Random Feature Mapping with Signed Circulant Matrix Projection

Chang Feng, Qinghua Hu, Shizhong Liao[*]
School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
{changfeng,huqinghua,szliao}@tju.edu.cn

Abstract

Random feature mappings have been successfully used for approximating non-linear kernels to scale up kernel methods. Some work aims at speeding up the feature mappings, but brings increasing variance of the approximation. In this paper, we propose a novel random feature mapping method that uses a signed Circulant Random Matrix (CRM) instead of an unstructured random matrix to project input data. The signed CRM has linear space complexity as the whole signed CRM can be recovered from one column of the CRM, and ensures loglinear time complexity to compute the feature mapping using the Fast Fourier Transform (FFT). Theoretically, we prove that approximating Gaussian kernel using our mapping method is unbiased and does not increase the variance. Experimentally, we demonstrate that our proposed mapping method is time and space efficient while retaining similar accuracies with state-of-the-art random feature mapping methods. Our proposed random feature mapping method can be implemented easily and make kernel methods scalable and practical for large scale training and predicting problems.

1 Introduction

Support Vector Machine (SVM) is one of the most popular classification tools, which is based on statistical learning theory and delivers excellent results for non-linear classification in machine learning [Vapnik, 1998; Schölkopf and Smola, 2002]. The underlying training problem can be formulated as a Quadratic Programming (QP) problem that can be solved by standard optimization algorithms in $O(l^3)$ time and $O(l^2)$ space complexity, where l is the number of training data [Tsang *et al.*, 2005]. This is computationally infeasible for training on very large datasets.

There is a lot of work proposed for scaling up non-linear kernel SVMs on large scale datasets, such as solving the QP problem exactly through decomposition methods [Platt, 1999; Chang and Lin, 2011], approximation algorithms using core vector set [Tsang *et al.*, 2005], and parallel interior point solver [Chang *et al.*, 2007]. However, the increasing volumes of datasets would lead to the *curse of support* [Kar and Karnick, 2012], where the number of Support Vectors (SVs) that have to be explicitly maintained grows linearly with the sample size on noisy data [Steinwart, 2003]. According to Representer Theorem [Kimeldorf and Wahba, 1970], and Karush-Kuhn-Tucker conditions [Boyd and Vandenberghe, 2004], one can typically represent the decision function $f(\boldsymbol{x})$ via the kernel trick with SVs, $f(\boldsymbol{x}) = \sum_{\boldsymbol{x}_i \in \mathrm{SVs}} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})$, $\alpha_i > 0$. Obviously, it requires huge computational time and space to train a model as SVs increase on large scale datasets. Moreover, to evaluate a learned model with hundreds of thousands of SVs will bring in additional time and space burden in the predicting stage.

Random feature mapping methods, such as Random Kitchen Sinks (RKS) [Rahimi and Recht, 2007; 2008] and Random Maclaurin Feature Maps [Kar and Karnick, 2012], are proposed for addressing the curse of support, which embed the implicit kernel feature space into a relatively low-dimensional explicit Euclidean space. In the embedded feature space, the kernel value of any two points is well approximated by their inner product and one can apply existing fast linear algorithms, such as linear SVMs that run in time linear with sample size [Joachims, 2006; Fan *et al.*, 2008], to abstract data relations corresponding to non-linear kernel methods. After learning a hyperplane with a linear classifier, one can predict an input in $O(dD)$ time complexity (independent on the number of training data and mainly used for computing random feature mapping), where d represents the dimensionality of the input and D the dimensionality of the random feature space. Therefore, random feature mapping methods with liner classifiers can inherit high efficiency of linear learning algorithms and good generalization performance of non-linear kernel methods, providing a promising way to the curse of support. However, the random feature mapping itself is a bottleneck when dD is not small.

Fastfood uses an approximation of the unstructured Gaussian matrix of RKS with several special matrices to accelerate the computation of random feature mapping [Le *et al.*, 2013]. Because the special matrices are easy to store and multiply, Fastfood computes the random feature mapping in $O(D \log d)$ time and $O(D)$ space complexity, a significant improvement from $O(dD)$ computation and storage. However, Fastfood brings great increasing variance of approxi-

[*]Corresponding author

mating the kernel, which will cause inaccurate approximation and loose concentration bound.

In this paper, we propose a novel random feature mapping method, called Signed Circulant Random Feature mapping (SCRF), for approximating non-linear kernel. Our proposed SCRF uses the structured random matrix $\mathbf{\Pi} \in \mathbb{R}^{D \times d}$, a stacking of D/d signed circulant Gaussian matrices, to project input data, followed by a non-linear transform, i.e., $\phi(\mathbf{\Pi}\boldsymbol{x})$. We prove that the approximation using SCRF is an unbiased estimate of the corresponding non-linear kernel and does not increase the variance of that using RKS. Therefore, the approximation using SCRF concentrates with the same rate as RKS and faster than that using Fastfood. Because $\mathbf{\Pi}$ has the circulant structure, instead of saving it directly, we can store the first columns of the corresponding circulant Gaussian matrices with a random sequence in $O(D)$ space complexity and then easily recover it. For the structured matrix projection, we can apply the Fast Fourier Transform (FFT) to compute the feature mapping in $O(D \log d)$ time complexity. Experimental results demonstrate that our proposed SCRF is much more time and space efficient than both RKS and Fastfood while retaining similar accuracies with state-of-the-art random feature mapping methods.

1.1 Related Work

Random feature mappings have been successfully used for approximating non-linear kernels to address the curse of support. As the first proposed random feature mapping method, Random Kitchen Sinks (RKS) [Rahimi and Recht, 2007] focuses on approximating translation invariant kernels (e.g., Gaussian kernel, Laplacian kernel). There have been several approaches proposed to approximate other kernels as well, including intersection [Maji and Berg, 2009], group invariant [Li *et al.*, 2010], additive [Vedaldi and Zisserman, 2012] and dot product kernels [Kar and Karnick, 2012]. Moreover, there has been work aiming at improving quality of the existing random feature mapping methods, some of which include a more compact representation of accurately approximating polynomial kernels [Hamid *et al.*, 2014] and a more effective Quasi-Monte Carlo feature mapping for translation invariant kernels [Yang *et al.*, 2014]. Yen *et al.* [Yen *et al.*, 2014] propose a sparse random feature algorithm so that the resulting model doesn't grow linearly with the number of random features.

Recently, two promising quasilinear kernel approximation techniques have been proposed to accelerate the existing random feature mappings. Tensor sketching applies recent results in tensor sketch convolution to deliver approximations for polynomial kernels in $O(d + D \log D)$ time [Pham and Pagh, 2013]. Fastfood uses an approximation of the unstructured Gaussian matrix of RKS with several special matrices to project input data [Le *et al.*, 2013]. Because the special matrices are inexpensive to multiply and store, Fastfood computes the random feature mapping in $O(D \log d)$ time and $O(D)$ space complexity.

We summarize the contributions of this paper as follows:

- We propose a novel scheme for random feature mapping that uses structured random matrix (signed circulant random matrix) instead of unstructured random matrix to project data.
- We prove that the approximating Gaussian kernel using our proposed SCRF is unbiased and does not increase the variance of that using RKS.
- We save the random parameters in $O(D)$ space complexity and implement our random feature mapping in $O(D \log d)$ time complexity by using FFT.
- Empirically, our random feature mapping method is time and space efficient, making kernel methods scalable for large scale training and predicting problems.

2 Preliminaries

In this section, we first review two well-known random feature mapping methods, Random Kitchen Sinks (RKS) [Rahimi and Recht, 2007] and Fastfood [Le *et al.*, 2013], for kernel approximation, and then introduce a structured matrix, circulant matrix [Davis, 1979; Gray, 2006].

2.1 Random Feature Mapping

The starting point of RKS is a celebrated result that characterizes the class of positive definite functions.

Theorem 1 (Bochner's theorem [Rudin, 2011]). *For any normalized continuous positive definite function $f : \mathbb{R}^d \to \mathbb{C}$, there exists a finite non-negative Borel measure μ on $\mathbb{R}^d$ such that*

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^d} \mathrm{e}^{-\mathrm{i}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}} \mathrm{d}\mu(\boldsymbol{w}), \tag{1}$$

i.e. f is the Fourier transform of a finite non-negative Borel measure μ on $\mathbb{R}^d$.

Without loss of generality, we assume henceforth that the kernel $\kappa(\delta)$ is properly scaled and $\mu(\cdot)$ is a probability measure with associated probability density function $p(\cdot)$.

Corollary 1. *For shift-invariant kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \kappa(\boldsymbol{x} - \boldsymbol{y})$,*

$$k(\boldsymbol{x}, \boldsymbol{y}) = \int_{\mathbb{R}^d} p(\boldsymbol{w}) \mathrm{e}^{-\mathrm{i}\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y})} \mathrm{d}\boldsymbol{w}, \tag{2}$$

where $p(\boldsymbol{w})$ is a probability density function and can be calculated through the inverse Fourier transform of κ.

For Gaussian kernel, $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2/(2\sigma^2))$, we calculate $p(\boldsymbol{w})$ through the inverse Fourier transform of $k(\boldsymbol{x}, \boldsymbol{y})$ and obtain $\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{I}/\sigma^2)$, where $\mathbf{I}$ is an identity matrix. In addition, we have $\mathbb{E}_{\boldsymbol{w}}[\sin(\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{y}))] = 0$ and $\mathbb{E}_{\boldsymbol{w},b}[\cos(\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x} + \boldsymbol{y}) + 2b)] = 0$, where b is drawn from $[-\pi, \pi]$ uniformly. Note that

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}) &= \mathbb{E}_{\boldsymbol{w}}[\mathrm{e}^{-\mathrm{i}\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y})}] \\ &= \mathbb{E}_{\boldsymbol{w}}[\cos(\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{y}))] \\ &= \mathbb{E}_{\boldsymbol{w},b}[\sqrt{2}\cos(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)\sqrt{2}\cos(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y} + b)]. \end{aligned}$$

Defining $Z_{\boldsymbol{w},b}(\boldsymbol{x}) = \sqrt{2}\cos(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)$, we get

$$k(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}[\langle Z_{\boldsymbol{w},b}(\boldsymbol{x}), Z_{\boldsymbol{w},b}(\boldsymbol{y})\rangle], \tag{3}$$

so $\langle Z_{\boldsymbol{w},b}(\boldsymbol{x}), Z_{\boldsymbol{w},b}(\boldsymbol{y})\rangle$ is an unbiased estimate of the Gaussian kernel. Through a standard Monte Carlo (MC) approximation to the integral representation of the kernel, we can lower the variance of $\langle Z_{\boldsymbol{w},b}(\boldsymbol{x}), Z_{\boldsymbol{w},b}(\boldsymbol{y})\rangle$ by concatenating D randomly chosen $Z_{\boldsymbol{w},b}$ into a column feature mapping vector and normalizing each component by $\sqrt{D}$,

$$\boldsymbol{\Phi}_{\mathrm{RKS}} : \boldsymbol{x} \mapsto \frac{\sqrt{2}}{\sqrt{D}} \cos(\boldsymbol{Z}\boldsymbol{x} + \boldsymbol{b}), \tag{4}$$

where $\boldsymbol{Z} \in \mathbb{R}^{D \times d}$ is a Gaussian matrix with each entry drawn i.i.d. from $\mathcal{N}(0, 1/\sigma^2)$, $\boldsymbol{b} \in \mathbb{R}^D$ is a random vector drawn i.i.d. from $[-\pi, \pi]$ uniformly and $\cos(\cdot)$ is an element-wise function.

As derived above, the associated feature mapping converges in expectation to the Gaussian kernel. In fact, convergence occurs with high probability and at the rate of independent empirical averages. In the explicit random feature space, we can use primal space methods for training, which delivers a potential solution to the curse of support. However, such approach is still limited by the fact that we need to store the unstructured Gaussian matrix $\boldsymbol{Z}$ and, more importantly, we need to compute $\boldsymbol{Z}\boldsymbol{x}$ for each $\boldsymbol{x}$. Because the unstructured matrix has the disadvantage that no fast matrix multiplication is available, large scale problems are not practicable with such unstructured matrix.

Fastfood finds that Hadamard matrix $\boldsymbol{H}$ when combined with binary scaling matrix $\boldsymbol{B}$, permutation matrix $\boldsymbol{Q}$, diagonal Gaussian matrix $\boldsymbol{G}$ and scaling matrix $\boldsymbol{S}$ exhibits properties similar to the unstructured Gaussian matrix $\boldsymbol{Z}$, i.e., $\boldsymbol{V} \approx \boldsymbol{Z}$, where

$$\boldsymbol{V} = \frac{1}{\sigma\sqrt{d}} \boldsymbol{SHGQHB}.$$

The decomposed matrices are inexpensive to multiply and store, which speeds up the random feature mapping. However, approximating Gaussian kernel using Fastfood brings increasing variance, which causes inaccurate approximation and loose concentration bound. Different from Fastfood, in this paper, we make use of a structured random matrix to take the place of $\boldsymbol{Z}$ and propose a novel scheme for random feature mapping.

2.2 Circulant Matrix

A circulant matrix $\boldsymbol{C}$ is an $m \times m$ Toeplitz matrix with the form

$$\boldsymbol{C} = \begin{bmatrix} c_0 & c_{m-1} & c_{m-2} & & \cdots & c_1 \\ c_1 & c_0 & c_{m-1} & c_{m-2} & & \vdots \\ & c_1 & c_0 & c_{m-1} & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & c_{m-2} \\ & & & \ddots & \ddots & c_{m-1} \\ c_{m-1} & c_{m-2} & \cdots & & c_1 & c_0 \end{bmatrix}, \tag{5}$$

where each column is a cyclic shift of its left one [Davis, 1979]. The structure can also be characterized as follows: For the (k, j) entry of $\boldsymbol{C}$, $\boldsymbol{C}_{kj}$,

$$\boldsymbol{C}_{kj} = c_{(k-j) \mod m}. \tag{6}$$

A circulant matrix is fully determined by its first column, so we rewrite the circulant matrix of order m as follows

$$\boldsymbol{C}_{[m]} = \mathrm{circ}\left[c_j : j \in \{0, 1, \ldots, m-1\}\right]. \tag{7}$$

Therefore, it only needs to store the first column vector so that we can reconstruct the whole matrix, which saves a lot of storage.

Definition 1 (Circulant Random Matrix, CRM). *A circulant matrix $\boldsymbol{C}_{[m]} = \mathrm{circ}\left[c_j : j \in \{0, 1, \ldots, m-1\}\right]$ is called a circulant random matrix if its first column is a random sequence with each entry drawn i.i.d according to some distribution probability.*

Definition 2 (Signed Circulant Random Matrix, Signed CRM). *If a matrix $\boldsymbol{P} = [\sigma_0 \boldsymbol{C}_{0\cdot}; \sigma_1 \boldsymbol{C}_{1\cdot}; \ldots; \sigma_{m-1} \boldsymbol{C}_{(m-1)\cdot}]$ satisfies that $\boldsymbol{C}_{i\cdot}$ is the i-th row vector of a circulant random matrix $\boldsymbol{C}_{[m]}$ and σ_i is a Rademacher variable ($\mathbb{P}[\sigma_i = 1] = \mathbb{P}[\sigma_i = -1] = 1/2$), where $i = 0, 1, \ldots, m-1$, we call $\boldsymbol{P}$ a signed circulant random matrix.*

The following lemma provides an equivalent form of a circulant matrix with the Discrete Fourier Transform (DFT) [Davis, 1979; Tyrtyshnikov, 1996].

Lemma 1. *Suppose $\boldsymbol{C}$ is a matrix with the first column $\boldsymbol{c} = [c_0, c_1, \ldots, c_{m-1}]^{\mathrm{T}}$. Then $\boldsymbol{C}$ is a circulant matrix, i.e.*

$$\boldsymbol{C} = \mathrm{circ}(\boldsymbol{c}),$$

if and only if

$$\boldsymbol{C} = \frac{1}{m} \boldsymbol{F}^* \mathrm{diag}(\boldsymbol{F}\boldsymbol{c}) \boldsymbol{F}, \tag{8}$$

where

$$\boldsymbol{F} = \left[\mathrm{e}^{\mathrm{i}\frac{2\pi}{m}kn}\right]_{k,n=0}^{m-1}$$

is the discrete Fourier matrix of order m ($\mathrm{i} = \sqrt{-1}$), $\mathrm{diag}(\boldsymbol{F}\boldsymbol{c})$ signifies the diagonal matrix whose diagonal comprises the entries of the vector $\boldsymbol{F}\boldsymbol{c}$, and $\boldsymbol{F}^*$ represents the conjugate transpose of $\boldsymbol{F}$.

Obviously, we could realize the calculation of $\boldsymbol{C}\boldsymbol{x}$ efficiently via the Fast Fourier Transform algorithm (FFT). In the following, we introduce our structured feature mapping method and apply such computational advantage to accelerate the feature mapping.

3 Signed Circulant Random Feature Mapping

In this section, we propose a random feature mapping method with signed circulant matrix projection, called Signed Circulant Random Feature mapping (SCRF), and analyse its approximation error.

3.1 Our Random Feature Mapping

Without loss of generality, we assume that $d \mid D$ (D is divisible by d). And then we generate $t = D/d$ signed circulant Gaussian matrices $\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \ldots, \boldsymbol{P}^{(t)}$ and stack them by row to take the place of the unstructured Gaussian matrix $\boldsymbol{Z}$ in Eq. (4), i.e., replacing $\boldsymbol{Z}$ with projection matrix $\boldsymbol{\Pi} = [\boldsymbol{P}^{(1)}; \boldsymbol{P}^{(2)}; \cdots; \boldsymbol{P}^{(t)}]$. The first column of the corresponding circulant random matrix $\boldsymbol{C}_{[d]}^{(i)}$ of $\boldsymbol{P}^{(i)} (i = 1, 2, \ldots, t)$ is randomly drawn i.i.d. from a Gaussian distribution $\mathcal{N}(0, \mathbf{I}/\sigma^2)$. Therefore, we obtain a novel random feature mapping, called Signed Circulant Random Feature mapping (SCRF),

$$\boldsymbol{\Phi}_{\mathrm{SCRF}} : \boldsymbol{x} \mapsto \frac{\sqrt{2}}{\sqrt{D}} \cos(\boldsymbol{\Pi}\boldsymbol{x} + \boldsymbol{b}). \tag{9}$$

If $d \nmid D$, we can generate $t = \lceil D/d \rceil$ signed circulant Gaussian matrices. By combining these t signed circulant Gaussian matrices in a column, after random feature mapping we take the $\{1, 2, \ldots, d, \ldots, (t-1)d, (t-1)d+1, \ldots, D\}$-th entries of Eq. (9) as the final random feature mapping. In this paper, without any specification, we assume that $d \mid D$.

With the structured random matrix projection, we can implement the SCRF efficiently by using FFT.

Lemma 2 (Computational Complexity). *The feature mappings of SCRF* $(Eq.(9))$ *can be computed in* $O(D \log d)$ *time and* $O(D)$ *space complexity. To predict an input data is in* $O(D \log d)$ *time complexity.*

Proof. For the i-th block (Lemma 1), the corresponding feature representation $\boldsymbol{x}^{(i)}$ can be calculated as follows,

$$\boldsymbol{x}^{(i)} = \boldsymbol{P}^{(i)}\boldsymbol{x} = \frac{1}{d}\mathrm{diag}(\boldsymbol{\sigma}_i)\boldsymbol{F}^*\mathrm{diag}(\boldsymbol{F}\boldsymbol{c}_i)\boldsymbol{F}\boldsymbol{x}.$$

To compute $\boldsymbol{x}^{(i)}$ could proceed efficiently by using FFT and the inverse FFT, which is in $O(d \log d)$ time complexity. Changing the signs of rows will be in $O(d)$ time complexity. $\boldsymbol{x}^{(i)} + \boldsymbol{b}_i$ will cost $O(d)$ time complexity. Only $\boldsymbol{\sigma}_i$, $\boldsymbol{c}_i$ and $\boldsymbol{b}_i$ need to be stored, which is in $O(3d)$ space complexity. To sum up, the total time complexity is $O(D \log d)$ and the space complexity is $O(3D)$.

Obviously, the predicting function has the form $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{x}) \rangle$, which will be computed in $O(D \log d)$ time complexity. □

3.2 Theoretical Analysis

The goal of this subsection is to analyse the approximation quality of SCRF. Lemma 3 states that the approximation using our proposed SCRF is an unbiased estimate of the Gaussian kernel. Corollary 2 shows that the variance of the approximation using SCRF is the same with that using RKS and much smaller than that using Fastfood. SCRF has smaller concentration error than Fastfood as shows in Theorem 3.

Lemma 3. *The expected feature mapping* $(Eq.(9))$ *recovers the Gaussian kernel, i.e.,*

$$\mathbb{E}\left[\langle \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{x}), \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{y}) \rangle\right] = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right). \quad (10)$$

Proof. Any given row of a circulant Gaussian matrix is a Gaussian vector with distribution $\mathcal{N}(0, \mathbf{I}/\sigma^2)$.

$$\begin{aligned}\mathbb{E}\left[\langle \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{x}), \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{y}) \rangle\right] &= \mathbb{E}\left[\cos(\sigma_i \boldsymbol{C}_{i\cdot}(\boldsymbol{x}-\boldsymbol{y}))\right] \\ &= \mathbb{E}\left[\cos(\boldsymbol{C}_{i\cdot}(\boldsymbol{x}-\boldsymbol{y}))\right].\end{aligned}$$

Obviously, Eq. (10) holds. □

Remark 1. *Any two rows of circulant Gaussian matrix are considerably more correlated, which causes increasing variance of approximating Gaussian kernel. We change signs of the rows randomly to eliminate the correlations and define the signed circulant random matrix. Theoretical analysis states that our proposed random feature mapping with signed circulant matrix projection is unbiased and does not increase the variance of the approximation. Fastfood generates correlated random features while SCRF generates independent random features.*

Theorem 2. *Assume that* $\boldsymbol{P}$ *is a signed circulant Gaussian matrix with the associated circulant Gaussian matrix* $\boldsymbol{C} = \mathrm{Circ}(c_0, c_1, \ldots, c_{d-1})$*, where* $c_0, c_1, \ldots, c_{d-1}$ *are randomly drawn i.i.d. from* $\mathcal{N}(0, 1/\sigma^2)$*. Let* $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{y}$ *and* $\phi_j(\boldsymbol{v}) = \cos([\boldsymbol{P}\boldsymbol{v}]_j)$*, the estimate of the kernel value that comes from the* k*-th pair of random features for each* $k \in \{0, 1, \ldots, d-1\}$*. Then for each* k *we have*

$$\mathrm{Var}[\phi_k(\boldsymbol{v})] = \frac{\left(1 - \mathrm{e}^{-\|\frac{\boldsymbol{v}}{\sigma}\|^2}\right)^2}{2}, \quad (11)$$

and

$$\mathrm{Var}\left[\frac{1}{d}\sum_{j=0}^{d-1}\phi_j(\boldsymbol{v})\right] = \frac{\left(1 - \mathrm{e}^{-\|\frac{\boldsymbol{v}}{\sigma}\|^2}\right)^2}{2d}. \quad (12)$$

Proof. $\boldsymbol{P}_{k\cdot} = \sigma_k \boldsymbol{C}_{k\cdot}$, $k \in \{0, 1, \ldots d-1\}$. Sign changes retain Gaussianity so that $\boldsymbol{P}_{k\cdot}$ is also Gaussian vector with i.i.d entries, where $\mathbb{E}[\boldsymbol{P}_{kj}] = \mathbb{E}[\sigma_k \boldsymbol{C}_{kj}] = 0$ and

$$\mathrm{Var}[\boldsymbol{P}_{kj}] = \mathrm{Var}[\sigma_k \boldsymbol{C}_{kj}] = \mathbb{E}[\sigma_k^2 \boldsymbol{C}_{kj}^2] - \mathbb{E}[\sigma_k \boldsymbol{C}_{kj}]^2 = 1/\sigma^2,$$

for $j = 0, 1, \ldots, d-1$. Therefore, $\boldsymbol{P}_{kj} \sim \mathcal{N}(0, 1/\sigma^2)$.

Let $\boldsymbol{z} = \boldsymbol{P}\boldsymbol{v}$. We have $\mathbb{E}[\boldsymbol{z}_k] = \mathbb{E}\left[\sum_{a=0}^{d-1}\boldsymbol{P}_{ka}\boldsymbol{v}_a\right] = 0$ and

$$\mathrm{Var}[\boldsymbol{z}_k] = \mathrm{Var}\left[\sum_{a=0}^{d-1}\boldsymbol{P}_{ka}\boldsymbol{v}_a\right] = \sum_{a=0}^{d-1}\boldsymbol{v}_a^2\mathrm{Var}[\boldsymbol{P}_{ka}] = \frac{\|\boldsymbol{v}\|^2}{\sigma^2},$$

so $\boldsymbol{z}_k \sim \mathcal{N}(0, \|\boldsymbol{v}\|^2/\sigma^2)$. Therefore, we have

$$\mathbb{E}[\cos(\boldsymbol{z}_k)] = \exp\left(-\left\|\frac{\boldsymbol{v}}{\sigma}\right\|^2/2\right).$$

Hence, we can obtain the following equation

$$\begin{aligned}\mathrm{Var}[\phi_k(\boldsymbol{v})] &= \mathbb{E}\cos^2(\boldsymbol{z}_k) - [\mathbb{E}\cos(\boldsymbol{z}_k)]^2 \\ &= \mathbb{E}\left[\frac{1}{2}\left(1 + \cos(2\boldsymbol{z}_k)\right)\right] - [\mathbb{E}\cos(\boldsymbol{z}_k)]^2 \\ &= \frac{1}{2}\left(1 - \mathrm{e}^{-\|\frac{\boldsymbol{v}}{\sigma}\|^2}\right)^2.\end{aligned}$$

For $j \neq k$,

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{z}_j, \boldsymbol{z}_k) &= \sum_{a,b=0}^{d-1}\boldsymbol{v}_a\boldsymbol{v}_b\mathrm{Cov}(\boldsymbol{P}_{ja}, \boldsymbol{P}_{kb}) \\ &= \sum_{a,b=0}^{d-1}\boldsymbol{v}_a\boldsymbol{v}_b\left(\mathbb{E}[\boldsymbol{P}_{ja}\boldsymbol{P}_{kb}] - \mathbb{E}[\boldsymbol{P}_{ja}]\mathbb{E}[\boldsymbol{P}_{kb}]\right) \\ &= \sum_{a,b=0}^{d-1}\boldsymbol{v}_a\boldsymbol{v}_b\mathbb{E}[\sigma_j\boldsymbol{C}_{ja}\sigma_k\boldsymbol{C}_{kb}] = 0.\end{aligned}$$

Therefore, the correlation coefficient ρ_{jk} between $\boldsymbol{z}_j$ and $\boldsymbol{z}_k$ equals to zero,

$$\rho_{jk} = \frac{\mathrm{Cov}(\boldsymbol{z}_j, \boldsymbol{z}_k)}{\sqrt{\mathrm{Var}[\boldsymbol{z}_j]}\sqrt{\mathrm{Var}[\boldsymbol{z}_k]}} = 0.$$

Obviously, Eq. (12) holds.

The proof completes. □

Corollary 2. *Let $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{y}$. Then for the feature mapping of SCRF ($Eq.$(9)), $\boldsymbol{\Phi} : \mathbb{R}^d \to \mathbb{R}^D$, obtained by stacking D/d signed circulant Gaussian matrices, we have that*

$$\mathrm{Var}[\langle \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{x}), \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{y})\rangle] = \frac{\left(1 - \mathrm{e}^{-\|\frac{\boldsymbol{v}}{\sigma}\|^2}\right)^2}{2D}. \quad (13)$$

Proof. $\langle \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{x}), \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{y})\rangle$ is the average of D/d independent esitmates. Following the proof of Theorem 2, if the two rows $\boldsymbol{P}_{j\cdot}$ and $\boldsymbol{P}_{k\cdot}$ are in distinct signed circulant Gaussian matrix blocks, the corresponding correlation coefficient ρ_{jk} between $\boldsymbol{z}_j$ and $\boldsymbol{z}_k$ is also zero. Obviously, this corollary holds. □

All of the three approximations are unbiased, but lower variance means tighter concentration error. Corollary 2 states that SCRF approximates kernel with the same variance as RKS and smaller variance than Fastfood. Combining Corollary 2 and Chebyshev's inequality, we can easily obtain the concentration error of SCRF, which will converge quickly as D increases with probability $1 - \delta$.

Theorem 3 (Concentration Error). *For the feature mapping of SCRF ($Eq.$(9)), $\boldsymbol{\Phi} : \mathbb{R}^d \to \mathbb{R}^D$, obtained by stacking D/d signed circulant Gaussian matrices, and any $\delta \in (0, 1)$, the following inequality holds with probability $1 - \delta$,*

$$|\langle \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{x}), \boldsymbol{\Phi}_{\mathrm{SCRF}}(\boldsymbol{y})\rangle - k(\boldsymbol{x}, \boldsymbol{y})| \leq \frac{1 - \mathrm{e}^{-\|\frac{\boldsymbol{x}-\boldsymbol{y}}{\sigma}\|^2}}{\sqrt{2\delta D}}.$$

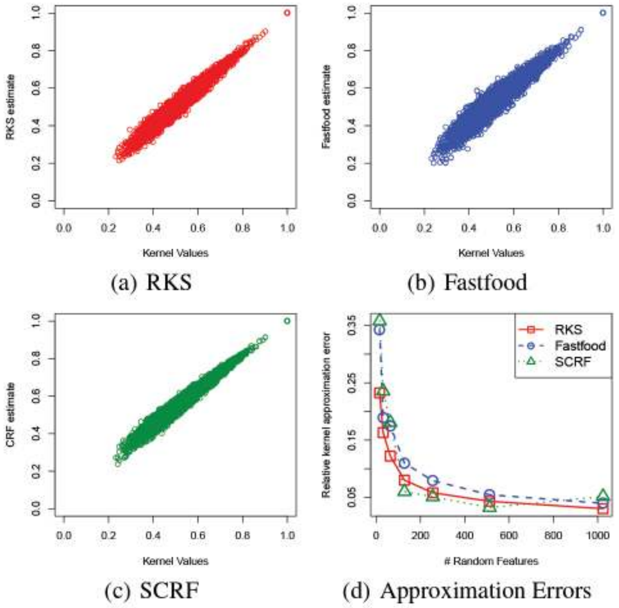
4 Experiments

We implement random feature mappings in R 3.1.1 and conduct experiments on a public SUSE Linux enterprise server 10 SP2 platform with 2.2GHz AMD Opteron Processor 6174 CPU and 48GB RAM. We compare the performance of random feature mappings, including RKS, Fastfood and our SCRF in terms of kernel approximation, efficiency of kernel expansion and generalization performance.

4.1 Kernel Approximation

First, we evaluate the kernel estimates from RKS, Fastfood and SCRF. We uniformly sample $l = 100$ vectors from $[0, 1]^{16}$, and set $D = 512$ and kernel parameter $\gamma = 0.25$ ($\gamma = 1/(2\sigma^2)$). Figure 1(a)–1(c) show kernel estimates from the three methods plotted against exact kernel values respectively. Each point represents a combination of two vectors $\boldsymbol{x}, \boldsymbol{y}$. The coordinate corresponds to $(k(\boldsymbol{x}, \boldsymbol{y}), \langle \boldsymbol{\Phi}(\boldsymbol{x}), \boldsymbol{\Phi}(\boldsymbol{y})\rangle)$. A perfect mapping would manifest as a narrow 45-degree line. As we can see, both RKS and SCRF perform a little better than Fastfood in terms of kernel estimates, which coincides with the fact that the variance of approximation using SCRF is the same with that using RKS and smaller than that using Fastfood.

Next, we investigate the convergence of kernel approximation using SCRF. We uniformly sample $l = 500$ vectors from $[0, 1]^{16}$ and compare RKS, Fastfood and SCRF with the exact kernel values. Figure 1(d) shows the relative kernel approximation error ($\|\widehat{\boldsymbol{K}} - \boldsymbol{K}\|_{\mathrm{F}} / \|\boldsymbol{K}\|_{\mathrm{F}}$) w.r.t. D. From Figure 1(d), all the three approaches converge quickly to the exact kernel values as D increases, where both RKS and SCRF converge faster than Fastfood.



(a) RKS

(b) Fastfood

(c) SCRF

(d) Approximation Errors

Figure 1: Kernel estimates and kernel approximation errors of different random feature mappings.

Table 1: Computation time, speedup and memory improvement of Fastfood and SCRF relative to RKS.

d	RKS	Fastfood	Speed	RAM	SCRF	Speed	RAM
512	39.8s	18.8s	2.1x	109x	9.3s	4.3x	164x
1024	113.1s	18.9s	6.0x	223x	9.6s	11.7x	334x
2048	255.0s	21.7s	11.8x	450x	10.5s	24.1x	675x
4096	424.1s	27.0s	15.7x	905x	10.6s	40.1x	1358x

4.2 Efficiency of Kernel Expansion

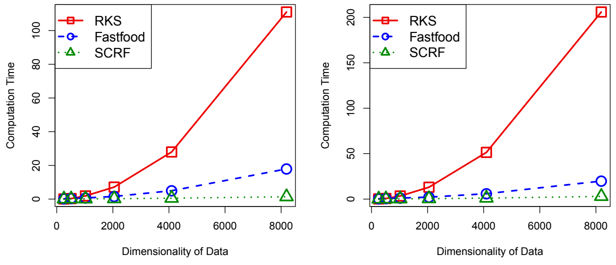
In this subsection, we compare the CPU time of computing random feature mappings of the three approaches. We uniformly sample $l = 5000$ vectors from $[0, 1]^d$ and set $D = 8192$. As analysed above, both Fastfood and SCRF share $O(lD \log d)$ time complexity while RKS requires $O(ldD)$ time. Obviously, the running time of both Fastfood and SCRF are less dependent on d, a very promising property since random feature mapping often contributes a significant computational cost in predicting.

Table 1 shows the CPU time in seconds of the three methods with speedup and memory improvement. The running time of both Fastfood and SCRF is almost independent on d. It costs more time in generating random parameters for Fastfood than SCRF, thus SCRF runs a little faster than Fastfood. Both Fastfood and SCRF save much storage compared with RKS. However, SCRF uses around 1.5x less storage than Fastfood. This is because Fastfood needs to store $5D$ parameters while SCRF only stores $3D$ parameters.

For higher-dimensional problems, we need to increase D to boost the accuracy, $D = O(d)$. Figure 2 shows that the CPU time of computing $D = d$ random feature mapping with RKS is quadratic with d, a bottleneck of kernel methods on high-dimensional datasets. However, both Fastfood and SCRF scale well in this case. Especially, SCRF runs faster and uses less memory than Fastfood.

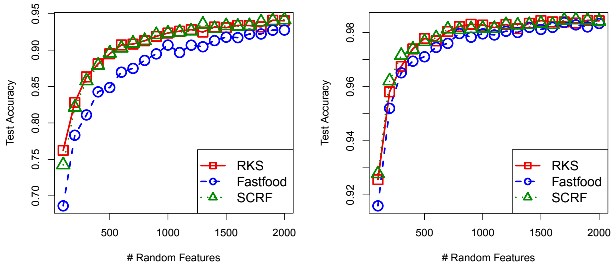
Table 2: Comparison of RKS, Fastfood, SCRF with LIBLINEAR and Gaussian kernel with LIBSVM. Parameters (C, γ) are selected by 5-fold cross validation with Gaussian kernel SVM. Test accuracy and training time + predicting time are listed.

Dataset	log(C)	log(γ)	D	LIBSVM	RKS+LIBLINEAR	Fastfood+LIBLINEAR	SCRF+LIBLINEAR
splice ($d = 60$)	0	-8	400	86.90% 0.7s+0.4s	86.26±0.50% 0.2s+0.4s	86.23±0.34% 0.7s+0.8s	86.31±0.65% 0.2s+0.4s
dna ($d = 180$)	2	-6	1000	95.44% 4.2s+0.8s	92.34±0.67% 2.6s+0.6s	90.70±0.50% 3.8s+1.1s	92.34±0.43% 2.1s+0.4s
mushrooms ($d = 112$)	0	-4	100	100.0% 2.5s+0.8s	99.57±0.12% 0.3s+0.2s	99.31±0.33% 1.2s+1.0s	99.26±0.23% 0.2s+0.1s
usps ($d = 256$)	6	-6	2000	95.56% 55.9s+5.8s	94.83±0.21% 35.4s+2.5s	95.05±0.28% 40.8s+3.4s	94.31±0.11% 31.9s+1.3s
a9a ($d = 123$)	4	-6	1000	85.11% 6.2m+42.1s	85.17±0.07% 28.7s+7.1s	85.14±0.03% 49.3s+16.3s	85.15±0.09% 30.8s+7.2s
w8a ($d = 300$)	2	-4	2000	99.08% 3.6m+12.0s	99.07±0.01% 1.9m+21.8s	99.05±0.05% 2.3m+24.5s	99.08±0.05% 1.4m+10.5s
ijcnn1 ($d = 22$)	0	-8	500	98.78% 2.0m+34.7s	97.78±0.18% 22.5s+12.8s	97.10±0.17% 61.9s+79.2s	97.66±0.23% 25.1s+19.2s
mnist ($d = 784$)	6	-6	3000	98.42% 53.5m+3.3m	97.29±0.14% 14.2m+49.6s	96.85±0.13% 11.9m+26.6s	96.38±0.17% 11.4m+13.7s
cifar10 ($d = 3072$)	2	-8	10000	56.40% 63.1h+2.8h	50.09% 4.0h+21.8m	48.77% 3.1h+2.0m	45.09% 3.0h+54.8s



(a) $l = 500$ (b) $l = 1000$

Figure 2: CPU time of computing random feature mappings of the three approaches w.r.t. d.



(a) dna (b) ijcnn1

Figure 3: Test accuracies of RKS, Fastfood and SCRF with LIBLINEAR w.r.t. D on dna and ijcnn1.

4.3 Generalization Performance

In this subsection, we compare RKS, Fastfood and SCRF with non-linear SVMs on 9 well-known classification benchmark datasets of size ranging from 1,000 to 60,000 with dimensionality ranging from 22 to 3,072. For mushrooms, we select 4,062 training data randomly and the left as test data. We use LIBSVM [Chang and Lin, 2011] for non-linear kernels and LIBLINEAR [Fan *et al.*, 2008] for random feature mappings for classification task. All averages and standard deviations are over 5 runs of the algorithms except on cifar10. We select the kernel parameter γ and regularization coefficient C of Gaussian kernel SVM by using 5-fold cross validation with LIBSVM.

Figure 3 depicts the test accuracies of RKS, Fastfood and SCRF with LIBLINEAR w.r.t. D on dna and ijcnn1. Their generalization performances are not obviously distinguishable as D increases. Table 2 shows the results of the comparison. There is virtually no difference among them in terms of test accuracy except on cifar10. All of the results of the three approaches on cifar10, which come from one random trial respectively, are worse than LIBSVM. This is because the selected γ and C by LIBSVM may not be optimal for the other three approaches. Except on mnist and cifar10, Fastfood runs slower than RKS and SCRF in both training and predicting. This is because Fastfood needs to format $d = 2^k, k \in \mathbb{N}$, through padding the vector with zeros, which demands more time for preprocessing input data and random feature mapping. In addition, if d is not so large that there will be no speedup of computing random feature mappings. For larger dimensional dataset mnist/cifar10, Fastfood can obtain efficiency gains. In general, SCRF is more time and space efficient than both Fastfood and RKS.

5 Conclusion

In this paper, we have proposed a random feature mapping method for approximating Gaussian kernel using signed circulant matrix projection that makes the approximation unbiased and have lower variance. The adoption of circulant matrix projection guarantees a quasilinear random feature mapping and promotes scalable and practical kernel methods for large scale machine learning.

Acknowledgement

This work was supported in part by National Natural Foundation of China (No. 61170019), 973 Program (2013CB329304), National Natural Foundation of China (No. 61222210) and Key Program of National Natural Science Foundation of China (No. 61432011).

References

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

[Chang *et al.*, 2007] Edward Y Chang, Kaihua Zhu, Hao Wang, Hongjie Bai, Jian Li, Zhihuan Qiu, and Hang Cui. PSVM: Parallelizing support vector machines on distributed computers. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 257–264, 2007.

[Davis, 1979] Philip J. Davis. *Circulant Matrices*. John Wiley & Sons, 1979.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[Gray, 2006] Robert M Gray. *Toeplitz and circulant matrices: A review*. Now Publishers Inc, 2006.

[Hamid *et al.*, 2014] Raffay Hamid, Ying Xiao, Alex Gittens, and Dennis DeCoste. Compact random feature maps. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, pages 19–27, 2014.

[Joachims, 2006] Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 217–226, 2006.

[Kar and Karnick, 2012] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, pages 583–591, 2012.

[Kimeldorf and Wahba, 1970] George S Kimeldorf and Grace Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495–502, 1970.

[Le *et al.*, 2013] Quoc Le, Tamás Sarlós, and Alexander J. Smola. Fastfood — Approximating kernel expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, pages 244–252, 2013.

[Li *et al.*, 2010] Fuxin Li, Catalin Ionescu, and Cristian Sminchisescu. Random Fourier approximations for skewed multiplicative histogram kernels. *Lecture Notes in Computer Science*, 6376:262–271, 2010.

[Maji and Berg, 2009] Subhransu Maji and Alexander C. Berg. Max-margin additive classifiers for detection. In *Proceedings of 12th International Conference on Computer Vision (ICCV 2009)*, pages 40–47, 2009.

[Pham and Pagh, 2013] Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2013)*, pages 239–247, 2013.

[Platt, 1999] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, chapter Support Vector Learning, pages 185–208. MIT Press, 1999.

[Rahimi and Recht, 2007] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 1177–1184, 2007.

[Rahimi and Recht, 2008] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems 21 (NIPS 2008)*, pages 1313–1320, 2008.

[Rudin, 2011] Walter Rudin. *Fourier Analysis on Groups*. John Wiley & Sons, 2011.

[Schölkopf and Smola, 2002] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[Steinwart, 2003] Ingo Steinwart. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4:1071–1105, 2003.

[Tsang *et al.*, 2005] Ivor W Tsang, James T Kwok, Pak-Ming Cheung, and Nello Cristianini. Core vector machines: Fast SVM Training on very large data sets. *Journal of Machine Learning Research*, 6(4):363–392, 2005.

[Tyrtyshnikov, 1996] Evgenij E Tyrtyshnikov. A unifying approach to some old and new theorems on distribution and clustering. *Linear Algebra and its Applications*, 232:1–43, 1996.

[Vapnik, 1998] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[Vedaldi and Zisserman, 2012] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–492, 2012.

[Yang *et al.*, 2014] Jiyan Yang, Vikas Sindhwani, Haim Avron, and Michael W. Mahoney. Quasi-Monte Carlo feature maps for shift-invariant kernels. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, pages 485–493, 2014.

[Yen *et al.*, 2014] En-Hsu Yen, Ting-Wei Lin, Shou-De Lin, Pradeep K. Ravikumar, and Inderjit S. Dhillon. Sparse random feature algorithm as coordinate descent in Hilbert space. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 2456–2464, 2014.