Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Random Features for Shift-Invariant Kernels with Moment Matching

Weiwei Shen,[†,‡] **Zhihui Yang,**[‡] **Jun Wang**[†]
[†]School of Computer Science and Software Engineering
East China Normal University, Shanghai, China
[‡]GE Global Research Center, Niskayuna, NY, USA,
realsww@gmail.com, yangz@ge.com, wongjun@gmail.com

Abstract

In order to grapple with the conundrum in the scalability of kernel-based learning algorithms, the method of approximating nonlinear kernels via random feature maps has attracted wide attention in large-scale learning systems. Specifically, the associated sampling procedure is one critical component that dictates the quality of random feature maps. However, for high-dimensional features, the standard Monte Carlo sampling method has been shown to be less effective in producing low-variance random samples. In consequence, it demands constructing a large number of features to attain the desired accuracy for downstream use. In this paper, we present a novel sampling algorithm powered by moment matching techniques to reduce the variance of random features. Our extensive empirical studies and comparisons with several highly competitive peer methods verify the superiority of the proposed algorithm in Gram matrix approximation and generalization errors in regression. Our rigorous theoretical proofs justify that the proposed algorithm is guaranteed achieving lower variance than the standard Monte Carlo method in high dimensional settings.

1 Introduction

Kernel methods have evoked remarkable repercussions in machine learning tasks, ranging from regression to classification to image reconstruction (Schölkopf and Smola 2002). As kernel methods allow malleable generalization of algorithms developed in explicit linear feature spaces to implicit nonlinear feature spaces, nonlinear structures of data can be efficaciously explored. Specifically, the well-known *kernel trick* allows us to circumvent operating in high (often infinite) dimensional nonlinear feature spaces through directly exploiting nonlinear kernel functions (Aizerman, Braverman, and Rozoner 1964). By reaping benefits of obviating explicitly computing coordinates in those high-dimensional feature spaces, computational costs are dramatically saved. While algorithms in high-dimensional feature spaces inevitably encounter *the curse of dimensionality*, the classical Representation Theorems guarantee the existence of finite-dimensional solutions to associated optimization problems even in infinite-dimensional feature spaces (Argyriou, Micchelli, and Pontil 2009). However, this solution to the curse of dimensionality renders *the curse of support*, i.e., that kernel-based learning algorithms scale poorly with the number of training samples (Bengio, Delalleau, and Roux 2005). For example, out-of-sample evaluation requires computing the kernel measures between a new data point and all the training data. Hence, it becomes the tradeoff between generalization accuracy and computational costs.

In order to eradicate the challenge in the scalability of kernel methods, intensive research has been conducted over recent years. The seminal paper by (Rahimi and Recht 2007) provides one solution to this difficulty via a randomized construction of low-dimensional approximate *feature maps*. Denote $k(\mathbf{x}, \mathbf{y}) = \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle_{\mathcal{H}}$ as a kernel function, which is an inner product of the associated feature map $\Psi \colon \mathbb{R}^d \to \mathcal{H}$, with two data points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and a Hilbert space $\mathcal{H}$. The essence of randomized feature construction is to embed the original nonlinear feature space $\mathcal{H}$ into a relatively low-dimensional Euclidean space while incurring an arbitrarily small distortion in the inner product values. Mathematically, the explicit randomized feature map from the d-dimensional data to an m-dimensional Euclidean inner product space can be represented as $\mathbf{Z} \colon \mathbb{R}^d \to \mathbb{C}^m$, and the kernel value can be approximately evaluated by the inner product between the transformed data pair, i.e., $k(\mathbf{x}, \mathbf{y}) \approx \langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle_{\mathbb{C}^m}$, where $\mathbb{C}^m$ denotes an m-dimensional complex Euclidean space with the inner product $\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle_{\mathbb{C}^m} = \sum_{j=1}^{m} \alpha_j \beta_j^*$, with vectors $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^\top$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)^\top$. Its theoretical foundation builds upon the classical Bochner's theorem that links a continuous shift-invariant kernel to a unique probability density (Bochner 1933), thereby allowing random sampling to play a critical role in defining random feature maps. Thus, the performance of this construction predominately relies on the quality of the elected sampling procedure. Yet, that procedure has been relatively less acknowledged, highlighted and studied. Consequently, approximating the kernel up to sufficient accuracy generally demanding a large sample size m to perform well has fatally affected the applicability and popularization of the method of random features (Hamid et al. 2014; Huang et al. 2014).

Recently, the illuminating paper by (Yang et al. 2014a) demonstrates one effective way of replacing pseudo-random numbers by quasi-random numbers to improve the sampling efficiency, i.e., reducing the sample size to achieve the desired accuracy. However, quasi-

random numbers often perform markedly poorly in high-dimensional sampling (Niederreiter 1992). Its convergence rate $O((\log m)^d/m)$ indicates its use only for problems of moderately high dimension d, with the suggested upper limit at 40 dimensions (Glasserman 2003). In other words, its advantages over pseudo-random numbers in enjoying a faster convergence rate gradually disappear in high dimensions.

To cope with those challenges, in this paper, we propose an algorithm to advance the efficiency of random features approximation to shift-invariant kernels via a moment matching sampling method. For validation, we provide detailed theoretical proofs and empirical comparisons with six state-of-the-art sampling methods across four standard benchmarks. Specifically, using the Gaussian kernel as the notable representative, we prove that to attain the same accuracy the proposed algorithm is guaranteed requiring fewer features than the standard Monte Carlo sampling method in (Rahimi and Recht 2007). Further, theoretically and empirically, we show that different from the quasi-Monte Carlo sampling based method in (Yang et al. 2014a) the effectiveness of the new algorithm remains robust in high dimensions. Furthermore, our empirical experiments illustrate the superiority of the proposed approach in the approximation of Gram matrices with comparable downstream generalization errors. Additionally, the new method is easy to implement and to combine with other random feature construction techniques without necessitating extensive structural changes.

2 Background and Related Work

In this section, we first recount the Monte Carlo method, and then describe the backbone and the recent research of the kernel approximation by random feature maps.

Monte Carlo Method

In this part, we offer an overview of the Monte Carlo method. More discussions about methodology and applications can be found in (Caflisch 1998).

Given a function $f(\mathbf{u})$ with $\mathbf{u}$ being a uniform random variable over $[0,1]^d$, i.e., $\mathbf{u} \sim U[0,1]^d$, the expectation of $f(\mathbf{u})$ can be estimated by the average of m simulation trials:

$$\mathbb{E}[f] = \int_{[0,1]^d} f(\mathbf{u}) d\mathbf{u} \approx \frac{1}{m}\sum_{j=1}^{m} f(\mathbf{u}_j), \tag{1}$$

where $\{\mathbf{u}_j\}_{j=1}^m$ are uniform random samples from $U[0,1]^d$. The central limit theorem states that the root mean square error of this approximation decays at a rate of $O_p(m^{-1/2})$. Although the slow convergence rate has weakened its competitiveness in low-dimensional problems, being independent of dimensions, such a convergence rate promotes its vast use in high-dimensional applications.

To address its shortcoming of having a slow error decay, researchers appeal to designing effective variance reduction methods for specific applications (Caflisch 1998). A variety of variance reduction methods, such as the method of control variates, importance sampling, moment matching and antithetic sampling, are necessarily armed by Monte Carlo algorithms in practical systems (Glasserman 2003). Those methods bolster the computational efficiency of the associated numerical methods by devising low-variance samples as to achieve the same level of accuracy with fewer samples. While they cannot speed up the convergence rate by changing the factor $m^{-1/2}$, they have demonstrated substantial benefits in practice by reducing the commonly large multiplicative factor (Niederreiter 1992). Therefore, Monte Carlo algorithms along with variance reduction techniques remain among the topmost choices in practice.

Related Work

In this part, we recapitulate the algorithm of generating random feature maps. First, the classical Bochner's theorem characterizes the class of positive definite functions:

Theorem 0 ((Bochner 1933))**.** *A complex-valued continuous function $g : \mathbb{R}^d \to \mathbb{C}$ is positive definite if and only if it is the Fourier transform of a finite nonnegative Borel measure μ on $\mathbb{R}^d$, i.e., for any $\mathbf{x} \in \mathbb{R}^d$,*

$$g(\mathbf{x}) = \int_{\mathbb{R}^d} e^{-i\mathbf{x}^\top \mathbf{w}} \, d\mu(\mathbf{w}). \tag{2}$$

A kernel function is called shift-invariant if it satisfies $k(\mathbf{x},\mathbf{y}) = g(\mathbf{x}-\mathbf{y})$ for some complex-valued positive definite function g on $\mathbb{R}^d$. By assuming that $\mu(\cdot)$ is a probability measure with the probability density function $p(\cdot)$, a scaled shift-invariant kernel, such as Gaussian kernel, can be rewritten as a characteristic function for a unique probability density. Namely, for two data points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$k(\mathbf{x},\mathbf{y}) = g(\mathbf{x}-\mathbf{y}) = \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{y})^\top \mathbf{w}} p(\mathbf{w}) \, d\mathbf{w}, \tag{3}$$

where the probability density function $p(\mathbf{w})$ is the inverse Fourier transform of $k(\cdot)$. For the Gaussian kernel $k(\mathbf{x},\mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2})$, the associated d-dimensional normal probability density is $p(\mathbf{w}) \sim \mathcal{N}_d(\mathbf{0}, \Sigma)$ with the diagonal covariance matrix being $\Sigma = \sigma^{-2} I_d \in \mathbb{R}^{d\times d}$. By denoting the data matrix as $\mathbf{X} \in \mathbb{R}^{n\times d}$, the corresponding Gram matrix of the kernel function $k(\cdot)$ can be denoted as $\mathbf{K} \in \mathbb{C}^{n\times n}$ with its element $\mathbf{K}_{lh}$ defined as $\mathbf{K}_{lh} = k(\mathbf{x}_l, \mathbf{x}_h)$, for any $l, h = 1, \ldots, n$. Hence, the kernel function (3) can be approximately computed via averaging over m simulation trials:

$$k(\mathbf{x},\mathbf{y}) \approx E_{MC} \equiv \langle \mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{y}) \rangle_{\mathbb{C}^m}, \tag{4}$$

where the m-dimensional random feature map $\mathbf{Z}(\mathbf{x})$ is defined as a row vector:

$$\mathbf{Z}(\mathbf{x}) = \frac{1}{\sqrt{m}}(e^{-i\mathbf{x}^\top \mathbf{w}_1}, \ldots, e^{-i\mathbf{x}^\top \mathbf{w}_m}) \in \mathbb{C}^{1\times m}, \tag{5}$$

and the random numbers $\{\mathbf{w}_j\}_{j=1}^m$ are sampled from $p(\mathbf{w})$. Apparently, the standard Monte Carlo estimator E_{MC} is unbiased, i.e., $\mathbb{E}(E_{MC}) = k(\mathbf{x},\mathbf{y})$. Denote its variance as $V_{MC} \equiv \mathrm{Var}(E_{MC})$.

Next, to sample random numbers from the d-dimensional multivariate normal probability density $p(\mathbf{w}) \sim \mathcal{N}_d(\mathbf{0}, \Sigma)$, the inverse transform sampling method is a common choice (Glasserman 2003). Specifically, given a d-dimensional uniform random variable $\mathbf{u}$, we take the inverse transform of

the cumulative density function of a standard normal distribution on each element of the vector, which produces a standard normal random variable $\Phi^{-1}(\mathbf{u}) \sim \mathcal{N}_d(\mathbf{0}, I_d)$. After scaling, we obtain the desired random variable as $\mathbf{w} = \sigma^{-1}\Phi^{-1}(\mathbf{u}) \sim \mathcal{N}_d(\mathbf{0}, \Sigma)$. Accordingly, the kernel function in equation (3) is equivalent to

$$k(\mathbf{x}, \mathbf{y}) = \int_{[0,1]^d} e^{-i(\mathbf{x}-\mathbf{y})^\top \sigma^{-1}\Phi^{-1}(\mathbf{u})}\, d\mathbf{u}. \tag{6}$$

Combining equations (1), (4) and (6) generates the random feature maps proposed in the pioneering work by (Rahimi and Recht 2007). Replacing pseudo-random numbers by quasi-random numbers in equation (1) together with equations (6) and (4) depicts the algorithm in (Yang et al. 2014a).

Further, besides the Gaussian kernel, various types of kernels have been studied for random feature maps. Dot-product and polynomial kernels are approximated in (Kar and Karnick 2012; Pham and Pagh 2013; Hamid et al. 2014; Pennington, Yu, and Kumar 2015). Histogram intersection kernels are investigated by (Maji and Berg 2009) and are further generalized to a class of additive homogeneous kernels in (Vedaldi and Zisserman 2012). The premise of shift-invariant kernels is relaxed and generalized to a larger class of group invariance in (Li, Ionescu, and Sminchisescu 2010).

Furthermore, to speed up the approximation process, efforts have been expended on extending the seminal work. An accelerated implementation of the randomized algorithm via Walsh-Hadamard transformations is proposed in (Le, Sarlós, and Smola 2013). One way to construct data-dependent random features is considered in (Chen et al. 2015). A sparse random feature algorithm of obtaining models without growing linearly with the number of random features is proposed in (Yen et al. 2014). The complex basis function is shown having the lowest embedding variance among several representations of Fourier basis functions in (Sutherland and Schneider 2015). The relation between random features and quadrature rules in integrals is explored in (Bach 2015). The performance of random features and the Nyström method is compared in (Yang et al. 2012). Naturally, the success of this randomized feature map approach enriches its versatility and promotes its development in a wide range of applications (Dai et al. 2014; Avron and Sindhwani 2015; Yang et al. 2014b; Lopez-Paz et al. 2014).

3 Methodology

In this section, we first describe the proposed moment matching sampling algorithm, then offer a theoretical underpinning to support its effectiveness, and finally provide detailed discussions.

Moment Matching

In order to accurately approximate kernel (3), a potent sampling method should be able to reduce sampling variance, thereby requiring constructing fewer features for downstream use. To this end, we propose to apply moment matching to enhance the quality of sampling.

Specifically, the proposed moment matching sampling algorithm is composed of three pivotal steps. First, like equation (6), m uncorrelated multivariate normal distributed samples with covariance matrix Σ are produced by the inverse transform $\{\mathbf{w}_j\}_{j=1}^m$ with $\mathbf{w}_j = \sigma^{-1}\Phi^{-1}(\mathbf{u}_j)$. Second, the truly uncorrelated standard normal distributed samples are constructed. For a finite number of samples, the sample mean of $\{\mathbf{w}_j\}_{j=1}^m$ would not exactly be zero and the corresponding sample covariance would not be the same as the desired covariance matrix Σ. Hence, by viewing $\{\mathbf{w}_j\}_{j=1}^m$ as weakly correlated samples, we apply a moment matching method by reversing the procedure of generating correlated normal distributed samples from uncorrelated normal distributed samples $\{\hat{\mathbf{w}}_j\}_{j=1}^m$ with $\hat{\mathbf{w}}_j = \hat{A}^{-1}(\mathbf{w}_j - \hat{\boldsymbol{\mu}})$, where $\hat{\boldsymbol{\mu}} = m^{-1}\sum_{j=1}^m \mathbf{w}_j$ is the sample mean, and $\hat{A}\hat{A}^\top = \mathrm{Cov}(\mathbf{w}_j - \hat{\boldsymbol{\mu}})$ is the square root decomposed sample covariance matrix of the centered samples $\{\mathbf{w}_j - \hat{\boldsymbol{\mu}}\}_{j=1}^m$. Thus, the new covariance matrix of $\{\hat{\mathbf{w}}_j\}_{j=1}^m$ is $\mathrm{Cov}(\hat{\mathbf{w}}_j) = I_d$. Third, in our particular case with the target covariance $\Sigma = \sigma^{-2} I_d$, the desired correlated random samples are generated by $\{\tilde{\mathbf{w}}_j\}$ with $\tilde{\mathbf{w}}_j = \sigma^{-1}\hat{\mathbf{w}}_j$, where the covariance matrix of the new samples matches Σ:

$$\mathrm{Cov}(\tilde{\mathbf{w}}_j) = \mathrm{Cov}(\sigma^{-1}\hat{\mathbf{w}}_j) = \sigma^{-2}\mathrm{Cov}(\hat{\mathbf{w}}_j) = \Sigma. \tag{7}$$

In sum, with the proposed moment matching method the Gaussian kernel is approximated by

$$k(\mathbf{x}, \mathbf{y}) \approx E_{MM} \equiv \frac{1}{m}\sum_{j=1}^m e^{-i\mathbf{c}^\top \tilde{\mathbf{w}}_j} = \langle \mathbf{Q}(\mathbf{x}), \mathbf{Q}(\mathbf{y}) \rangle_{\mathbb{C}^m} \tag{8}$$

with

$$\mathbf{Q}(\mathbf{x}) = \frac{1}{\sqrt{m}}(e^{-i\mathbf{x}^\top \tilde{\mathbf{w}}_1}, \ldots, e^{-i\mathbf{x}^\top \tilde{\mathbf{w}}_m}), \tag{9}$$

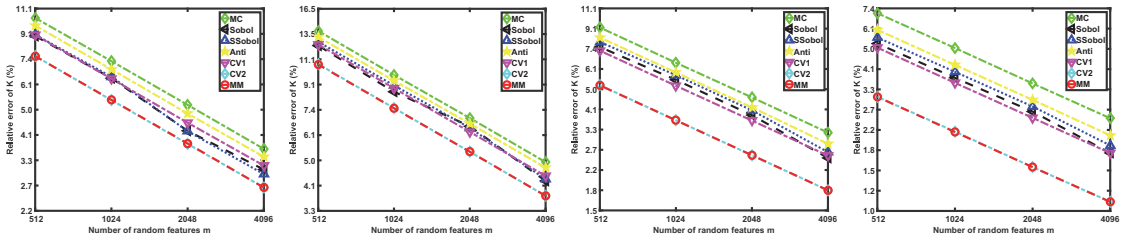
where we denote the moment matching estimator as E_{MM} and have the set of updated samples $\{\tilde{\mathbf{w}}_j\}_{j=1}^m$ with $\tilde{\mathbf{w}}_j = \sigma^{-1}\hat{A}^{-1}(\Phi^{-1}(\mathbf{u}_j) - \hat{\boldsymbol{\mu}})$. Denote its variance as $V_{MM} \equiv \mathrm{Var}(E_{MM})$. Algorithm 1 summarizes the proposed procedure of constructing random feature maps $\mathbf{Q}(\mathbf{x})$.

Algorithm 1 Random Features with Moment Matching

1: **Inputs:** Data $\mathbf{X}$; Parameters σ, m.
2: Draw m d-dimensional uniform samples $\{\mathbf{u}_j\}_{j=1}^m$;
3: Generate m d-dimensional normal samples by the inverse transform: $\{\mathbf{w}_j\}_{j=1}^m$ with $\mathbf{w}_j = \sigma^{-1}\Phi^{-1}(\mathbf{u}_j)$;
4: Compute the sample mean $\hat{\boldsymbol{\mu}}$ of $\{\mathbf{w}_j\}_{j=1}^m$ and the square root matrix $\hat{A}$ of the sample covariance matrix of $\{\mathbf{w}_j - \hat{\boldsymbol{\mu}}\}_{j=1}^m$: $\hat{\boldsymbol{\mu}} = \frac{1}{m}\sum_{j=1}^m \mathbf{w}_j$ and $\hat{A}\hat{A}^\top = \mathrm{Cov}(\mathbf{w}_j - \hat{\boldsymbol{\mu}})$;
5: Generate the truly uncorrelated m d-dimensional standard normal samples by moment matching: $\{\hat{\mathbf{w}}_j\}_{j=1}^m$ with $\hat{\mathbf{w}}_j = \hat{A}^{-1}(\mathbf{w}_j - \hat{\boldsymbol{\mu}})$;
6: Generate the desired samples $\{\tilde{\mathbf{w}}_j\}_{j=1}^m$: $\tilde{\mathbf{w}}_j = \sigma^{-1}\hat{\mathbf{w}}_j$;
7: Form the random feature maps: $\mathbf{Q}(\mathbf{x}) = m^{-1/2}(e^{-i\mathbf{x}^\top \tilde{\mathbf{w}}_1}, \ldots, e^{-i\mathbf{x}^\top \tilde{\mathbf{w}}_m})$;
8: **Output:** $\mathbf{Q}(\mathbf{x}) \colon \mathbb{R}^d \to \mathbb{C}^m$.

Theoretical Analysis

In this part, we show that to attain the same approximation accuracy the moment matching estimator E_{MM} re-



(a) YP90 (b) QM274 (c) MNIST300 (d) LR500

Figure 1: Errors of approximated Gram matrices in the Frobenuis norm by different sample sizes (log-log).

quires fewer features than the standard Monte Carlo estimator E_{MC} so that the computing for downstream uses such as classification and regression will be cheaper. Detailed proofs are given in the long version of this work. Let us denote by $\xrightarrow{P}$ the convergence in probability, and denote by $\mathbf{c} \equiv \mathbf{x} - \mathbf{y}$ the coefficient from data.

Definition 1. *The random vector sequence $\{X_j\}_{j=1}^m$ is uniformly tight means that for every $\epsilon > 0$ there is a constant M such that $\sup_j P(\| X_j \| > M) < \epsilon$.*

Definition 2. *Let $\{X_j\}_{j=1}^m$ be a sequence of random vectors. Let $\{R_j\}_{j=1}^m$ be a sequence of strictly positive number indexed by j. The following statements give the definitions of o_p and O_p: $X_j = o_p(1)$ if and only if $X_j \xrightarrow{P} \mathbf{0}$; $X_j = O_p(1)$ if and only if $\{X_j\}_{j=1}^m$ is uniformly tight.*

First, the unbiased control variate estimator based on the first two moments E_{CV} can be constructed as:

$$E_{CV} \equiv \frac{1}{m}\sum_{j=1}^{m}(e^{-i\mathbf{c}^\top \mathbf{w}_j} - \beta_1(-i\mathbf{c}^\top \mathbf{w}_j - 0) - \beta_2(-\frac{1}{2}\mathbf{c}^\top \mathbf{w}_j \mathbf{w}_j^\top \mathbf{c} + \frac{1}{2}\mathbf{c}^\top \Sigma \mathbf{c})), \tag{10}$$

where β_1 and β_2 are the control coefficients optimally determined later. Clearly, $\mathbb{E}(E_{CV}) = k(\mathbf{x}, \mathbf{y})$. Denote the variance of E_{CV} as $V_{CV} \equiv \mathrm{Var}(E_{CV})$, the symbol $\otimes$ as the Kronecker product, and the symbol $\mathrm{vec}(\cdot)$ as the vectorization operator which converts a matrix into a column vector.

Theorem 1. *The control variate estimator* (10) *can be rewritten as*

$$E_{CV} = E_{MC} + \beta_1(i\mathbf{c}^\top \hat{\boldsymbol{\mu}}) + \frac{1}{2}\beta_2(\mathbf{c}^\top \otimes \mathbf{c}^\top)\mathrm{vec}(\hat{\Sigma} - \Sigma), \tag{11}$$

and the optimal coefficients for the two controls that minimize V_{CV} are $\beta_1^* = e^{-\frac{1}{2}\mathbf{c}^\top \Sigma \mathbf{c}}$ and $\beta_2^* = e^{-\frac{1}{2}\mathbf{c}^\top \Sigma \mathbf{c}}$.

Next, denote by E_{CV}^* the optimal control variate estimator with the optimal coefficients β_1^* and β_2^* and $V_{CV}^* \equiv \mathrm{Var}(E_{CV}^*)$ the corresponding minimized variance. Because the optimal control variate estimator is generally guaranteed achieving variance reduction (Owen 2013), the optimal control variate estimator E_{CV}^* has lower variance than the standard Monte Carlo estimator E_{MC} (4):

Corollary 1. *For $\mathbf{c} \neq \mathbf{0}$,*

$$V_{CV}^* = V_{MC}(1 - \rho_1^2 - \frac{1}{2}\rho_2^2), \tag{12}$$

where $V_{MC} = m^{-1}(1 - e^{-\mathbf{c}^\top \Sigma \mathbf{c}})$, $\rho_1^2 = \dfrac{e^{-\mathbf{c}^\top \Sigma \mathbf{c}}}{1 - e^{-\mathbf{c}^\top \Sigma \mathbf{c}}}\mathbf{c}^\top \Sigma \mathbf{c}$, and $\rho_2^2 = \dfrac{e^{-\mathbf{c}^\top \Sigma \mathbf{c}}}{1 - e^{-\mathbf{c}^\top \Sigma \mathbf{c}}}(\mathbf{c}^\top \Sigma \mathbf{c})^2$.

As shown in Corollary 1, the optimal control variate estimator provably has lower variance than the standard Monte Carlo estimator by a factor of $1 - \rho_1^2 - \frac{1}{2}\rho_2^2$. On the other hand, the proposed moment matching estimator E_{MM} (8) can be written in an asymptotic form:

Theorem 2. *For the optimal coefficients β_1^* and β_2^* defined in Theorem 1, the moment matching estimator* (8) *can be rewritten as*

$$E_{MM} = E_{MC} + \beta_1^*(i\mathbf{c}^\top \hat{\boldsymbol{\mu}}) + \frac{1}{2}\beta_2^*(\mathbf{c}^\top \otimes \mathbf{c}^\top)\mathrm{vec}(\hat{\Sigma} - \Sigma) + o_p(m^{-1/2}). \tag{13}$$

Therefore, comparing (13) and (11) implies that the moment matching estimator is asymptotically equivalent to the optimal control variate estimator. The following corollary summarizes the implication:

Corollary 2. *The moment matching estimator* (13) *is asymptotically equivalent to the control variate estimator* (11) *with the optimal coefficients:*

$$E_{MM} = E_{CV}^* + o_p(m^{-1/2}). \tag{14}$$

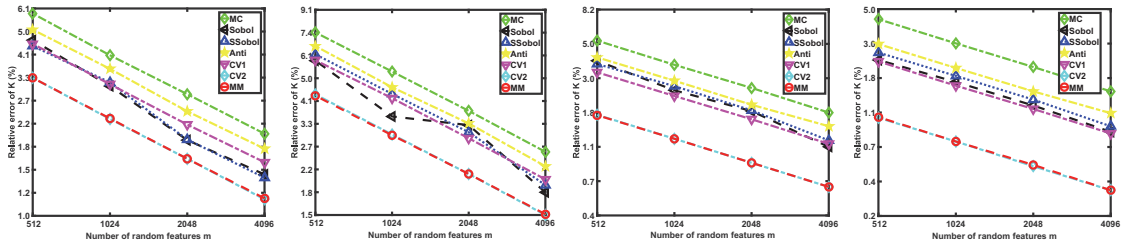
The variance of the moment matching estimator (13) *is asymptotically equivalent to that of the control variate estimator* (11) *with the optimal coefficients:*

$$V_{MM} = V_{CV}^* + o(m^{-1}). \tag{15}$$

With Corollaries 1 and 2, we conclude that the proposed moment matching estimator E_{MM} provably has lower variance than the standard Monte Carlo estimator E_{MC}. The convergence rate $m^{-1/2}$ is independent with the dimension of input data d. As m increases, we will expect the moment matching estimator E_{MM} behaves similarly to the control variate estimator E_{CV}^*, which is confirmed by our numerical results in Section 4 and the long version of this work.

Discussions

In general, the moment matching method achieves the improvement in approximation accuracy by ensuring the generated set of finite random samples exactly match the first



(a) YP90 (b) QM274 (c) MNIST300 (d) LR500

Figure 2: Errors of approximated Gram matrices in the spectral norm by different sample sizes (log-log).

a few moments of the desired probability distribution. It has been empirically shown to be efficacious in reducing sampling variance in various financial and machine learning applications (Glasserman 2003; Gretton et al. 2006; Owen 2013). However, limited theoretical work has existed showing its effectiveness and its asymptotic properties in any integral calculation. Thus, to the best of our knowledge, those theoretical results, especially the linkages between two estimators in Corollaries 1 and 2, have filled some void and enriched the moment matching method on its own.

On the other hand, while other common variance reduction techniques, such as the methods of control variates, have also been widely applied successfully, their approaches of adding extra terms to adjust the finite sum approximation to the integral are incompatible with the inner product framework expressed by (4). In particular, as given in equation (10), although the auxiliary control variate estimator E_{CV} can be used to approximate the kernel integral (3), it cannot trivially produce random feature maps for downstream applications. In contrast, as shown in equation (9), the moment matching estimator E_{MM} is adaptive to the inner product framework without changing the underlying structure.

In addition, the new method is cheap to implement and easy to use. The extra computational cost mainly from the square root decomposition is at the order of $O(d^3)$, which is unaffected by the number of random samples or that of input data. Also, as the Gaussian distribution is completely characterized by its first two moments, matching the higher order moments would generally be unnecessary. Structurally, as the produced new random feature $\mathbf{Q}(\mathbf{x})$ merely replaces the random samples $\{\mathbf{w}_j\}_{j=1}^m$ by $\{\tilde{\mathbf{w}}_j\}_{j=1}^m$, the new method could be easily combined with other techniques thereafter to further enhance the quality of the approximation.

4 Experiments

In this section, we describe the experimental settings, and then demonstrate the performance of the Gram matrix approximation and downstream use in regression.

Data and Settings

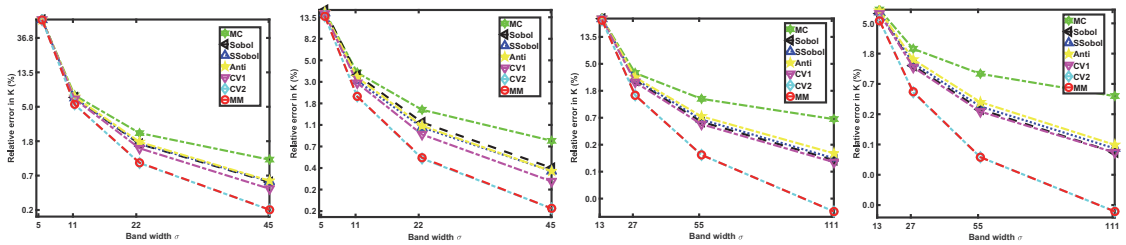
Here we introduce the benchmarks, the competing methods, as well as the evaluation metrics.

Data: Four benchmark datasets with relatively high dimensions are examined in our experiments: (a) YP90 with 10000 and 2000 90-dimensional data points for training and testing, respectively; (b) QM274 with 6000 and 1165 274-dimensional data points for training and testing, respectively; (c) MNIST300 with 8000 and 2000 300-dimensional data points for training and testing, respectively; and (d) LR500 with 8000 and 2000 500-dimensional data points for training and testing, respectively. The first three datasets are from real-world applications and the last one is synthesized by simulation based on the 500-dimensional normal density function with a randomly generated correlation matrix. 1-dimensional outputs for each of the four datasets represent either continuous response values or discrete label information. For the synthetic dataset, the response values are generated through a linear transformation of the input data peppered with random noises.

Baselines: In our comparison study, we consider kernel approximation based on the following seven sampling methods: (i) standard Monte Carlo sampling by pseudo-random numbers (**MC**); (ii) quasi-Monte Carlo sampling by Sobol numbers (**Sobol**); (iii) quasi-Monte Carlo sampling by scrambled Sobol numbers (**SSobol**); (iv) antithetic sampling (**Anti**); (v) control variate sampling with controlling the first moment (**CV1**); (vi) control variate sampling with controlling the first two moments (**CV2**); and (vii) the proposed moment matching sampling method (**MM**). Specifically, **MC** is the original method in (Rahimi and Recht 2007). **Sobol** is a well-tested method in (Yang et al. 2014a). **SSobol** as a randomized variant of **Sobol** often achieves a faster convergence rate (Dick, Kuo, and Sloan 2013). **Anti** represents one commonly adopted variance reduction method with cheap computational costs (Niederreiter 1992). **CV1** and **CV2** that respectively employ the first term and the both terms in (11) represent one of the most effective variance reduction techniques in practice (Owen 2013).

Performance metrics: To fairly appraise the approximation of Gram matrices, we first report the relative errors in both the Frobenius norm as $\|\mathbf{K}-\hat{\mathbf{K}}\|_F/\|\mathbf{K}\|_F$ and the spectral norm as $\|\mathbf{K}-\hat{\mathbf{K}}\|_2/\|\mathbf{K}\|_2$. Second, we report relative errors in l_2-norm as $\|\mathbf{z}-\hat{\mathbf{z}}\|_2/\|\mathbf{z}\|_2$ to evaluate regression tasks, where $\hat{\mathbf{z}}$ is predicted value and $\mathbf{z}$ is the true value. To mitigate the intervention of parameter tuning, we compare the predictive power in a principal component regression setting (Hastie, Tibshirani, and Friedman 2009). We specify the band width σ as the average distance of all data points to their tenth nearest neighbors unless otherwise stated.

To show the statistical significance, for all the methods ex-



(a) Frobenius Norm (b) Spectral Norm (c) Frobenius Norm (d) Spectral Norm

Figure 3: Relative errors of approximated Gram matrices by different band widths across two datasets, i.e., (a) and (b) on QM274; (c) and (d) on LR500 (semilog). The tenth nearest neighbors for QM274 and LR500 are 11 and 27, respectively. The approximated Gram matrices $\hat{\mathbf{K}}$ are constructed based on 2048 samples.

cept the deterministic sampling method (ii), 100 random and independent executions are repeated to compute the average performance for Gram matrix approximation and regression. For clarity, we do not report the observed negligible error bands in plots and figures. Additionally, as the computational costs are negligible when compared with those for any reasonable downstream use, we do not report running times as they are of the same order for the different methods.

Notably, while all the preceding sampling methods can be used to construct Gram matrices, only the methods (i), (ii), (iii) and (vii) can produce random features, thereby being tested in the regression task.

Results

We report the approximation errors of Gram matrices by increasing the number of random features m, the approximation errors of Gram matrices by varying the band width σ, and out-of-sample errors in principal component regression.

By increasing the sampling size m, we demonstrate the convergence curves of Gram matrix approximation for the seven methods across the four datasets in Figure 1 and Figure 2 in the Frobenuis norm and the spectral norm, respectively. First, **MM** constantly achieves the lowest approximation errors, e.g., that **MM** apparently outperforms **MC** and **Sobol**. Second, the observation that **MM** and **CV2** are consistent with each other across all the experiments numerically supports the theoretical conclusion in Corollary 2. In other words, with the current size of the random features **MM** and **CV2** have achieved the asymptotic equivalence. Third, **MM** shows markedly robust variance reduction in high dimensions, from 90 to 500 dimensions. In addition, as observed in (Yang et al. 2014a), **MC** underperforms all the other sampling methods such as **Sobol** and **SSobol**. Further, consistent with Corollary 1, **CV1** and **CV2** both beat **MC**, and with the assist of one extra orthogonal control **CV2** demonstrates an additional improvement over **CV1**.

Figure 3 illustrates the trend of relative approximation errors by increasing the size of the band width σ in the Gaussian kernel across two datasets. From the standpoint of approximating the Gram matrix, the size of the band width determines how challenging it is to achieve an accurate approximation. The performances of all the methods are in line with those in Figures 1 and 2, i.e., that **MM** constantly has lower approximation errors with noticeable effective sizes than others. This observation signifies that the proposed sampling method is sufficiently robust in tuning σ for applications.

Table 1: Out-of-sample regression error (%).

Dataset	PC #	MC	Sobol	SSobol	MM
(a)	4096	0.71	0.71	0.71	**0.70**
	512	0.51	0.51	0.52	**0.50**
(b)	4096	6.27	6.50	6.36	**6.20**
	512	5.30	5.27	5.39	**5.27**
(c)	4096	75.40	75.90	76.50	**75.40**
	512	**81.00**	81.45	81.24	81.23
(d)	4096	30.62	30.07	29.30	**29.30**
	512	24.34	23.63	24.44	**23.00**

Note: We first implement principal component decomposition on 4096 sampled random features. Then, we build principal component regression models by all 4096 principal components and by the first 512 principal components, respectively. Error bands are all negligible after repeating the simulation 100 times.

Finally, following (Yang et al. 2014a), we investigate the generalization errors in principal component regression of the four methods on the four datasets in Table 1. Accordingly, **MM** shows comparable performance without any deterioration and has the statistically lowest out of sample errors in most tests. Such performance gain further confirms the effectiveness of the proposed **MM** sampling method for kernel approximation.

5 Conclusions and Discussions

In this paper, we have presented a novel and easy-to-use algorithm to effectively approximate the Gaussian kernel with random features. The proposed sampling algorithm has taken the advantages of the moment matching technique to achieve higher accuracy. Our theoretical proofs have shown that the new method demands fewer features than the standard Monte Carlo method in any dimensions to achieve the same approximation accuracy. Besides echoing with the theorems, our empirical studies have also demonstrated that the new algorithm has noticeably lessened the approximation errors in Gram matrices while achieving a comparable level of accuracy in regression tasks. Our future work includes ap-

plying the new algorithm to other classes of kernels (Hamid et al. 2014) and combing it with other random feature construction methods (Feng, Hu, and Liao 2015).

References

Aizerman, A.; Braverman, E. M.; and Rozoner, L. I. 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control* 25:821–837.

Argyriou, A.; Micchelli, C. A.; and Pontil, M. 2009. When is there a representer theorem? Vector versus matrix regularizers. *The Journal of Machine Learning Research* 10:2507–2529.

Avron, H., and Sindhwani, V. 2015. High-performance kernel machines with implicit distributed optimization and randomization. *Technometrics* 1–27.

Bach, F. 2015. On the equivalence between quadrature rules and random features. *arXiv preprint arXiv:1502.06800*.

Bengio, Y.; Delalleau, O.; and Roux, N. L. 2005. The curse of highly variable functions for local kernel machines. In *Advances in Neural Information Processing Systems*, 107–114.

Bochner, S. 1933. Monotone funktionen, Stieltjessche integrale und harmonische analyse. *Mathematische Annalen* 108(1):378–410.

Caflisch, R. E. 1998. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica* 7:1–49.

Chen, X.; Yang, H.; King, I.; and Lyu, M. R. 2015. Training-efficient feature map for shift-invariant kernels. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.

Dai, B.; Xie, B.; He, N.; Liang, Y.; Raj, A.; Balcan, M. F.; and Song, L. 2014. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, 3041–3049.

Dick, J.; Kuo, F. Y.; and Sloan, I. H. 2013. High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica* 22:133–288.

Feng, C.; Hu, Q.; and Liao, S. 2015. Random feature mapping with signed circulant matrix projection. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 3490–3496.

Glasserman, P. 2003. *Monte Carlo Methods in Financial Engineering*, volume 53. Springer.

Gretton, A.; Borgwardt, K. M.; Rasch, M.; Schölkopf, B.; and Smola, A. J. 2006. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, 513–520.

Hamid, R.; Xiao, Y.; Gittens, A.; and DeCoste, D. 2014. Compact random feature maps. In *Proceedings of the 31st International Conference on Machine Learning*, 19–27.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning*. Springer.

Huang, P.-S.; Avron, H.; Sainath, T. N.; Sindhwani, V.; and Ramabhadran, B. 2014. Kernel methods match deep neural networks on timit. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 205–209.

Kar, P., and Karnick, H. 2012. Random feature maps for dot product kernels. *The Journal of Machine Learning Research* 22:583–591.

Le, Q.; Sarlós, T.; and Smola, A. 2013. Fastfood-Approximating kernel expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning*, 244–252.

Li, F.; Ionescu, C.; and Sminchisescu, C. 2010. Random Fourier approximations for skewed multiplicative histogram kernels. In *Pattern Recognition*. Springer. 262–271.

Lopez-Paz, D.; Sra, S.; Smola, A.; Ghahramani, Z.; and Schölkopf, B. 2014. Randomized nonlinear component analysis. In *Proceedings of the 31st International Conference on Machine Learning*, 1359–1367.

Maji, S., and Berg, A. C. 2009. Max-margin additive classifiers for detection. In *IEEE 12th International Conference on Computer Vision*, 40–47. IEEE.

Niederreiter, H. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63. SIAM.

Owen, A. B. 2013. Monte Carlo Theory, Methods and Examples. Working Book, Stanford University.

Pennington, J.; Yu, F.; and Kumar, S. 2015. Spherical random features for polynomial kernels. In *Advances in Neural Information Processing Systems*, 1846–1854.

Pham, N., and Pagh, R. 2013. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*, 239–247. ACM.

Rahimi, A., and Recht, B. 2007. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 1177–1184.

Schölkopf, B., and Smola, A. J. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

Sutherland, D. J., and Schneider, J. 2015. On the error of random Fourier features. In *Conference on Uncertainty in Artificial Intelligence*, 862–871.

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

Yang, T.; Li, Y.-F.; Mahdavi, M.; Jin, R.; and Zhou, Z.-H. 2012. Nyström method vs random Fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems*, 476–484.

Yang, J.; Sindhwani, V.; Avron, H.; and Mahoney, M. 2014a. Quasi-Monte Carlo feature maps for shift-invariant kernels. In *Proceedings of the 31st International Conference on Machine Learning*, 485–493.

Yang, J.; Sindhwani, V.; Fan, Q.; Avron, H.; and Mahoney, M. W. 2014b. Random Laplace feature maps for semigroup kernels on histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 971–978.

Yen, I. E.-H.; Lin, T.-W.; Lin, S.-D.; Ravikumar, P. K.; and Dhillon, I. S. 2014. Sparse random feature algorithm as coordinate descent in Hilbert space. In *Advances in Neural Information Processing Systems*, 2456–2464.