Random Features and Quadratures for Kernel Approximation and Double Descent

Fanghui Liu

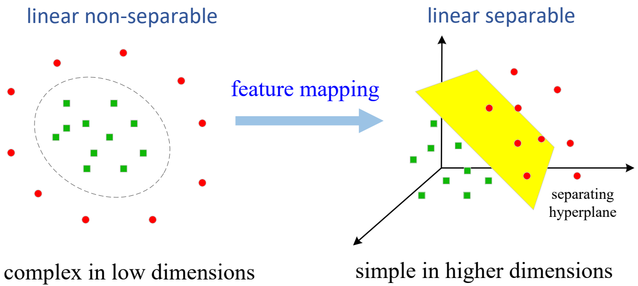
fanghui.liu@epfl.ch

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)
Switzerland

10th Nov. 2021

Outline

Research Overview: Kernel approximation



Scalability of kernel methods: n-by-n kernel matrix.
Solution: approximate the kernel by a low-rank representation

- Nyström approximation: approximate the kernel matrix
- Random Fourier features[1]: approximate the kernel function

[1] Rahimi A, Recht B. Random features for large-scale kernel machines, NeurIPS2007. (the test-of-time award in NeurIPS2017)

Research Overview: Random Fourier features

$$k(\mathbf{x},\mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}} \approx \varphi^\top(\mathbf{x})\varphi(\mathbf{x}')\,,$$

where $\varphi(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^s$ is an **explicit** feature mapping

Bochner's theorem [1]

For a shift-invariant $k(\mathbf{x},\mathbf{x}') = k(\mathbf{x}-\mathbf{x}')$ and positive definite kernel,

$$\begin{aligned} k(\mathbf{x},\mathbf{x}') &= \int_{\mathbb{R}^d} p(\omega) \exp\left(\mathrm{i}\omega^\top(\mathbf{x}-\mathbf{x}')\right) \mathrm{d}\omega \\ &\approx \frac{1}{s}\sum_{j=1}^{s} \exp(\mathrm{i}\omega_j^\top \mathbf{x}) \exp(\mathrm{i}\omega_j^\top \mathbf{x}')^* = \varphi(\mathbf{x})^\top \varphi(\mathbf{x}') \end{aligned}$$

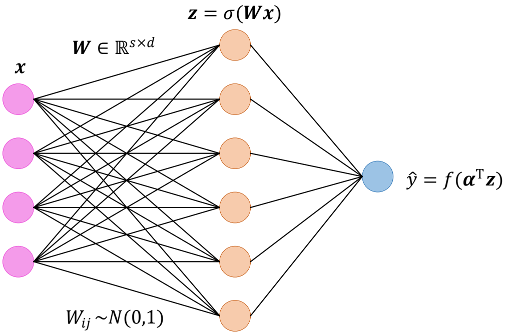
the explicit feature mapping:

$$\varphi(\mathbf{x}) := \frac{1}{\sqrt{s}}\left[\exp(-\mathrm{i}\omega_1^\top \mathbf{x}), \cdots, \exp(-\mathrm{i}\omega_s^\top \mathbf{x})\right]^\top.$$

Research Overview: Neural network view

RF model: a two-layer, (infinite)-width, fully-connected neural network

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \mathbb{E}_{\omega \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)}[\sigma(\omega^\top \mathbf{x})\sigma(\omega^\top \mathbf{x}')]$$



- Gaussian kernel: $\sigma(x) = [\cos(x), \sin(x)]^\top$
- the 1st-order arc-cosine kernel: $\sigma(x) = \max\{0, x\}$
- soft-max in attention: $\sigma(x) = \exp(x)$

Research Overview: Applied to Linearized Attention in Transformers

self attention

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \underbrace{\mathrm{softmax}(\mathbf{Q}\mathbf{K}^\top)}_{:=\mathbf{A}}\mathbf{V} \approx \mathbf{Q}'\mathbf{K}'^\top\mathbf{V}\,,$$

where $\mathbf{A}_{ij} = k(\mathbf{q}_i,\mathbf{k}_j) = \mathbb{E}[\sigma(\mathbf{q}_i)^\top\sigma(\mathbf{k}_j)]$

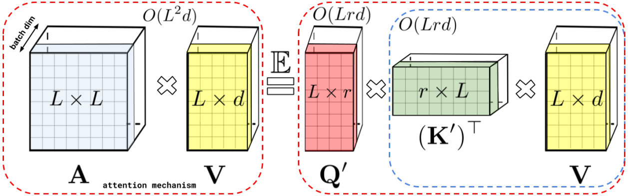


Figure: Approximation of self-attention. source: [2].

- soft-max in attention: $\exp(\mathbf{x}^\top\mathbf{x}') = \mathbb{E}_{\omega\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\exp\left(\omega^\top\mathbf{x} - \frac{\|\mathbf{x}\|_2^2}{2}\right)\exp\left(\omega^\top\mathbf{x}' - \frac{\|\mathbf{x}'\|_2^2}{2}\right)\right]$

Research Overview: Taxonomy

i) Monte Carlo sampling
- variance reduction
- acceleration

ii) Quasi-Monte Carlo sampling
- QMC
- SSF
- MM

$$
\begin{aligned}
k(\boldsymbol{x}-\boldsymbol{x}') &= \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) \exp\left(\mathrm{i}\boldsymbol{\omega}^\top(\boldsymbol{x}-\boldsymbol{x}')\right)\mathrm{d}\boldsymbol{\omega} \\
&= \int_{[0,1]^d} \exp\left(\mathrm{i}(\boldsymbol{x}-\boldsymbol{x}')^\top \Phi^{-1}(\boldsymbol{t})\right)\mathrm{d}\boldsymbol{t} \\
&= \prod_{j=1}^d \left(\int_{-\infty}^{\infty} p_j(\omega^{(j)}) \exp\left(\mathrm{i}\omega^{(j)}(x^{(j)} - x'^{(j)})\right)\mathrm{d}\omega^{(j)}\right) \\
&= (2\pi)^{-d/2} \int_{U_d} \int_0^\infty e^{-\frac{r^2}{2}} |r|^{d-1} g(r\boldsymbol{u})\mathrm{d}r\mathrm{d}\boldsymbol{u} \\
&= \int_{\mathbb{R}^d} q(\boldsymbol{\omega}) \frac{p(\boldsymbol{\omega})}{q(\boldsymbol{\omega})} \exp\left(\mathrm{i}\boldsymbol{\omega}^\top(\boldsymbol{x}-\boldsymbol{x}')\right)\mathrm{d}\boldsymbol{w}
\end{aligned}
$$

data-dependent
- random features selection/learning
- leverage score

iii) Quadrature rules
- GQ, SGQ
- SSR

- **Towards a Unified Quadrature Framework for Large-scale Kernel Machines**, TPAMI2021.
  Fanghui Liu, Xiaolin Huang (SJTU), Yudong Chen (Cornell), Johan A.K. Suykens (KU Leuven)
- **On the Double Descent of Random Features Models Trained with SGD**, arXiv:2110.06910
  Fanghui Liu, Johan A.K. Suykens (KU Leuven), Volkan Cevher (EPFL)

Outline

Background: Numerical integration via quadrature

$$k\left(\mathbf{x},\mathbf{x}'\right) = \mathbb{E}_{\omega\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)} \underbrace{[\sigma(\omega^\top \mathbf{x})\sigma(\omega^\top \mathbf{x}')]}_{\triangleq f(\omega)} := I_d(f) \approx \sum_{i=1}^{N} a_i f(\gamma_i)\,,$$

Quadrature rule: few integration nodes & high polynomial exactness

Gaussian quadrature (GQ)

- construction: one dimensional scheme
- $N = L^d$ nodes for $(2L-1)$-degree rule

$\Rightarrow$ **curse of dimension**

Sparse grid quadrature (SGQ) [3]

- not necessarily use full grid nodes
- how to construct: tensor products

$\Rightarrow N = \texttt{poly}(d)$.

Deterministic Fully symmetric (D-FS) rule: *fully symmetric* concept

$$k(\mathbf{x},\mathbf{y}) := I_d(f) = \int_{\mathbb{R}^d} f(\omega)p(\omega)\mathrm{d}\omega$$

- integration domain $\mathbb{R}^d$
- the Gaussian measure $p(\omega)$

Definition (fully symmetric [4])

unchanged under permutations and sign changes

- A point set/integration domain $\Omega \subset \mathbb{R}^d$ is *fully symmetric* if $(x_1, x_2, \cdots, x_d) \in \Omega$,

$$(\pm x_{i_1}, \pm x_{i_2}, \cdots, \pm x_{i_d}) \in \Omega\,,$$

  where $(i_1, i_2, \cdots, i_d)$ is any permutation of $(1, 2, \ldots, d)$.
- a function g is *fully symmetric* on $\Omega \subset \mathbb{R}^d$ if Ω is fully symmetric set and for any $(x_1, x_2, \cdots, x_d) \in \Omega$

$$g(x_1, x_2, \cdots, x_d) = g(\pm x_{i_1}, \pm x_{i_2}, \cdots, \pm x_{i_d})$$

Deterministic Fully symmetric (D-FS) rule: Definition

Definition (fully symmetric interpolatory rule [5])

Given a generator $\lambda_{\mathbf{p}} = [\lambda_{p_1}, \lambda_{p_2}, \cdots, \lambda_{p_d}]^\top$ with $p_i \in \{0, 1, \ldots, m\}$
$\Pi_{\mathbf{p}}$: permutations of $\mathbf{p}$
$\mathcal{V}_d$: the set of all vectors with sign changes

$$f(\lambda_{\mathbf{p}}) = \sum_{q \in \Pi_{\mathbf{p}}} \sum_{\nu \in \mathcal{V}_d} f\left(\nu_1 \lambda_{q_1}, \nu_2 \lambda_{q_2}, \ldots, \nu_d \lambda_{q_d}\right) .$$

$$I_d(f) \approx Q^{(m,d)}(f) = \sum_{\mathbf{p} \in \mathcal{P}^{(m,d)}} a_{\mathbf{p}}^{(m,d)} f(\lambda_{\mathbf{p}}) .$$

Convergence rate [6]

$$\left\| Q^{(m,d)}(f) - I_d(f) \right\|_{L_2 \text{ or } L_\infty} = \mathcal{O}(N^{-\theta}) ,$$

where θ is some constant.

Kernel approximation via D-FS rules: Example

$$k(\mathbf{x}, \mathbf{x}') \approx Q^{(1,d)}(f) = a_0^{(1,d)} f(\mathbf{0}) + a_1^{(1,d)} \sum_{i=1}^{d} [f(\lambda_1 \mathbf{e}_i) + f(-\lambda_1 \mathbf{e}_i)] \,,$$

with *generator*: $\lambda_0 = 0$ and $\lambda_1 = \sqrt{3}$.

Example: Gaussian kernel

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|_2^2 / (2\sigma^2)\right) \approx \sum_{i=1}^{N} a_i \cos[\omega_i^\top (\mathbf{x} - \mathbf{y})]$$

$$\begin{cases} \text{RFF: } \begin{cases} \text{dense: } \mathbf{W} = [W_{ij}]_{d \times N} \text{ with } W_{ij} \sim \mathcal{N}(0, 1/\sigma^2) \\ a_i \equiv 1/N \end{cases} & \mathcal{O}(Nd) \\ \text{D-FS: } \begin{cases} \text{sparse: } \mathbf{W} = [\gamma_0, \gamma_1, \cdots, \gamma_{2d}] \\ \text{the weight is } a_0 = 1 - \frac{d}{\lambda_1^2} \text{ and } a_i = \frac{1}{2\lambda_1^2} \end{cases} & \mathcal{O}(d) \end{cases}$$

Third degree D-FS rule: Example

$$k(\mathbf{x},\mathbf{x}') \approx Q^{(1,d)}(f) = a_0^{(1,d)} f(\mathbf{0}) + a_1^{(1,d)} \sum_{i=1}^{d} \left[f(\lambda_1 \mathbf{e}_i) + f(-\lambda_1 \mathbf{e}_i) \right] .$$

Example: Gaussian kernel

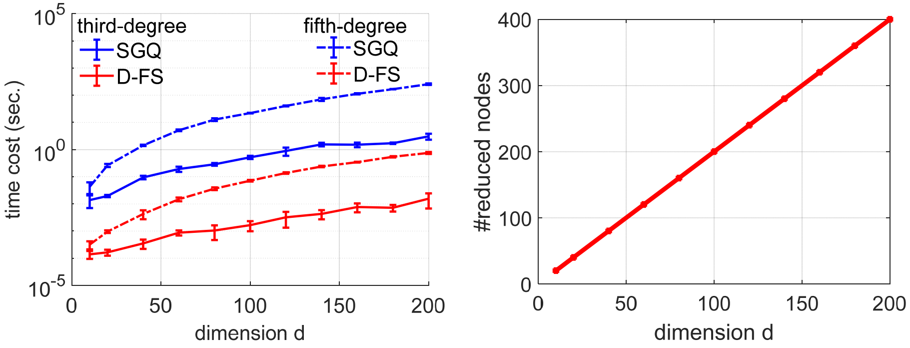
$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\|\mathbf{x}-\mathbf{y}\|_2^2/(2\sigma^2)\right) \approx \sum_{i=1}^{N} a_i \cos[\omega_i^\top (\mathbf{x}-\mathbf{y})]$$

The transformation matrix $\mathbf{W} = [\gamma_0, \gamma_1, \cdots, \gamma_{2d}] \in \mathbb{R}^{d\times(2d+1)}$ is

$$\mathbf{W} = \begin{bmatrix} 0 & -\lambda_1 & \lambda_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & -\lambda_1 & \lambda_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -\lambda_1 & \lambda_1 \end{bmatrix} \in \mathbb{R}^{d\times(2d+1)} .$$

Deterministic fully symmetric (D-FS) rule: Benefit

- time cost
- the number of required nodes: $N_{\text{D-FS}} \leqslant N_{\text{SGQ}}$



(a) Time cost

(b) $N_{\text{SGQ}} - N_{\text{D-FS}}$

Stochastic version: Semi-stochastic rule

D-FS outputs **fixed**-dimensional feature mapping

- third-degree: $N = 2d + 1$
- fifth-degree: $N = 1 + 2d^2$

“semi-stochastic" version

randomize the weights but keep the (deterministic) nodes unchanged

$$
\begin{cases}
a_0 = 1 - \dfrac{d}{\lambda_1^2} \quad \longrightarrow \quad \tilde{a}_0^{(1,d)}(\omega) \equiv 1 - \displaystyle\sum_{i=1}^{d} \omega_i^2 / \lambda_1^2 \\
a_0 = \dfrac{1}{2\lambda_1^2} \quad \longrightarrow \quad \tilde{a}_1^{(1,d)}(\omega) \equiv \displaystyle\sum_{i=1}^{d} \omega_i^2 / (2d\lambda_1^2) \,.
\end{cases}
$$

$$
M^{(1,d)}(f, \omega) = \tilde{a}_0^{(1,d)} f(\mathbf{0}) + \tilde{a}_1^{(1,d)} \sum_{i=1}^{d} \Big[f(\lambda_1 \mathbf{e}_i) + f(-\lambda_1 \mathbf{e}_i) \Big] \,. \tag{1}
$$

Stochastic version: Definition

"semi-stochastic rule"

- still output fixed-dimensional feature mapping
- biased: $\mathbb{E}_\omega[M^{(1,d)}(f,\omega)] = Q^{(1,d)}(f) \neq I_d(f)$.

control variates: $f(\omega) \rightarrow$ difference

$k(\mathbf{x},\mathbf{y}) = Q^{(1,d)}(f) + \mathbb{E}_\omega[f(\omega) - M^{(1,d)}(f,\omega)]$

Stochastic fully-symmetric rule

define $R_1(f,\omega) = Q^{(1,d)}(f) + f(\omega) - M^{(1,d)}(f,\omega)$, third-degree S-FS is

$$k(\mathbf{x},\mathbf{y}) \approx \bar{R}_1(f,\omega) := \frac{1}{D}\sum_{i=1}^{D} R_1(f,\omega_i)\,, \tag{2}$$

with $\{\omega_i\}_{i=1}^D \sim \mathcal{N}(\mathbf{0},\mathbf{I}_d)$.

Stochastic version: Feature mapping

the final feature mapping associated with $\bar{R}_1(f,\omega)$ is given by

$$\widehat{\Phi}(\mathbf{x}) = \left[\varphi(\mathbf{x})^\top, \left(\frac{\mathrm{i}}{D}\sum_{i=1}^{D}\widetilde{\Phi}(\mathbf{x},\omega_i)\right)^\top, \Phi(\mathbf{x})^\top\right]^\top \in \mathbb{R}^{D+4d+2}, \tag{3}$$

where $\{\omega_i\}_{i=1}^D \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$.

- $\varphi(\mathbf{x})^\top$ corresponds to RFF
- $\widetilde{\Phi}(\mathbf{x},\omega_i)$ corresponds to "semi-stochastic" version $M^{(1,d)}(f)$
- $\Phi(\mathbf{x})$ corresponds to deterministic version $Q^{(1,d)}(f)$

Stochastic version: Statistical properties

Unbiased

$k(\mathbf{x},\mathbf{y}) := I_d(f) = \mathbb{E}_\omega \bar{R}_1(f,\omega)$.

Variance reduction

For Gaussian kernel $k(\mathbf{x},\mathbf{y}) = \exp\left(-\|\mathbf{x}-\mathbf{y}\|_2^2/(2\sigma^2)\right)$, denoting $z := \|\mathbf{z}\|_2$ with $\mathbf{z} := (\mathbf{x}-\mathbf{y})/\sigma$, $Q := Q^{(1,d)}(f)$,we have

$$\mathbb{V}[\bar{R}_1(f,\omega)] - \mathbb{V}[\text{RFF}] = \frac{2}{Dd} \underbrace{\left(\left[(1-Q) - \frac{1}{2} z^2 e^{-\frac{z^2}{2}}\right]^2 - \frac{1}{4} z^4 e^{-z^2}\right)}_{\triangleq h_{\text{S-FS}}(\mathbf{z})}, \tag{4}$$

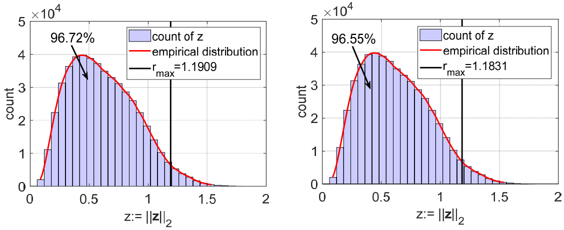
which implies $\mathbb{V}[\bar{R}_1(f,\omega)] - \mathbb{V}[\text{RFF}] < 0$ when $1 - Q < z^2 e^{-\frac{z^2}{2}}$.

Stochastic version: Condition validation

$$\mathbb{V}[\bar{R}_1(f,\omega)] - \mathbb{V}[\text{RFF}] < 0 \quad \text{when} \quad 1 - Q < z^2 e^{-\frac{z^2}{2}}\,.$$

$$\Longleftrightarrow \frac{d}{3} - \frac{1}{3}\sum_{i=1}^{d} \cos(\sqrt{3}\mathbf{e}_i^\top \mathbf{z}) - \|\mathbf{z}\|_2^2 \exp(-\|\mathbf{z}\|_2^2/2) < 0\,.$$

Existence: find a hyper-ball $\mathcal{S}^d(r_{\max}) := \{\mathbf{z} \in \mathbb{R}^d : \|\mathbf{z}\|_2 \le r_{\max}\}$
(a one-dimensional optimization to solve $r_{\max}$)

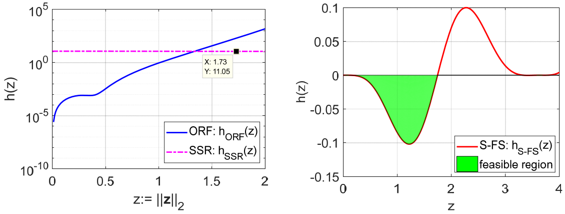


(c) ijcnn1: $d = 22$

(d) covtype: $d = 54$

Figure: Empirical distribution of $\|\mathbf{z}\|_2$

Stochastic version: Comparison



(a) $d = 10$ (b) $d = 1$

Figure: Comparison of ORF[2], SSR[3] (a) and S-FS (b).

$$\mathbb{V}[\text{ORF}] - \mathbb{V}[\text{RFF}] \leq \frac{1}{D} h_{\text{ORF}}(z) \quad \text{[an exp. growth function]}$$

$$\mathbb{V}[\text{SSR}] - \mathbb{V}[\text{RFF}] \leq \frac{1}{D} h_{\text{SSR}}(z)\,, \quad \text{[at } \mathcal{O}(1) \text{ order]}$$

[2] Yu et al. Orthogonal random features. NeurIPS2016.

[3] Munkhoeva et al. Quadrature-based features for kernel approximation. NeurIPS2018.

Unified framework

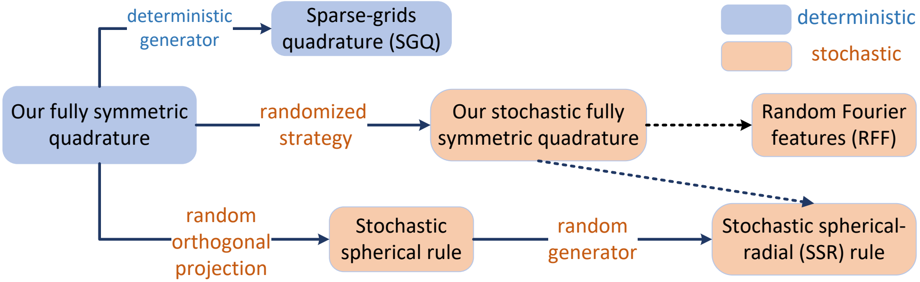


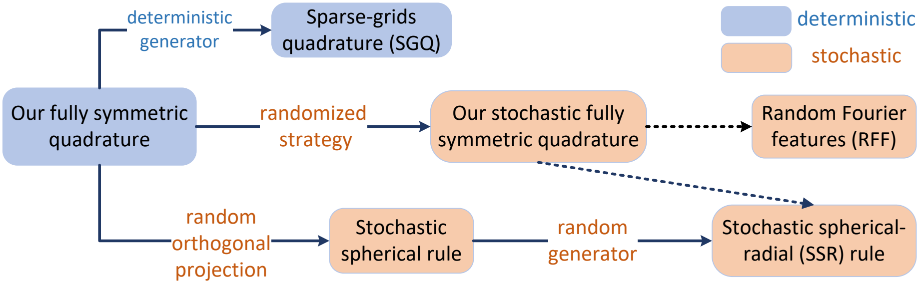
Figure: Relationship between quadrature based methods.

SGQ: nodes: $\{-\hat{p}_1, 0, \hat{p}_1\}$ and weights $(\hat{a}_1, \hat{a}_0, \hat{a}_1)$,

$$I_d(f) \approx (1-d+d\hat{a}_0)\, f(\mathbf{0}) + \hat{a}_1 \sum_{j=1}^{d} \left[f(\hat{p}_1 \mathbf{e}_j) + f(-\hat{p}_1 \mathbf{e}_j) \right].$$

by taking $\hat{a}_0 := 1 - \frac{1}{\lambda_1^2},\ \hat{p}_1 := \lambda_1,\ \hat{a}_1 = \frac{1}{2\lambda_1^2}$.

Unified framework



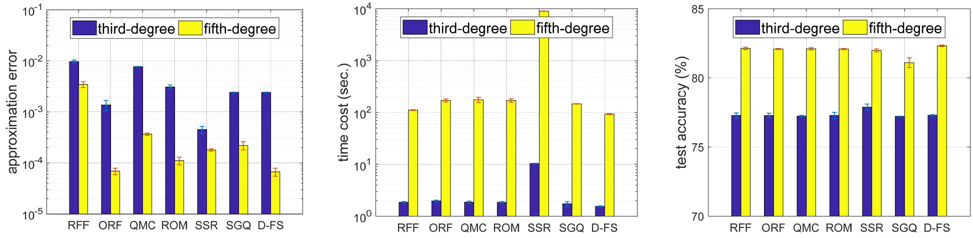
SSR: $$I_d(f) \approx f(\mathbf{0})\left(1-\frac{d}{\rho^2}\right) + \sum_{j=1}^{d} \frac{f(-\rho \mathbf{Q} \mathbf{e}_j) + f(\rho \mathbf{Q} \mathbf{e}_j)}{2\rho^2}, \quad \rho \sim \chi(d+2).$$

stochastic spherical rule: $I_{\mathbf{Q},U_d}(f) = \frac{|U_d|}{2d} \sum_{j=1}^{d} \left[f(\mathbf{Q}\mathbf{e}_j) + f(-\mathbf{Q}\mathbf{e}_j) \right]$.

Experimental results: Evaluation on deterministic rules

Compared methods

RFF ([7] NeurIPS2007): Monte Carlo sampling from $p(\omega)$
QMC ([8] JMLR2016): a low-discrepancy Halton sequence
Orthogonal constraint: ORF ([9] NeurIPS2016), ROM([10] NeurIPS2017)
Quadrature methods: SGQ([11] NeurIPS2017), SSR([12] NeurIPS2018)

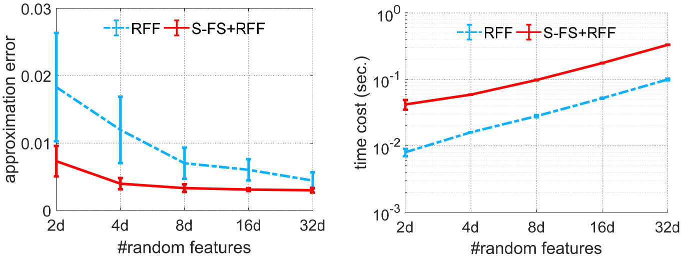


(a) Approximation error (b) Time cost (c) Test accuracy

Figure: Results on the *covtype* dataset with $n = 581,012$ across Gaussian kernel.

Experimental results: Variance reduction of stochastic rules

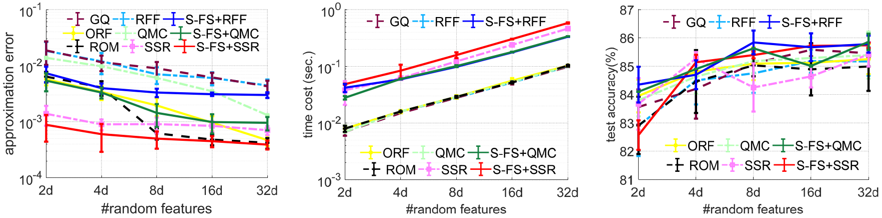
adaptive feature mapping dimension: $D = \{2d, 4d, 8d, 16d, 32d\}$.



(a) approximation error (b) time cost

Figure: Benefits of our S-FS rule in Eq. (3) against RFF across the Gaussian kernel on the *magic04* data set.

Experimental results: Evaluation on stochastic rules



(a) approximation error (b) time cost (c) test accuracy

Figure: Results on the *magic04* dataset across the Gaussian kernel.

- reduction on approximation error
- in the same time complexity
- no difference on generalization performance

Outline

Background: Double descent

over-parameterized models, e.g., neural networks, random features

- high dimensions: large n and d
- abnormal phenomena: training error can be zero but still generalize well

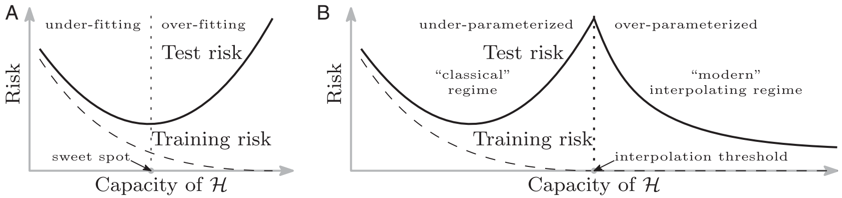


Figure: Bias-variance trade-off [13] (Belkin et al. PNAS2019).

Research Overview: Motivation

- interplay between optimization and excess risk: trained by SGD
- bias-variance decomposition for understanding multiple randomness sources

	data assumption	solution	result
(Hastie et al., 2019)	Gaussian	closed-form	variance $\nearrow\searrow$
(Ba et al., 2020)	Gaussian	GD	variance $\nearrow\searrow$
(Mei & Montanari, 2019)	i.i.d on sphere	closed-form	variance, bias $\nearrow\searrow$
(d'Ascoli et al., 2020a)	Gaussian	closed-form	refined [2]
(Gerace et al., 2020)	Gaussian	closed-form	$\nearrow\searrow$
(Adlam & Pennington, 2020)	Gaussian	closed-form	refined
(Dhifallah & Lu, 2020)	Gaussian	closed-form	$\nearrow\searrow$
(Hu & Lu, 2020)	Gaussian	closed-form	$\nearrow\searrow$
(Liao et al., 2020)	general	closed-form	$\nearrow\searrow$
(Lin & Dobriban, 2021)	isotropic features with finite moments	closed form	refined
(Li et al., 2021)	correlated features with polynomial decay on Σ_d	closed form	interpolation learning
Ours	(at least) sub-exponential data	SGD	variance $\nearrow\searrow$, bias $\searrow$

[1] A refined decomposition on variance is conducted by sources of randomness on data sampling, initialization, label noise to possess each term (d'Ascoli et al., 2020b) or their full decomposition in (Adlam & Pennington, 2020; Lin & Dobriban, 2021).

Problem settings: Random features regression model

data: $y = f_\rho(\mathbf{x}) + \varepsilon$

- training data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim \rho$
  Assumption: sub-exponential data and $\|\mathbf{x}\|_2^2 \sim \mathcal{O}(d)$
- target function: $f_\rho(\mathbf{x}) = \int_Y y \, \mathrm{d}\rho(y \mid \mathbf{x})$
- noise: $\mathbb{E}(\varepsilon) = 0$ and $\mathbb{E}(\varepsilon^2) = \tau^2$

function space

define the random features mapping $\varphi(\mathbf{x}) := \frac{1}{\sqrt{m}} \sigma(\mathbf{W}\mathbf{x}/\sqrt{d})$,

$$\mathcal{H} := \left\{ f \in L^2_{\rho_X} \,\middle|\, f(\mathbf{x}) = \langle \theta, \varphi(\mathbf{x}) \rangle \right\}, \quad \mathbf{W}_{ij} \sim \mathcal{N}(0,1)$$

covariance operator: $\Sigma_m := \int_X [\varphi(\mathbf{x}) \otimes \varphi(\mathbf{x})] \mathrm{d}\rho_X(\mathbf{x})$

expected covariance operator: $\widetilde{\Sigma}_m := \mathbb{E}_{\mathbf{x},\mathbf{W}} [\varphi(\mathbf{x}) \otimes \varphi(\mathbf{x})]$

Problem settings: averaged SGD under adaptive step-size setting

$$\theta_t = \theta_{t-1} + \gamma_t [y_t - \langle \theta_{t-1}, \varphi(\mathbf{x}_t) \rangle] \varphi(\mathbf{x}_t), \qquad t = 1, 2, \ldots n\,,$$

- averaged output: $\bar{\theta}_n := \frac{1}{n} \sum_{t=0}^{n-1} \theta_t \Longrightarrow \bar{f}_n = \langle \varphi(\cdot), \bar{\theta}_n \rangle$
- adaptive step-size: $\gamma_t := \gamma_0 t^{-\zeta}, \zeta \in [0, 1)$
- optimal solution: $f^* = \arg\min_{f \in \mathcal{H}} \| f - f_\rho \|^2_{L^2_{\rho_X}}$
- averaged excess risk: $\mathbb{E} \| \bar{f}_n - f^* \|^2_{L^2_{\rho_X}} = \mathbb{E}_{\mathbf{X}, \mathbf{W}, \varepsilon} \langle \bar{f}_n - f^*, \Sigma_m (\bar{f}_n - f^*) \rangle$

Properties of covariance operators

$\sigma(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ Lipschitz continuous
covariance operator $\Sigma_m := \mathbb{E}_{\mathbf{x}}[\varphi(\mathbf{x}) \otimes \varphi(\mathbf{x})]$
expected covariance operator $\widetilde{\Sigma}_m := \mathbb{E}_{\mathbf{x},\mathbf{W}}[\varphi(\mathbf{x}) \otimes \varphi(\mathbf{x})]$

eigenvalue of $\widetilde{\Sigma}_m$

the same diagonal/non-diagonal elements: $\mathcal{O}(1/m)$
two distinct eigenvalues: $\widetilde{\lambda}_1 \sim \mathcal{O}(1)$, $\widetilde{\lambda}_2 \sim \mathcal{O}(1/m)$

sub-exponential random variables

$\|\Sigma_m\|_2$, $\|\Sigma_m - \widetilde{\Sigma}_m\|_2$, $\mathrm{Tr}(\Sigma_m)$, and $\left\|\widetilde{\Sigma}_m^{-1}\mathbb{E}_{\mathbf{W}}(\Sigma_m^2)\right\|_2$ with $\mathcal{O}(1)$ sub-exponential norm order

Proof framework

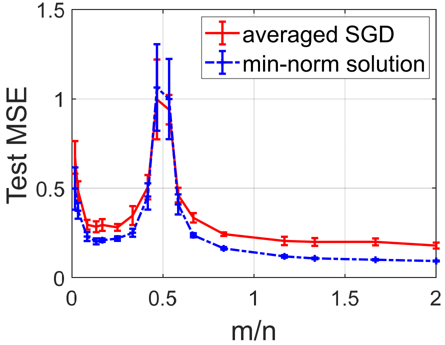
- excess risk $\mathbb{E}_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{\varepsilon}}\langle \bar{\eta}_n, \Sigma_m \bar{\eta}_n \rangle$
  - Variance $\mathbb{E}_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{\varepsilon}}\langle \bar{\eta}_n^{\texttt{var}}, \Sigma_m \bar{\eta}_n^{\texttt{var}} \rangle$
    - V1: $\bar{\eta}_n^{\texttt{var}} - \bar{\eta}_n^{\texttt{vX}}$ $\begin{cases} \mathcal{O}(n^{\zeta-1}m) & \text{if } m \leqslant n \\ \mathcal{O}(1) & \text{if } m \geqslant n \end{cases}$
    - V2: $\bar{\eta}_n^{\texttt{vX}} - \bar{\eta}_n^{\texttt{vXW}}$ $\begin{cases} \mathcal{O}(n^{\zeta-1}m) \\ \mathcal{O}(1) \end{cases}$
    - V3: $\bar{\eta}_n^{\texttt{vXW}}$ $\begin{cases} \mathcal{O}(n^{\zeta-1}m) \\ \mathcal{O}(n^{\zeta-1} + \frac{n}{m}) \end{cases}$
  - Bias $\mathbb{E}_{\boldsymbol{X},\boldsymbol{W}}\langle \bar{\eta}_n^{\texttt{bias}}, \Sigma_m \bar{\eta}_n^{\texttt{bias}} \rangle$
    - B1: $\bar{\eta}_n^{\texttt{bias}} - \bar{\eta}_n^{\texttt{bX}}$ $\mathcal{O}(n^{\zeta-1})$
    - B2: $\bar{\eta}_n^{\texttt{bX}} - \bar{\eta}_n^{\texttt{bXW}}$ $\begin{cases} \mathcal{O}(n^{2(\zeta-1)}m) \\ \mathcal{O}(\frac{1}{m}) \end{cases}$
    - B3: $\bar{\eta}_n^{\texttt{bXW}}$ $\mathcal{O}(n^{\zeta-1})$

Findings:

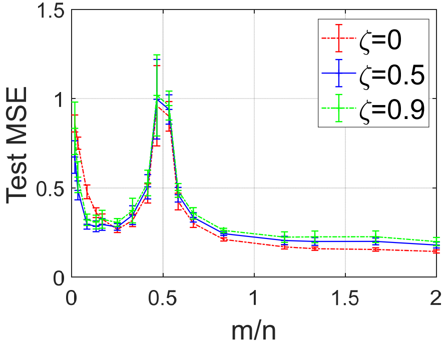
- expected covariance operator $\widetilde{\Sigma}_m$ has only two distinct eigenvalues
- monotonic bias and unimodal variance
- same convergence rates: constant step-size SGD vs. min-norm solution

Experiments on MNIST

Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2d}\right)$



(a) SGD vs. min-norm solution

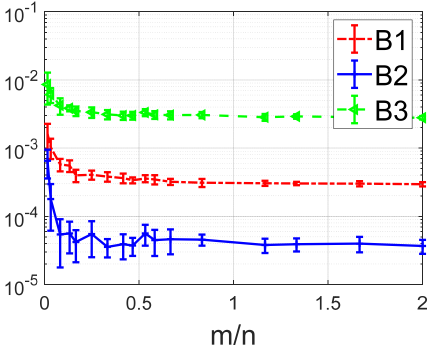


(b) step-size

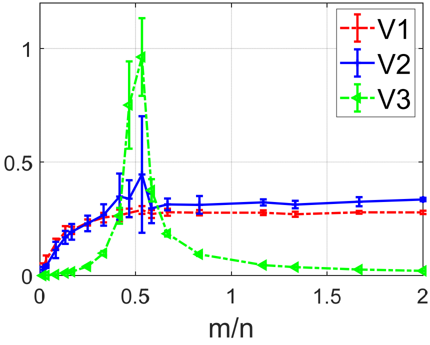
Figure: Test MSE (mean±std.) of RF regression as a function of the ratio m/n on MNIST data set (digit 3 vs. 7) for $d = 784$ and $n = 600$.

Validation for bias and variance

- noise: $\varepsilon \sim \mathcal{N}(0,1)$
- Σ_m, $\widetilde{\Sigma}_m$: sample covariance matrices with Monte Carlo sampling



(a) Bias



(b) Variance

Outline

Take-away message

- a unified framework for quadrature rules
- algorithm
  - *deterministic*: low complexity and approximation error
  - *stochastic*: dimension-adaptive feature mapping
- theory: unbiasedness and variance reduction

- high dimensional random features model trained by SGD
- findings
  - bias-variance decomposition: multiple randomness sources
  - monotonic decreasing bias and unimodal variance
  - optimization effect on excess risk

Future works:

- applications for high dimensional integration
- random features model in deep learning theory

Thanks for your attention!

Q & A

my homepage http://lfhsgre.org for more information!

NEW: ERC Advanced Grant E-DUALITY

Exploring duality for future data-driven modelling

References I

[1] Salomon Bochner.
Harmonic Analysis and the Theory of Probability.
Courier Corporation, 2005.

[2] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, and Weller Adrian.
Rethinking attention with performers.
In *International Conference on Learning Representations*, 2021.

[3] Florian Heiss and Viktor Winschel.
Likelihood approximation by numerical integration on sparse grids.
Journal of Econometrics, 144(1):62–80, 2008.

[4] Philip J. Davis and Philip Rabinowitz.
Methods of numerical integration.
Courier Corporation, 2007.

[5] Alan Genz and Bradley D Keister.
Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight.
Journal of Computational and Applied Mathematics, 71(2):299–309, 1996.

References II

[6] Aicke Hinrichs and Erich Novak.
Cubature formulas for symmetric measures in higher dimensions with few points.
Mathematics of computation, 76(259):1357–1372, 2007.

[7] Ali Rahimi and Benjamin Recht.
Random features for large-scale kernel machines.
In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2007.

[8] Haim Avron, Vikas Sindhwani, Jiyan Yang, and Michael W. Mahoney.
Quasi-Monte Carlo feature maps for shift-invariant kernels.
Journal of Machine Learning Research, 17(1):4096–4133, 2016.

[9] Felix Xinnan Yu, Ananda Theertha Suresh, Krzysztof Choromanski, Daniel Holtmannrice, and Sanjiv Kumar.
Orthogonal random features.
In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016.

[10] Krzysztof M. Choromanski, Mark Rowland, and Adrian Weller.
The unreasonable effectiveness of structured random orthogonal embeddings.
In *Advances in Neural Information Processing Systems*, pages 219–228, 2017.

References III

[11] Tri Dao, Christopher M. De Sa, and Christopher Ré.
Gaussian quadrature for kernel features.
In *Advances in neural information processing systems*, pages 6107–6117, 2017.

[12] Marina Munkhoeva, Yermek Kapushev, Evgeny Burnaev, and Ivan Oseledets.
Quadrature-based features for kernel approximation.
In *Advances in Neural Information Processing Systems*, pages 9147–9156, 2018.

[13] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal.
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
the National Academy of Sciences, 116(32):15849–15854, 2019.