  be a compact subset of  $\mathbb{R}^d$  with diameter  $\text{diam}(\mathcal{M})$ . Then, for the mapping  $z$  defined in Algorithm 1, we have

$$\begin{aligned} & \mathbb{P}\left(\sup_{x,y \in \mathcal{M}} |z(\mathbf{x}_n)^\top z(\mathbf{x}_m) - K(\mathbf{x}_n, \mathbf{x}_m)| \geq \epsilon\right) \\ & \leq 2^8 \left(\frac{\sigma_{p(\cdot)} \text{diam}(\mathcal{M})}{\epsilon}\right)^2 \exp\left(-\frac{s\epsilon^2}{4(d+2)}\right). \end{aligned}$$



# Common RFF

Kernel	$K(\Delta)$	$p(\omega)$
Gaussian	$\exp(-\gamma \ \Delta\ _2^2)$	$(2\pi)^{-\frac{s}{2}} \exp(-\gamma \ \omega\ _2^2)$
Laplacian	$\exp(-\ \Delta\ _1)$	$\prod_d (\pi(1 + \omega_d^2))^{-1}$
Cauchy	$\prod_d 2(1 + \Delta_d^2)^{-1}$	$\exp(-\ \Delta\ _1)(?)$



# The challenge that RFF faces in the learning regime

Consider a machine learning system with  $d$  parameters, trained on a dataset of size  $N$ , asymptotic analysis has

**Classical regime:** either focuses on the (statistical) population  $N \rightarrow \infty$  limit, for  $d$  fixed, or the over-parameterized  $d \rightarrow \infty$  limit, for a given  $N$ .

**Modern regime:** modern learning system (e.g. Neural Network) usually has model complexity and data size increase together. A double asymptotic regime where  $N, d \rightarrow \infty, d/N \rightarrow c$  is established.

RFF has been shown that entry-wise the Gram matrix  $\xi(\mathbf{x})$  converges to the Gaussian kernel matrix as  $s \rightarrow \infty$  and this property remains in modern regime.

However, the convergence  $\|\Xi^T \Xi / s - \mathbf{K}\| \rightarrow 0$  no longer holds in spectral norm (blow-up). Here  $\Xi$  is the matrix formed by stacking  $\xi(\mathbf{x})$  for all samples.

# Setup

$$0 < \liminf_N \min\left\{\frac{s}{N}, \frac{d}{N}\right\} \leq \limsup_N \max\left\{\frac{s}{N}, \frac{d}{N}\right\} < \infty.$$

$$\limsup_N \|\mathbf{X}\|_2 < \infty \quad \limsup_N \|\mathbf{y}\|_\infty < \infty$$

In classical regime  $\|\mathbf{\Xi}^\top \mathbf{\Xi} / s\| \equiv \mathbf{K} \equiv \mathbf{K}_{\cos} + \mathbf{K}_{\sin}$

Training MSE:  $\mathcal{L}_{train} = \frac{1}{N} \|\mathbf{y} - \mathbf{\Xi}^\top \mathbf{w}\|_2^2 = \frac{\lambda^2}{N} \|\mathbf{Q}(\lambda) \mathbf{y}\|_2^2$  where

$$\mathbf{Q}(\lambda) \equiv \left( \frac{1}{N} \mathbf{\Xi}^\top \mathbf{\Xi} + \lambda \mathbf{I}_N \right)^{-1}$$

We want to assess the asymptotic  $\mathcal{L}_{train}$  by expectation which is equivalent to assess the asymptotic  $\mathbb{E}_\Omega \{\mathbf{Q}(\lambda)\}$  where  $\Omega$  is the matrix form of  $\omega$ , which is numerically hard.

**Object:** Find an asymptotic “alternative” for  $\mathbb{E}_\Omega \{\mathbf{Q}(\lambda)\}$  when  $d, s, N \rightarrow \infty$ .



## Some Vague Idea from me...

We want to show that with consideration of  $d, s, N$

$$\begin{aligned}\|\mathbb{E}_{\Omega}\{\mathbf{Q}(\lambda)\} - \hat{\mathbf{Q}}(\lambda)\|_2 &\rightarrow 0 \\ \hat{\mathbf{Q}}(\lambda) &\equiv \left( \frac{s}{N} \left( \frac{\mathbf{K}_{\cos}}{1 + \delta_{\cos}} + \frac{\mathbf{K}_{\sin}}{1 + \delta_{\sin}} \right) + \lambda \mathbf{I}_N \right)^{-1} \\ \delta_{\cos} &= \frac{1}{N} \text{tr}(\mathbf{K}_{\cos} \hat{\mathbf{Q}}) \quad \delta_{\sin} = \frac{1}{N} \text{tr}(\mathbf{K}_{\sin} \hat{\mathbf{Q}})\end{aligned}$$

When  $\frac{s}{N} \rightarrow \infty$ ,  $\delta_{\cos}, \delta_{\sin} \rightarrow 0$  and thus  $\hat{\mathbf{Q}} \simeq (\frac{s}{N} \mathbf{K})^{-1}$