Spherical Structured Feature Maps for Kernel Approximation

Yueming Lyu [1]

Abstract

We propose Spherical Structured Feature (SSF) maps to approximate shift and rotation invariant kernels as well as b^{th}-order arc-cosine kernels (Cho & Saul, 2009). We construct SSF maps based on the point set on $d-1$ dimensional sphere $\mathbb{S}^{d-1}$. We prove that the inner product of SSF maps are unbiased estimates for above kernels if asymptotically uniformly distributed point set on $\mathbb{S}^{d-1}$ is given. According to (Brauchart & Grabner, 2015), optimizing the discrete Riesz s-energy can generate asymptotically uniformly distributed point set on $\mathbb{S}^{d-1}$. Thus, we propose an efficient coordinate decent method to find a local optimum of the discrete Riesz s-energy for SSF maps construction. Theoretically, SSF maps construction achieves linear space complexity and loglinear time complexity. Empirically, SSF maps achieve superior performance compared with other methods.

1. Introduction

Kernel methods such as Gaussian processes (GPs) (Rasmussen, 2006; Srinivas et al., 2009; Snoek et al., 2012) and support vector machines (SVMs) (Chang & Lin, 2011; Fan et al., 2008) have been successfully used in many statistical modeling and machine learning tasks. Despite of strong expressive power, kernel methods usually cannot scale up to the large scale datasets with L samples due to the need of manipulating $L\times L$ Gram matrix. Recently, random feature maps (Rahimi et al., 2007; Rahimi & Recht, 2009; Sutherland & Schneider, 2015) have demonstrated their effectiveness on kernel approximation to scale up kernel methods. Roughly speaking, a shift invariant kernel $\mathrm{K}(\mathbf{x},\mathbf{z}) = \mathbf{K}(\mathbf{x}-\mathbf{z}) : \mathbb{R}^{\mathbf{d}} \rightarrow \mathbb{C}$ can be approximated by $\mathrm{K}(x,z) \approx \Psi(\mathbf{x})^{\mathbf{T}}\Psi(\mathbf{z})$, where Ψ is the explicit mapped feature constructed as $\Psi(\mathbf{x}) = f(\mathbf{W}^T\mathbf{x})/\sqrt{N}$, where $f(\cdot)$ denotes the nonlinear function, $\mathbf{W} \in \mathbb{R}^{d\times N}$ is constructed by N i.i.d samples drawn from a distribution defined by K. Therefore, the training and inference of kernel methods can be greatly accelerated by working directly on the primal space of $\Psi(\cdot)$. For example, Gaussian Processes (GPs) have $O(L^3)$ computation and $O(L^2)$ storage complexity. By using feature maps, it reduces to $O(N^2L+N^3)$ computation and $O(NL+N^2)$ storage complexity. All these elegant properties make random feature maps promising for large scale kernel methods. Thus, many kernel methods (Le & Wilson, 2015; Cutajar et al., 2016; Oliva et al., 2016) have been proposed to deal with large scale statistical learning by directly working on feature maps.

Generally, two aspects of random feature maps are mostly concerned by literature for scaling up kernel methods. One is the approximation accuracy of feature maps while the other is the computational cost of feature maps construction. To achieve better approximation accuracy, (Yang, 2014; Avron et al., 2016) employ QMC (Dick et al., 2013) sampling instead of standard Monte Carlo sampling to construct feature maps. By mapping QMC points on $[0,1]^d$ through the inverse cumulative distribution function, they construct more effective feature maps. To reduce time complexity, (Le et al., 2013) propose Fastfood to construct feature maps. Benefiting from the special structured matrix multiplication, it reduces time complexity of feature maps construction from $O(Nd)$ to $O(N\log d)$. However, it achieves computational efficiency at the expense of increasing the variance of approximation. Recently, (Feng et al., 2015) employ the property of circulant matrix to accelerate feature maps construction of Gaussian kernel without increasing the variance. (Choromanski & Sindhwani, 2016) generalize the Fastfood and circulant feature maps to P model and particularly discuss the structured matrix with low-displacement rank. Despite of the success of P model, it still cannot achieve better approximation accuracy compared with feature maps obtained with fully Gaussian matrix.

To achieve better approximation accuracy and loglinear time complexity, we propose Spherical Structured Feature (SSF) maps to approximate shift and rotation invariant kernels as well as b^{th}-order arc-cosine kernels (Cho & Saul, 2009). Specifically, We construct SSF maps based on the point set on $d-1$ dimensional sphere $\mathbb{S}^{d-1}$, where the

[1] Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Hong Kong . Correspondence to: Yueming Lyu <LV_Yueming@outlook.com>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

points are columns of a particular structured matrix produced by a discrete Fourier matrix. The points on $\mathbb{S}^{d-1}$ for SSF maps construction can be generated by optimizing the discrete Riesz s-energy. According to (Brauchart et al., 2014), optimizing the discrete Riesz s-energy (for s in some ranges) can generate QMC designs on $\mathbb{S}^{d-1}$, which usually can achieve smaller approximation error compared with fully random methods. Moreover, Because of special structure of the point set, SSF maps construction can achieve loglinear time complexity via Fast Fourier Transform (FFT).

Our contributions are summarized as follows:

- We propose Spherical Structured Feature (SSF) maps to approximate shift and rotation invariant kernels as well as b^{th}-order arc-cosine kernels (Cho & Saul, 2009). We prove that the inner product of SSF maps are unbiased estimates for above kernels if asymptotically uniformly distributed point set on $d-1$ dimensional sphere $\mathbb{S}^{d-1}$ is given.
- We propose an efficient coordinate decent method to find a local optimum of the discrete Riesz s-energy (Brauchart & Grabner, 2015), thereby approximately generating asymptotically uniformly distributed points on $\mathbb{S}^{d-1}$.
- We can construct SSF maps with linear space complexity and loglinear time complexity. Empirically, SSF maps achieve superior performance compared with other methods.

2. Background and Preliminaries

We provide a brief review of random feature maps and the discrete Riesz s-energy in this section as preliminaries.

2.1. Random Feature Maps

Random feature maps can be viewed as equal weight approximation of multidimensional integrals. One earlier work (Rahimi et al., 2007) approximates the shift invariant kernels based on the Bochner's Theorem.

Theorem 2.1 Bochner's Theorem ((Rudin, 2011)) : A continuous shift invariant scaled kernel function $\mathrm{K}(\mathbf{x}, \mathbf{z}) = \mathbf{K}(\mathbf{x}-\mathbf{z}) : \ \mathbb{R}^{\mathbf{d}} \to \mathbb{C}$ is positive definite if and only if it is the Fourier Transform of a unique finite probability measure p on $\mathbb{R}^d$.

$$\mathrm{K}(\mathbf{x}, \mathbf{z}) = \int_{\mathbb{R}^{\mathbf{d}}} \mathbf{e}^{-\mathbf{i}(\mathbf{x}-\mathbf{z})^{\mathbf{T}}\mathbf{w}} \mathbf{p}(\mathbf{w}) \mathbf{dw} \tag{1}$$

For a real valued kernel $\mathrm{K}(\mathbf{x}, \mathbf{z})$, $p(\mathbf{w}) = p(-\mathbf{w}) \geq 0$ can ensure the imaginary parts of the integral vanish. According to the Bochner's theorem, there is a one-to-one correspondence between the kernel functions $\mathrm{K}(\mathbf{x}, \mathbf{z})$ and probability densities $p(\mathbf{w})$ defined on $\mathbb{R}^d$.

Shift and rotation invariant kernels are shift invariant kernels with the rotation invariant property, i.e. $K(\mathbf{x}, \mathbf{z}) = K(R\mathbf{x}, R\mathbf{z})$, given any rotation $R \in SO(d)$, where $SO(d)$ denotes rotation groups. The Gaussian kernel $K(\mathbf{x}, \mathbf{z}) = e^{-\|\mathbf{x}-\mathbf{z}\|_2^2/2\sigma^2}$ is a member of this family. From Bochner's theorem, the corresponding probability density is also Gaussian. For a general Gaussian RBF kernel $K(\mathbf{x}, \mathbf{z}) = e^{-(\mathbf{x}-\mathbf{z})^T \Sigma (\mathbf{x}-\mathbf{z})/2}$, it can be transformed into rotation invariant form by using $\mathbf{y} = \Sigma^{1/2}\mathbf{x}$ in the original domain.

$\mathbf{b^{th}}$**-order arc-cosine kernels** are rotation invariant kernels. As discussed in (Cho & Saul, 2009), b^{th}-order arc-cosine kernels have the following form:

$$K_b(\mathbf{x}, \mathbf{z}) = \tfrac{1}{\pi} \|\mathbf{x}\|_2^b \, \|\mathbf{z}\|_2^b \, J_b(\theta) \tag{2}$$

where $\theta = \cos^{-1}\left(\frac{\mathbf{x}^T\mathbf{z}}{\|\mathbf{x}\|_2 \|\mathbf{z}\|_2}\right)$

b^{th}-order arc-cosine kernels have trivial dependence on the norm of $\mathbf{x}$ and $\mathbf{z}$. The dependence on the angle is defined by function $J_b(\theta)$. b^{th}-order arc-cosine kernels are rotation invariant kernels but not shift invariant kernels in general. For example, the zero-order (3) and first-order (4) arc-cosine kernel are not shift invariant kernels.

$$K_0(\mathbf{x}, \mathbf{z}) = 1 - \tfrac{\theta}{\pi} \tag{3}$$

$$K_1(\mathbf{x}, \mathbf{z}) = \tfrac{1}{\pi} \|\mathbf{x}\|_2 \, \|\mathbf{z}\|_2 \left(\sin\theta + (\pi - \theta)\cos\theta\right) \tag{4}$$

The b^{th}-order arc-cosine kernel $\mathrm{K}_b(\mathbf{x}, \mathbf{z})$ can be reformulated via the integral representation:

$$\mathrm{K}_b(\mathbf{x}, \mathbf{z}) = 2\int_{\mathbb{R}^d} s(\mathbf{w}^T\mathbf{x}) s(\mathbf{w}^T\mathbf{z}) (\mathbf{w}^T\mathbf{x})^b (\mathbf{w}^T\mathbf{z})^b p(\mathbf{w}) d\mathbf{w} \tag{5}$$

where $s(\cdot)$ is a step function (i.e. $s(x) = 1$ if $x > 0$ and 0 otherwise) and the density p is standard Gaussian.

Feature maps: Both Monte Carlo and Quasi-Monte Carlo approximation (Dick et al., 2013) are equal weight approximation to integrals. Based on equal weight approximation, the feature maps can be constructed as:

$$K(\mathbf{x}, \mathbf{z}) \approx \frac{1}{N} \sum_{i=1}^{N} f\left(\mathbf{w}_i^T\mathbf{x}\right) f\left(\mathbf{w}_i^T\mathbf{x}\right) = \Psi(\mathbf{x})^T \Psi(\mathbf{z}) \tag{6}$$

where $\mathbf{w}_i, i \in 1, ..., N$ are samples constructed by Monte Carlo or Quasi-Monte Carlo methods. $f(\cdot)$ is a nonlinear function depending on the kernel. $\Psi(\cdot)$ is the explicit finite dimensional feature map. For Gaussian kernel with bandwidth σ, the associated nonlinear function is a complex exponential function $f(x) = e^{ix/\sigma}$. For a zero-order arc-cosine kernel in (3) and first-order arc-cosine kernel in (4), the associated nonlinear functions are step function $f(x) = s(x)$ and ReLU activation function $f(x) = max(0, x)$ respectively.

2.2. Discrete Riesz s-energy

The discrete Riesz s-energy is related to the equal weight numerical integration and uniformly distributed point set.

Equal weight numerical integration over a d-dimensional sphere $\mathbb{S}^d := \{\mathbf{x} \in \mathbb{R}^{d+1} \mid \|\mathbf{x}\|_2 = 1\}$ uses equal weight summation of finite point evaluations of the integrands to approximate the integrals:

$$\int_{\mathbb{S}^d} f(\mathbf{v}) d\sigma(\mathbf{v}) \approx \tfrac{1}{N} \sum_{i=1}^{N} f(\mathbf{v}_i) \tag{7}$$

where σ denotes the normalized surface area measure on $\mathbb{S}^d$.

According to (Brauchart & Grabner, 2015), the point set $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N] \in \mathbb{S}^{d\times N}$ is asymptotically uniformly distributed if equation (8) holds true.

$$\lim_{N\to\infty} \tfrac{1}{N} \sum_{i=1}^{N} f(\mathbf{v}_i) = \int_{\mathbb{S}^d} f(\mathbf{v}) d\sigma(\mathbf{v}) \tag{8}$$

The discrete Riesz s-energy(Götz, 2003; Brauchart & Grabner, 2015) is defined as equation (9):

$$E_s(\mathbf{V}) := \begin{cases} \sum\limits_{i=1}^{N} \sum\limits_{j=1,j\neq i}^{N} \frac{1}{\|\mathbf{v}_i - \mathbf{v}_j\|_2^s} & , s \neq 0 \\ \sum\limits_{i=1}^{N} \sum\limits_{j=1,j\neq i}^{N} \log \frac{1}{\|\mathbf{v}_i - \mathbf{v}_j\|_2}, & s = 0 \end{cases} \tag{9}$$

Theorem 2.2 ((Brauchart & Grabner, 2015)): For $s > -2$, the optimum N-point configuration of the Riesz s-energy on $\mathbb{S}^d$ is asymptotically uniformly distributed w.r.t the normalized surface area measure σ on $\mathbb{S}^d$.

According to (Brauchart et al., 2014; Brauchart & Grabner, 2015), the discrete Riesz s-energy can serve as a criterion to construct the point set $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N] \in \mathbb{S}^{d\times N}$ for QMC designs. Particularly, (Brauchart et al., 2014) have proved that maximizing the discrete Riesz s-energy with $s \in (-2, 0)$ can generate QMC designs for functions in Sobolev space. They also prove that QMC designs have higher convergence rate of worst-case error than fully randomly chosen points for functions in Sobolev space.

3. Spherical Structured Feature Maps

In this section, we propose SSF maps to approximate shift and rotation invariant kernels as well as b^{th}-order arc-cosine kernels by employing their rotation invariant property.

3.1. Feature Maps for Shift and Rotation Invariant Kernels

Shift and rotation invariant kernels are highly symmetric and structured because they satisfy both shift invariant property and rotation invariant property. Rotation invariant property means that $K(\mathbf{x}, \mathbf{z}) = K(R\mathbf{x}, R\mathbf{z})$, given any rotation $R \in SO(d)$, where $SO(d)$ denotes rotation groups. To benefit from rotation invariant property, it is reasonable to construct the feature maps by using spherical equal weight approximation in equation (7) and (8).

The feature maps for real valued shift and rotation invariant kernels $K(\mathbf{x}, \mathbf{z})$ can be constructed as equation (10):

$$\begin{aligned} \Psi(\mathbf{x}) = \tfrac{1}{\sqrt{NM}} [&\cos\left(\Phi^{-}(t_1)\mathbf{x}^T\mathbf{v}_1\right), \sin\left(\Phi^{-}(t_1)\mathbf{x}^T\mathbf{v}_1\right), \\ &..., \cos\left(\Phi^{-}(t_M)\mathbf{x}^T\mathbf{v}_N\right), \sin\left(\Phi^{-}(t_M)\mathbf{x}^T\mathbf{v}_N\right)]^T \end{aligned} \tag{10}$$

where $t_j = \frac{j}{M+1}$, $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N] \in \mathbb{S}^{d-1\times N}$ denotes the point set asymptotically uniformly distributed on $\mathbb{S}^{d-1}$and $\Phi^{-}(x)$ denotes the inverse cumulative distribution function w.r.t the nonnegative radial scale.

Theorem 3.1: $\Psi(\mathbf{x})^T\Psi(\mathbf{z})$ is an unbiased estimate of a real valued shift and rotation invariant kennel $\mathrm{K}(\mathbf{x}, \mathbf{z})$.

Proof: From Bochner's Theorem, a shift invariant kernel $K(\mathbf{x}, \mathbf{z})$ can be written as equation (1). Let $r = \|\mathbf{w}\|_2$and $p(r)$ be the density function of r. Because of the rotation invariant property of $K(\mathbf{x}, \mathbf{z})$, we achieve equation (11).

$$\begin{aligned} \mathrm{K}(\mathbf{x}, \mathbf{z}) &= \int_{\mathbb{R}_+} \int_{\mathbb{S}^{d-1}} e^{-ir(\mathbf{x}-\mathbf{z})^T\mathbf{v}} p(r) dr d\sigma(\mathbf{v}) \\ &= \int_{[0,1]} \int_{\mathbb{S}^{d-1}} e^{-i\,\Phi^{-}(t)(\mathbf{x}-\mathbf{z})^T\mathbf{v}} d\sigma(\mathbf{v}) dt \end{aligned} \tag{11}$$

where $\mathbb{R}_+$ denotes the nonnegative real values.

For real valued kernel $K(\mathbf{x}, \mathbf{z})$, the imaginary parts of the integral vanish. We can achieve equation (12).

$$\mathrm{K}(\mathbf{x}, \mathbf{z}) = \int_{[0,1]} \int_{\mathbb{S}^{d-1}} \cos\left(\Phi^{-}(t)(\mathbf{x}-\mathbf{z})^T\mathbf{v}\right) d\sigma(\mathbf{v}) dt \tag{12}$$

According to the property of asymptotically uniformly distributed point set $\mathbf{V}$ in equation (8) and the one-dimensional QMC rule, we obtain equation (13).

$$\begin{aligned} &\lim_{M,N\to\infty} \Psi(\mathbf{x})^T\Psi(\mathbf{z}) = \\ &\lim_{M,N\to\infty} \tfrac{1}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} (\cos\left(\Phi^{-}(t_j)\mathbf{x}^T\mathbf{v}_i\right) \cos\left(\Phi^{-}(t_j)\mathbf{z}^T\mathbf{v}_i\right) \\ &\qquad + \sin\left(\Phi^{-}(t_j)\mathbf{x}^T\mathbf{v}_i\right) \sin\left(\Phi^{-}(t_j)\mathbf{z}^T\mathbf{v}_i\right)) \\ &= \lim_{M,N\to\infty} \tfrac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} \cos\left(\Phi^{-}(t_j)(\mathbf{x}-\mathbf{z})^T\mathbf{v}_i\right) \\ &= \int_{[0,1]} \int_{\mathbb{S}^{d-1}} \cos\left(\Phi^{-}(t)(\mathbf{x}-\mathbf{z})^T\mathbf{v}\right) d\sigma(\mathbf{v}) dt \\ &= K(\mathbf{x}, \mathbf{z}) \end{aligned} \tag{13}$$

□

Proposition 3.1: Let $\mathbf{U} = [\mathbf{V}, -\mathbf{V}]$, using point set $\mathbf{U}$ to approximate a real valued shift and rotation invariant kernel $K(\mathbf{x}, \mathbf{z})$ by using equation (10) is equal to using point set $\mathbf{V}$ to approximate $K(\mathbf{x}, \mathbf{z})$:

$$\Psi(\mathbf{x}; \mathbf{U})^T \Psi(\mathbf{z}; \mathbf{U}) = \Psi(\mathbf{x}; \mathbf{V})^T \Psi(\mathbf{z}; \mathbf{V}) \tag{14}$$

Proof: Note that cosine function is an even function. Thus, we obtain equation (15).

$$\cos\left(\Phi^{-}(t_j)(\mathbf{x} - \mathbf{z})^T \mathbf{v}_i\right) = \cos\left(-\Phi^{-}(t_j)(\mathbf{x} - \mathbf{z})^T \mathbf{v}_i\right) \tag{15}$$

Thus, we achieve equation (16).

$$\begin{aligned}
&\Psi(\mathbf{x}; \mathbf{U})^T \Psi(\mathbf{z}; \mathbf{U}) \\
&= \tfrac{1}{2NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \cos\left(\Phi^{-}(t_j)(\mathbf{x} - \mathbf{z})^T \mathbf{v}_i\right) \\
&+ \tfrac{1}{2NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \cos\left(-\Phi^{-}(t_j)(\mathbf{x} - \mathbf{z})^T \mathbf{v}_i\right) \\
&= \tfrac{1}{2NM} \sum_{i=1}^{N} \sum_{j=1}^{M} 2\cos\left(\Phi^{-}(t_j)(\mathbf{x} - \mathbf{z})^T \mathbf{v}_i\right) \\
&= \Psi(\mathbf{x}; \mathbf{V})^T \Psi(\mathbf{z}; \mathbf{V})
\end{aligned} \tag{16}$$

□

Proposition 3.1 shows that for a shift and rotation invariant kernel, computing N points can achieve the same approximation effect compared with using $2N$ points.

3.2. Feature Maps for $\mathbf{b^{th}}$-order Arc-cosine Kernels

In this subsection, we discuss the feature maps for b^{th}-order arc-cosine kernels. We discuss them separately because they are rotation invariant kernels but not shift invariant kernels in general. Moreover, they are closely related to deep neural networks (Cho & Saul, 2009), which demonstrate super performance in many areas.

Lemma 3.1: The b^{th}-order arc-cosine kernels can be calculated as equation (17).

$$\begin{aligned}
\mathrm{K}_b(\mathbf{x}, \mathbf{z}) = C_b \textstyle\int_{\mathbb{S}^{d-1}} &\chi\left(\mathbf{v}^T\mathbf{x}\right) \chi\left(\mathbf{v}^T\mathbf{z}\right) \\
&+ \chi(-\mathbf{v}^T\mathbf{x})\chi(-\mathbf{v}^T\mathbf{z}) d\sigma(\mathbf{v})
\end{aligned} \tag{17}$$

where $\chi(x) = \max(0, sign(x)|x|^b)$, $C_b = \int_{\mathbb{R}_+} r^{2b} p(r) dr$. C_b is a constant that is independent of $\mathbf{x}$ and $\mathbf{z}$. $p(r)$ is the density function of the chi-distribution with d degrees freedom. For example, the constants associated with the zero, first and second-order arc-cosine kernels are $C_0 = 1$, $C_1 = d$ and $C_2 = d(d+2)$ respectively.

Proof: From equation (5), we can achieve equation (18).

$$\begin{aligned}
\mathrm{K}_b(\mathbf{x}, \mathbf{z}) &= 2\textstyle\int_{\mathbb{R}^d} s(\mathbf{w}^T\mathbf{x}) s(\mathbf{w}^T\mathbf{z}) (\mathbf{w}^T\mathbf{x})^b (\mathbf{w}^T\mathbf{z})^b p(\mathbf{w}) d\mathbf{w} \\
&= 2\textstyle\int_{\mathbb{R}^d} \chi\left(\mathbf{w}^T\mathbf{x}\right) \chi\left(\mathbf{w}^T\mathbf{z}\right) p(\mathbf{w}) d\mathbf{w}
\end{aligned} \tag{18}$$

Let $r = \|\mathbf{w}\|_2$. Since p is standard Gaussian, by taking rotation invariant property, we obtain equation (19).

$$\begin{aligned}
\mathrm{K}_b(\mathbf{x}, \mathbf{z}) &= \mathbf{2}\textstyle\int_{\mathbb{R}^\mathbf{d}} \chi\left(\mathbf{w^T x}\right) \chi\left(\mathbf{w^T z}\right) \mathbf{p(w)dw} \\
&= 2\textstyle\int_{\mathbb{S}^{d-1}} \int_{\mathbb{R}_+} \chi\left(r^b \mathbf{v}^T\mathbf{x}\right) \chi\left(r^b \mathbf{v}^T\mathbf{z}\right) p(r) d\sigma(\mathbf{v}) dr \\
&= 2\textstyle\int_{\mathbb{S}^{d-1}} \int_{\mathbb{R}_+} r^{2b} \chi\left(\mathbf{v}^T\mathbf{x}\right) \chi\left(\mathbf{v}^T\mathbf{z}\right) p(r) d\sigma(\mathbf{v}) dr \\
&= 2\textstyle\int_{\mathbb{R}_+} r^{2b} p(r) dr \int_{\mathbb{S}^{d-1}} \chi\left(\mathbf{v}^T\mathbf{x}\right) \chi\left(\mathbf{v}^T\mathbf{z}\right) d\sigma(\mathbf{v}) \\
&= 2C_b \textstyle\int_{\mathbb{S}^{d-1}} \chi\left(\mathbf{v}^T\mathbf{x}\right) \chi\left(\mathbf{v}^T\mathbf{z}\right) d\sigma(\mathbf{v})
\end{aligned} \tag{19}$$

Since $\mathrm{K}_b(\mathbf{x}, \mathbf{z})$ is rotation invariant, we have $\mathrm{K}_b(\mathbf{x}, \mathbf{z}) = \mathrm{K}_b(-\mathbf{x}, -\mathbf{z})$. Together with equation (19), we achieve equation (20).

$$\begin{aligned}
\mathrm{K}_b(\mathbf{x}, \mathbf{z}) = C_b \textstyle\int_{\mathbb{S}^{d-1}} &\chi\left(\mathbf{v}^T\mathbf{x}\right) \chi\left(\mathbf{v}^T\mathbf{z}\right) \\
&+ \chi(-\mathbf{v}^T\mathbf{x})\chi(-\mathbf{v}^T\mathbf{z}) d\sigma(\mathbf{v})
\end{aligned} \tag{20}$$

□

The feature maps for a b^{th}-order arc-cosine kernel $\mathrm{K}_b(\mathbf{x}, \mathbf{z})$ can be constructed as equation (21).

$$\begin{aligned}
\Psi(\mathbf{x}) = \sqrt{\tfrac{C_b}{N}} [&\chi\left(\mathbf{v}_1^T\mathbf{x}\right), \chi\left(-\mathbf{v}_1^T\mathbf{x}\right), ..., \\
&\chi\left(\mathbf{v}_N^T\mathbf{x}\right), \chi\left(-\mathbf{v}_N^T\mathbf{x}\right)]^T \in \mathbb{R}^{2N}
\end{aligned} \tag{21}$$

Theorem 3.2: $\Psi(\mathbf{x})^T \Psi(\mathbf{z})$ is an unbiased estimate of a b^{th}-order arc-cosine kernel $\mathrm{K}_b(\mathbf{x}, \mathbf{z})$.

Proof: According to the Lemma 3.1 and the property of the asymptotically uniformly distributed point set $\mathbf{V}$, we obtain equation (22).

$$\begin{aligned}
&\lim_{N\to\infty} \Psi(\mathbf{x})^T \Psi(\mathbf{z}) \\
&= \lim_{N\to\infty} \tfrac{C_b}{N} \sum_{i=1}^{N} \chi\left(\mathbf{v}_i^T\mathbf{x}\right) \chi\left(\mathbf{v}_i^T\mathbf{z}\right) + \chi(-\mathbf{v}_i^T\mathbf{x})\chi(-\mathbf{v}_i^T\mathbf{z}) \\
&= C_b \textstyle\int_{S^{d-1}} \chi\left(\mathbf{v}^T\mathbf{x}\right) \chi\left(\mathbf{v}^T\mathbf{z}\right) + \chi(-\mathbf{v}^T\mathbf{x})\chi(-\mathbf{v}^T\mathbf{z}) d\sigma(\mathbf{v}) \\
&= \mathrm{K}_b(\mathbf{x}, \mathbf{z})
\end{aligned} \tag{22}$$

□

From equation (17) and (22), we observe that the approximation is actually operated on the $(d-1)$-dimensional domain instead of d-dimensional domain (Cho & Saul, 2009). Generally, the approximation error of Quasi Monte Carlo methods with N points depends on the dimension of integration. A lower dimension leads to smaller approximation error, thus the feature maps in equation (21) can achieve lower approximation error.

The feature maps in equation (21) are closely related to the bidirectional activation neural network. Specifically, the feature maps for the first-order arc-cosine kernel are related to the bidirectional ReLU activation function (An et al., 2015) which has the distance preservation property compared with ReLU.

From equation (14) and (21), we know that the feature maps actually rely on the point set $\mathbf{U} = [\mathbf{V}, -\mathbf{V}]$. The design of the point set $\mathbf{U}$ will be discussed in section 4.

4. Design of Matrix U

We have discussed the construction of SSF maps in last section. However, one unsolved problem is how to obtain the matrix $\mathbf{U} = [\mathbf{V}, -\mathbf{V}]$. We employ the discrete Riesz s-energy as the objective function to obtain matrix $\mathbf{U}$ because it can generate asymptotically uniformly distributed points on $\mathbb{S}^{d-1}$ (Brauchart & Grabner, 2015). Moreover, to achieve computation and storage efficiency for feature maps construction , we add a structured constraint to the matrix $\mathbf{U}$. In this section, we show the structure of matrix $\mathbf{U}$ first and then the optimization of discrete Riesz s-energy.

It is worth noting that matrix $\mathbf{U}$ can be used not only for kernel approximation, but also for approximation of general integrals over hypersphere. Moreover, by using FFT, matrix $\mathbf{U}$ can accelerate the integral approximation which involves projection operations. In addition, it only needs to store the indexes with linear storage cost (i.e. $O(d)$) instead of to explicitly store the matrix with cost $O(Nd)$.

4.1. Structure of Matrix U

Since $\mathbf{U}$ can be constructed by $\mathbf{V}$, i.e. $\mathbf{U} = [\mathbf{V}, -\mathbf{V}]$, we only need to define structured matrix $\mathbf{V}$. To achieve log-linear time complexity of SSF maps construction, we construct $\mathbf{V}$ by extracting rows from a discrete Fourier matrix. The complexity analysis of SSF maps construction based on matrix $\mathbf{V}$ is given in section 5.

Mathematically, the construction of matrix $\mathbf{V}$ is shown as follows. Without loss of generality, we assume that $d = 2m, N = 2n$, $m < n$. Let $F \in \mathbb{C}^{n \times n}$ be a $n \times n$ discrete Fourier matrix. $F_{k,j} = e^{\frac{2\pi ikj}{n}}$ is the $(k,j)^{th}$entry of F, where $i = \sqrt{-1}$. Let $\Lambda = [k_1, k_2, ..., k_m] \subset \{1, ..., n-1\}$ be a subset of indexes.

The structured matrix $\mathbf{V}$ can be defined as equation (23).

$$\mathbf{V} = \mathbf{\frac{1}{\sqrt{m}}} \begin{bmatrix} \mathrm{Re}F_\Lambda & -\mathrm{Im}F_\Lambda \\ \mathrm{Im}F_\Lambda & \mathrm{Re}F_\Lambda \end{bmatrix} \in \mathbb{R}^{\mathbf{d \times N}} \tag{23}$$

where F_Λ in equation (24) is the matrix constructed by m rows of F.

$$F_\Lambda = \begin{bmatrix} e^{\frac{2\pi ik_1 1}{n}} & \cdots & e^{\frac{2\pi ik_1 n}{n}} \\ \vdots & \ddots & \vdots \\ e^{\frac{2\pi ik_m 1}{n}} & \cdots & e^{\frac{2\pi ik_m n}{n}} \end{bmatrix} \in \mathbb{C}^{m \times n} \tag{24}$$

With the $\mathbf{V}$ given in equation (23), it is easy to verify that $\|\mathbf{v}_i\|_2 = 1$ for $i \in \{1, ..., n\}$. Thus, each column of matrix $\mathbf{V}$ is a point on $\mathbb{S}^{d-1}$.

4.2. Minimize the Discrete Riesz s-energy

With structured matrix $\mathbf{V}$ defined in equation (23), our goal is to select a subset of indexes Λ that optimizes the discrete Riesz s-energy. Specifically, we will discuss how to minimize the Riesz 0-energy in equation (25). The other Riesz s-energy can be optimized in a similar way.

$$E(\mathbf{U}) = \sum_{i=1}^{2N} \sum_{j=1, j\neq i}^{2N} \log \frac{1}{\|\mathbf{u}_i - \mathbf{u}_j\|} \tag{25}$$

where $\mathbf{U} = [\mathbf{V}, -\mathbf{V}] = [\mathbf{u}_1, ..., \mathbf{u}_{2N}]$.

In the following, we will discuss how to minimize equation (25) by using a coordinate decent method.

Theorem 4.1: Let $\mathbf{U} = [\mathbf{V}, -\mathbf{V}]$ with $\mathbf{V}$ defined in (23), the discrete Riesz 0-energy of $\mathbf{U}$ can be calculated as equation (26).

$$\begin{aligned} E(\mathbf{U}) = C - 2n \sum_{p=1}^{n-1} \log \left(1 - (\mathrm{Im} \tfrac{1}{m} \sum_{s=1}^{m} e^{\frac{2\pi ik_s p}{n}})^2 \right) \\ -2n \sum_{p=1}^{n-1} \log \left(1 - (\mathrm{Re} \tfrac{1}{m} \sum_{s=1}^{m} e^{\frac{2\pi ik_s p}{n}})^2 \right) \end{aligned} \tag{26}$$

where C is a constant independent of the choice of Λ.

Proof: Since $\mathbf{U} = [\mathbf{V}, -\mathbf{V}] \in \mathbb{S}^{\mathbf{(d-1) \times 2N}}$, we obtain equation (27).

$$\begin{aligned} & E(\mathbf{U}) = -\sum_{\mathbf{i=1}}^{\mathbf{2N}} \sum_{\mathbf{j=1, j\neq i}}^{\mathbf{2N}} \log \|\mathbf{u_i} - \mathbf{u_j}\| \\ & = -2 \sum_{i=1}^{N} \log \|2\mathbf{v}_i\| \\ & -2 \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} (\log \|\mathbf{v}_i - \mathbf{v}_j\| + \log \|\mathbf{v}_i + \mathbf{v}_j\|) \\ & = C - 2 \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \log (\|\mathbf{v}_i - \mathbf{v}_j\| \, \|\mathbf{v}_i + \mathbf{v}_j\|) \\ & = C - 2 \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \log \left(\sqrt{2 - 2\mathbf{v}_i^T \mathbf{v}_j} \sqrt{2 + 2\mathbf{v}_i^T \mathbf{v}_j} \right) \end{aligned} \tag{27}$$

Recall that $N = 2n$. By separating the summation term into two parts (each part has $n \times n$ term), we achieve equation (28).

$$\begin{aligned} & E(\mathbf{U}) = \mathbf{C} - \mathbf{2} \sum_{\mathbf{i=1}}^{\mathbf{2n}} \sum_{\mathbf{j=1, j\neq i}}^{\mathbf{2n}} \log \left(\mathbf{2} \sqrt{\mathbf{1} - (\mathbf{v_i^T v_j})^{\mathbf{2}}} \right) \\ & = C - 4 \sum_{i=1}^{n} \sum_{j=n+1}^{2n} \log \left(2 \sqrt{1 - (\mathbf{v}_i^T \mathbf{v}_j)^2} \right) \\ & -4 \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \log \left(2 \sqrt{1 - (\mathbf{v}_i^T \mathbf{v}_j)^2} \right) \end{aligned} \tag{28}$$

Let $\mathbf{V}_{\cdot, \mathbf{1:n}} = [\mathbf{v_1}, ..., \mathbf{v_n}]$ and $\mathbf{V}_{\cdot, \mathbf{n+1:2n}} = [\mathbf{v_{n+1}}, ..., \mathbf{v_{2n}}]$ be the matrix consisting of the first n and last n columns of $\mathbf{V}$ respectively. We can obtain equation (29).

$$\begin{aligned} \mathbf{V}_{\cdot, \mathbf{1:n}}^{\mathbf{T}} \mathbf{V}_{\cdot, \mathbf{n+1:2n}} = \mathbf{\tfrac{1}{m}} \mathrm{Re} \mathbf{F_\Lambda}^{\mathbf{T}} (-\mathrm{Im} \mathbf{F_\Lambda}) \\ + \tfrac{1}{m} (\mathrm{Im} F_\Lambda)^T \mathrm{Re} F_\Lambda \end{aligned} \tag{29}$$

Note that all diagonal elements of $\mathbf{V}^{\mathbf{T}}_{\cdot,\mathbf{1:n}}\mathbf{V}_{\cdot,\mathbf{n+1:2n}}$ are zero. By further separating the first summation term of equation (28) into two parts, we obtain equation (30).

$$
\begin{aligned}
E(\mathbf{U}) = \mathbf{C} - \mathbf{4}\sum_{\mathbf{i=1}}^{\mathbf{n}}\sum_{\mathbf{j=n+i}}^{\mathbf{2n}} \log\left(\mathbf{2}\sqrt{\mathbf{1-0}}\right) \\
-4\sum_{i=1}^{n}\sum_{j=n+1,j\neq n+i}^{2n} \log\left(2\sqrt{1-(\mathbf{v}_i^T\mathbf{v}_j)^2}\right) \\
-4\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} \log\left(2\sqrt{1-(\mathbf{v}_i^T\mathbf{v}_j)^2}\right) \\
= C - 4\sum_{i=1}^{n}\sum_{j=n+1,j\neq n+i}^{2n} \log\left(2\sqrt{1-(\mathbf{v}_i^T\mathbf{v}_j)^2}\right) \\
-4\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} \log\left(2\sqrt{1-(\mathbf{v}_i^T\mathbf{v}_j)^2}\right)
\end{aligned}
\tag{30}
$$

To be concise, let $\mathbf{Z} = [\mathbf{z_1},,...,\mathbf{z_n}] = \frac{\mathbf{1}}{\sqrt{\mathbf{m}}}\mathbf{F_\Lambda}$.

For $1 \leq j \leq n, j \neq i$, we achieve equation (31).

$$
(\mathbf{v_i^T v_j})^{\mathbf{2}} = (\mathrm{Re}\mathbf{z_i^* z_j})^{\mathbf{2}} = \left(\frac{1}{m}\mathrm{Re}\sum_{s=1}^{m} e^{2\pi i k_s p/n}\right)^2
\tag{31}
$$

For $n+1 \leq j \leq 2n, , j \neq n+i$, we attain equation (32).

$$
(\mathbf{v_i^T v_j})^{\mathbf{2}} = (\mathrm{Im}\mathbf{z_i^* z_{j-n}})^{\mathbf{2}} = \left(\frac{1}{m}\mathrm{Im}\sum_{s=1}^{m} e^{2\pi i k_s p/n}\right)^2
\tag{32}
$$

In equation (31) and (32), $p \equiv i-j \pmod{n}$, where mod denotes the modulus operation on integers.

Note that $\mathbf{z_i^* z_j}$ has at most $n-1$ distinct values when $i \neq j \pmod{n}$. Together with equation (30), we achieve equation (33).

$$
\begin{aligned}
E(\mathbf{U}) = \mathbf{C} - \mathbf{4}\sum_{\mathbf{i=1}}^{\mathbf{n}}\sum_{\mathbf{j=n+1,j\neq n+i}}^{\mathbf{2n}} \log\left(\mathbf{2}\sqrt{\mathbf{1}-(\mathbf{v_i^T v_j})^{\mathbf{2}}}\right) \\
-4\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} \log\left(2\sqrt{1-(\mathbf{v_i^T v_j})^2}\right) \\
= C - 4\sum_{i=1}^{n}\sum_{j=n+1,j\neq n+i}^{2n} \log\left(2\sqrt{1-(\mathrm{Im}\mathbf{z_i^* z_{j-n}})^2}\right) \\
-4\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} \log\left(2\sqrt{1-(\mathrm{Re}\mathbf{z_i^* z_j})^2}\right) \\
= C - 4n\sum_{p=1}^{n-1} \log\left(2\sqrt{1-(\mathrm{Im}\frac{1}{m}\sum_{s=1}^{m} e^{2\pi i k_s p/n})^2}\right) \\
-4n\sum_{p=1}^{n-1} \log\left(2\sqrt{1-(\mathrm{Re}\frac{1}{m}\sum_{s=1}^{m} e^{2\pi i k_s p/n})^2}\right) \\
= C - 2n\sum_{p=1}^{n-1} \log\left(1-(\mathrm{Im}\frac{1}{m}\sum_{s=1}^{m} e^{2\pi i k_s p/n})^2\right) \\
-2n\sum_{p=1}^{n-1} \log\left(1-(\mathrm{Re}\frac{1}{m}\sum_{s=1}^{m} e^{2\pi i k_s p/n})^2\right)
\end{aligned}
\tag{33}
$$

□

Algorithm 1

```
**Initialization:** random sample $\Lambda = [k_1, k_2, ..., k_m]$ from $\{1, 2, ...n-1\}$ without replacement. Set $\widetilde{\mathbf{h}} = \mathbf{1}^T F_\Lambda$
repeat
  Set $J = J(\Lambda)$
  for $q = 1$ to $m$ do
    Set $\mathbf{g} = [e^{2\pi i k_q/n}, e^{2\pi i k_q 2/n}..., e^{2\pi i k_q(n-1)/n}]$
    Set $\mathbf{h} = \widetilde{\mathbf{h}} - \mathbf{g}$
    Find $k_q^*$ by $k_q^* = \underset{k_q\in\{1,...,n-1\}}{\arg\max} J(k_q)$ in (35)
    Update $\mathbf{g} = [e^{2\pi i k_q^*/n}, e^{2\pi i k_q^* 2/n}..., e^{2\pi i k_q^*(n-1)/n}]$
    Set $\widetilde{\mathbf{h}} = \mathbf{h} + \mathbf{g}$
  end for
until $J$ does not change
```

From Theorem 4.1, we know that minimizing $E(\mathbf{U})$ is equivalent to maximizing $J(\Lambda)$ which is defined in equation (34).

$$
\begin{aligned}
J(\Lambda) = \sum_{p=1}^{n-1} \log\left(1-(\mathrm{Im}\frac{1}{m}\sum_{s=1}^{m} e^{2\pi i k_s p/n})^2\right) \\
+ \sum_{p=1}^{n-1} \log\left(1-(\mathrm{Re}\frac{1}{m}\sum_{s=1}^{m} e^{2\pi i k_s p/n})^2\right)
\end{aligned}
\tag{34}
$$

By keeping all the indexes in $\Lambda = [k_1, k_2, ..., k_m]$ fixed except the q^{th} element, we can obtain equation (35).

$$
\begin{aligned}
J(k_q) = \sum_{p=1}^{n-1} \log\left(1-\left(\mathrm{Im}\left(h_p + e^{2\pi i k_q p/n}\right)/m\right)^2\right) \\
+ \sum_{p=1}^{n-1} \log\left(1-\left(\mathrm{Re}\left(h_p + e^{2\pi i k_q p/n}\right)/m\right)^2\right)
\end{aligned}
\tag{35}
$$

where $k_q \in \{1, 2, ...n-1\}$, $h_p = \sum_{s=1,s\neq q}^{m} e^{2\pi i k_s p/n}$.

With equation (35), we can maximize $J(\Lambda)$ by maximizing $J(k_q)$ with other indexes fixed each time. Let $\mathbf{h} = [h_1, ..., h_{n-1}]$, $\mathbf{g} = [e^{2\pi i k_q/n}, e^{2\pi i k_q 2/n}..., e^{2\pi i k_q(n-1)/n}]$. $\mathbf{1} = [1, ..., 1]^T \in \mathbb{R}^m$ is the vector of all ones. A coordinate ascent method to maximize $J(\Lambda)$ is given in Algorithm 1.

Obviously, it is a discrete optimization problem. Algorithm 1 can find a local optimum. The time complexity of the Algorithm 1 is $O(Tmn^2)$, where T denotes the number of outer iteration. Empirically, the outer iteration T is less than ten.

5. Fast Feature Maps Construction

In this section, we will discuss how to construct SSF maps in loglinear time complexity and linear space complexity by using the structure property of $\mathbf{V}$.

Theorem 5.1 Assume that $d = 2m, N = 2n, m < n$. Let

$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \in \mathbb{R}^{\mathbf{2m}}$ and $\mathbf{z} = \mathbf{x_1} + \mathbf{i}\mathbf{x_2} \in \mathbb{C}^{\mathbf{m}}$. Given $\Lambda = [k_1, k_2, ..., k_m] \subset \{1, ..., n-1\}$, let $\mathbf{y} \in \mathbb{C}^{\mathbf{n}}$ with $\mathbf{y_\Lambda} = \mathbf{z}$. Other elements outside the index set Λ are equal to zero. Given $\mathbf{V}$ defined in equation (23), equation (36) holds.

$$\mathbf{V}^T\mathbf{x} = \tfrac{1}{\sqrt{m}}[\mathrm{Re}(F^*\mathbf{y}), \mathrm{Im}(F^*\mathbf{y})]^T \tag{36}$$

Proof: Let $\Omega \in \mathbb{R}^{n\times n}$ be a diagonal matrix with all diagonal elements inside the index set Λ equal to one , the others equal to zero.

$$\begin{aligned}\mathbf{V}^T\mathbf{x} &= \tfrac{1}{\sqrt{m}}\begin{bmatrix} \mathrm{Re}F_\Lambda & -\mathrm{Im}F_\Lambda \\ \mathrm{Im}F_\Lambda & \mathrm{Re}F_\Lambda \end{bmatrix}^T \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \\ &= \tfrac{1}{\sqrt{m}}\begin{bmatrix} (\mathrm{Re}{F_\Lambda}^T)\mathbf{x}_1 + (\mathrm{Im}{F_\Lambda}^T)\mathbf{x}_2 \\ (-\mathrm{Im}{F_\Lambda}^T)\mathbf{x}_1 + (\mathrm{Re}{F_\Lambda}^T)\mathbf{x}_2 \end{bmatrix} \\ &= \tfrac{1}{\sqrt{m}}\begin{bmatrix} \mathrm{Re}(F_\Lambda^*\mathbf{z}) \\ \mathrm{Im}(F_\Lambda^*\mathbf{z}) \end{bmatrix} \\ &= \tfrac{1}{\sqrt{m}}\begin{bmatrix} \mathrm{Re}(F^*\Omega\mathbf{y}) \\ \mathrm{Im}(F^*\Omega\mathbf{y}) \end{bmatrix} \\ &= \tfrac{1}{\sqrt{m}}\begin{bmatrix} \mathrm{Re}(F^*\mathbf{y}) \\ \mathrm{Im}(F^*\mathbf{y}) \end{bmatrix}\end{aligned} \tag{37}$$

□

Thus, the projection operation $\mathbf{V}^T\mathbf{x}$ (previously mentioned in equation (10) and (21)) can be calculated by Fast Fourier Transform algorithm (FFT) in $O(n \log n)$ time complexity. Because scaling and taking nonlinear transform can be finished in $O(n)$, the total time complexity to construct SSF maps is $O(n \log n)$.

All steps to construct SSF maps are summarized as follows:

(a) Compute $\widetilde{\mathbf{x}}$ by $\widetilde{\mathbf{x}} = \mathbf{D}\mathbf{x}$, where $\mathbf{D} \in \{\mathbf{-1}, \mathbf{+1}\}^{\mathbf{d}\times\mathbf{d}}$ is a diagonal matrix where diagonal elements are uniformly sampled from $\{-1, +1\}$.

(b) Construct $\mathbf{y}$ such that $\mathbf{y_\Lambda} = \widetilde{\mathbf{x}}_\mathbf{1} + \mathbf{i}\widetilde{\mathbf{x}}_\mathbf{2}$, other elements outside the index set Λ are equal to zero.

(c) Compute $\mathbf{V}^T\widetilde{\mathbf{x}}$ by equation (36) via FFT.

(d) Construct feature maps $\Psi\,(\mathbf{x})$ via equation (10) or (21).

For each (m, n) pair , the index set Λ only need to be computed once. It takes $O(m)$ space to store Λ. For shift and rotation invariant kernels, it takes $O(M)$ space to store $\Phi^{-}(t_j), j \in 1, ..., M$ and takes $O(d)$ $(d = 2m)$ space to store Λ and $\mathbf{D}$. For b^{th}-order arc-cosine kernels, it only needs to store one parameter C_b and takes $O(d)$ space to store Λ and $\mathbf{D}$. By setting $M \leq d$, the total space complexity to store the projection matrix is $O(d)$.

6. Empirical Studies

We compare SSF maps with feature maps obtained by fully Gaussian (Cho & Saul, 2009; Rahimi et al., 2007), the Circulant (Choromanski & Sindhwani, 2016) matrices, QMC with Halton set and QMC with Sobol set (Avron et al., 2016). For Halton set and Sobol set, the implementation in MATLAB are employed in the experiments. The scrambling and shifting techniques are used for Haltonset and Sobolset. In all the experiments, we fix $M = 1$ (the number of one-dimensional QMC points) for SSF maps.

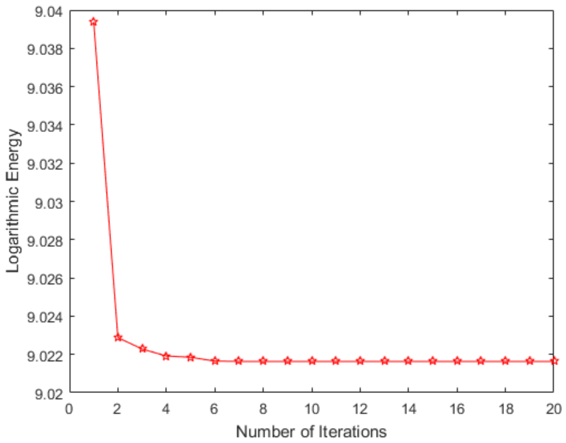


Figure 1. Convergence of the Logarithmic Energy

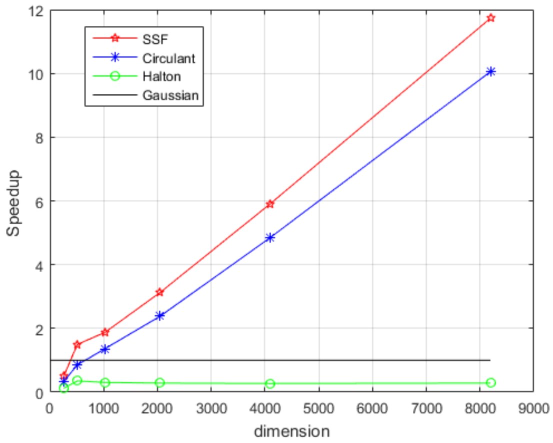
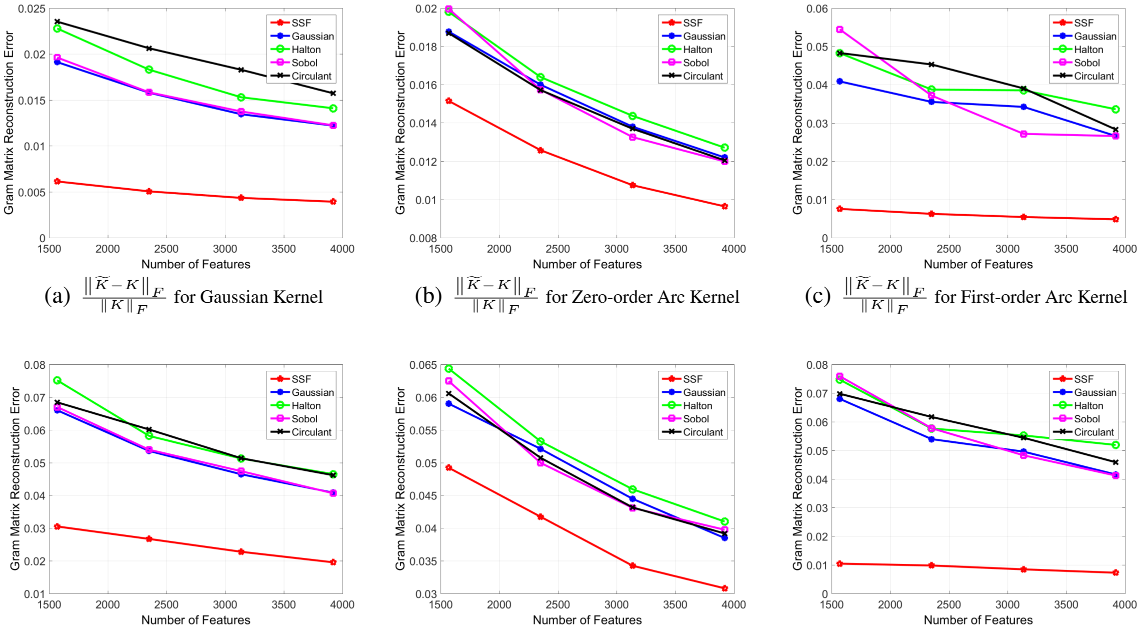


Figure 2. Speedup of the Feature Maps Construction

6.1. Convergence and Speedup

First, the convergence of the logarithmic energy ($-J(\Lambda)$ in equation (34)) with $(m, n) = (160, 1600)$ is shown in Figure 1. From Figure 1, we find that it takes less than ten iterations (i.e. $T < 10$) for Algorithm 1 to find a local optimum.

Second, the speedup results of all methods are shown in



(a) $\frac{\|\widetilde{K}-K\|_F}{\|K\|_F}$ for Gaussian Kernel

(b) $\frac{\|\widetilde{K}-K\|_F}{\|K\|_F}$ for Zero-order Arc Kernel

(c) $\frac{\|\widetilde{K}-K\|_F}{\|K\|_F}$ for First-order Arc Kernel

(d) $\frac{\|\widetilde{K}-K\|_\infty}{\|K\|_\infty}$ for Gaussian Kernel

(e) $\frac{\|\widetilde{K}-K\|_\infty}{\|K\|_\infty}$ for Zero-order Arc Kernel

(f) $\frac{\|\widetilde{K}-K\|_\infty}{\|K\|_\infty}$ for First-order Arc Kernel

Figure 3. Relative Mean and Max Reconstruction Error for Gaussian, Zero-order and First-order Arc-cosine Kernel on MNIST

Figure 2. We set $N = 2d$ for all the methods. The speedup of fully Gaussian projection is the baseline. We can observe that the speedup of QMC with Halton set is constant as the dimension d increases and is slower than the baseline. The speedup of both SSF maps and the Circulant increase fast as dimension increases, which is consistent with theoretical analysis. The speedup of Sobol set is not shown because the inbuilt Sobolset routine of MATLAB does not support dimension larger than 1,111.

6.2. Approximation Accuracy

We evaluate reconstruction error of Gaussian kernel, zero-order arc-cosine kernel and first-order arc-cosine kernel on CIFAR10 (Krizhevsky & Hinton, 2009), MNIST (LeCun & Cortes, 2010), usps and dna dataset. MNIST is a handwritten digit image dataset, which contains 70,000 samples with 784-dimensional features(pixel). For CIFAR10 with 60,000 samples, the 320-dimensional gist feature (Gong et al., 2013) are employed in the experiments. Both the relative Frobenius error (i.e. $\frac{\|\widetilde{K}-K\|_F}{\|K\|_F}$) and the relative element-wise maximum error (i.e. $\frac{\|\widetilde{K}-K\|_\infty}{\|K\|_\infty}$) are evaluated, where K and $\widetilde{K}$ denote the exact and approximated Gram matrices respectively. The Frobenius norm and the elementwise maximum norm are defined as $\|X\|_F = \sqrt{\sum_i \sum_j |X_{ij}|^2}$ and $\|X\|_\infty = \max_{i,j} |X_{ij}|$ respectively.

The reconstruction error in the experiments is the mean value over 10 independent runs. The dimensions of the feature maps are set to $\{2\times d, 3\times d, 4\times d, 5\times d\}$, where d is the dimension of the data. For MNIST and CIFAR10 dataset, each run randomly select 2,000 samples to construct the Gram matrix. The mean value of the reconstruction errors with different norms on MNIST are shown in Figure 3. Results on the other datasets are similar to that of Figure 3. One can refer to the supplementary material for results on other datasets.

Figure 3 shows that the feature maps obtained with fully Gaussian matrix, the Circulant matrix, QMC with Halton set and QMC with Sobol set have similar reconstruction error. SSF maps have the smallest approximation error among five methods. Especially for the first-order arc-cosine kernel, it achieves nearly one-fifth relative mean error and one-seventh relative max error of other methods. Moreover, even if $M = 1$, SSF maps can achieve about one-third relative mean error and half of the relative max error of other methods for Gaussian Kernel approximation.

7. Conclusion

We propose Spherical Structured Feature (SSF) maps to approximate shift and rotation invariant kernels as well as b^{th}-order arc-cosine kernels. SSF maps can achieve computation and storage efficiency as well as better approximation accuracy.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions.

References

An, Senjian, Boussaid, Farid, and Bennamoun, Mohammed. How can deep rectifier networks achieve linear separability and preserve distances? In *ICML*, pp. 514–523, 2015.

Avron, Haim, Sindhwani, Vikas, Yang, Jiyan, and Mahoney, Michael W. Quasi-monte carlo feature maps for shift-invariant kernels. *Journal of Machine Learning Research*, 17(120):1–38, 2016.

Brauchart, J, Saff, E, Sloan, I, and Womersley, R. Qmc designs: optimal order quasi monte carlo integration schemes on the sphere. *Mathematics of computation*, 83 (290):2821–2851, 2014.

Brauchart, Johann S and Grabner, Peter J. Distributing many points on spheres: minimal energy and designs. *Journal of Complexity*, 31(3):293–326, 2015.

Chang, Chih-Chung and Lin, Chih-Jen. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Cho, Youngmin and Saul, Lawrence K. Kernel methods for deep learning. In *Advances in neural information processing systems*, pp. 342–350, 2009.

Choromanski, Krzysztof and Sindhwani, Vikas. Recycling randomness with structure for sublinear time kernel expansions. 2016.

Cutajar, Kurt, Bonilla, Edwin V, Michiardi, Pietro, and Filippone, Maurizio. Practical learning of deep gaussian processes via random fourier features. *arXiv preprint arXiv:1610.04386*, 2016.

Dick, Josef, Kuo, Frances Y, and Sloan, Ian H. High-dimensional integration: the quasi-monte carlo way. *Acta Numerica*, 22:133–288, 2013.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

Feng, Chang, Hu, Qinghua, and Liao, Shizhong. Random feature mapping with signed circulant matrix projection. In *IJCAI*, pp. 3490–3496, 2015.

Gong, Yunchao, Lazebnik, Svetlana, Gordo, Albert, and Perronnin, Florent. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2013.

Götz, Mario. On the riesz energy of measures. *Journal of Approximation Theory*, 122(1):62–78, 2003.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

Le, Quoc, Sarlós, Tamás, and Smola, Alex. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, 2013.

Le, Zichao Yang Alexander J Smola and Wilson, Song Andrew Gordon. A la cartelearning fast kernels. 38, 2015.

LeCun, Yann and Cortes, Corinna. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Oliva, Junier B, Dubey, Avinava, Poczos, Barnabas, Schneider, Jeff, and Xing, Eric P. Bayesian nonparametric kernel-learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 1078–1086, 2016.

Rahimi, Ali and Recht, Benjamin. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pp. 1313–1320, 2009.

Rahimi, Ali, Recht, Benjamin, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, pp. 5, 2007.

Rasmussen, Carl Edward. Gaussian processes for machine learning. 2006.

Rudin, Walter. *Fourier analysis on groups*. John Wiley & Sons, 2011.

Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959, 2012.

Srinivas, Niranjan, Krause, Andreas, Kakade, Sham M, and Seeger, Matthias. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Sutherland, Dougal J and Schneider, Jeff. On the error of random fourier features. *arXiv preprint arXiv:1506.02785*, 2015.

Yang, J., Sindhwani. V. Avron H. Mahoney M. Quasi-monte carlo feature maps for shift-invariant kernels. *ICML*, 32, 2014.