Generalization Guarantees for Sparse Kernel Approximation with Entropic Optimal Features

Liang Ding [1] **Rui Tuo** [1] **Shahin Shahrampour** [1]

Abstract

Despite their success, kernel methods suffer from a massive computational cost in practice. In this paper, in lieu of commonly used kernel expansion with respect to N inputs, we develop a novel optimal design maximizing the entropy among kernel features. This procedure results in a kernel expansion with respect to entropic optimal features (EOF), improving the data representation dramatically due to features dissimilarity. Under mild technical assumptions, our generalization bound shows that with only $\mathcal{O}(N^{\frac{1}{4}})$ features (disregarding logarithmic factors), we can achieve the optimal statistical accuracy (i.e., $\mathcal{O}(1/\sqrt{N})$). The salient feature of our design is its sparsity that significantly reduces the time and space costs. Our numerical experiments on benchmark datasets verify the superiority of EOF over the state-of-the-art in kernel approximation.

1. Introduction

Kernel methods are powerful tools in describing nonlinear data models. However, despite their success in various machine learning tasks, kernel methods always suffer from scalability issues, especially when the learning task involves matrix inversion (e.g., kernel ridge regression). This is simply due to the fact that for a dataset of size N, the inversion step requires $\mathcal{O}(N^3)$ time cost. To tackle this problem, a great deal of research has been dedicated to the approximation of kernels using low-rank surrogates (Smola & Schökopf, 2000; Fine & Scheinberg, 2001; Rahimi & Recht, 2008). By approximating the kernel, these methods deal with a linear problem, potentially solvable in a linear time with respect to N (see e.g. (Joachims, 2006) for linear Support Vector Machines (SVM)).

In the approximation of kernel with a finite number of features, one fundamental question is how to select the features. As an example, in supervised learning, we are interested in identifying features that lead to low out-of-sample error. This question has been studied in the context of random features, which is an elegant method for kernel approximation (Rahimi & Recht, 2008). Most of the works in this area improve the out-of-sample performance by modifying the stochastic oracle from which random features are sampled (Sinha & Duchi, 2016; Avron et al., 2017; Shahrampour et al., 2018). Nevertheless, these methods deal with dense feature matrices (due to randomness) and still require a large number of features to learn the data subspace. Decreasing the number of features directly affects the time and space costs, and to achieve that, we must choose features that are as distinct as possible (to better span the space). Focusing on explicit features, we aim to achieve this goal in the current work.

1.1. Our Contributions

In this paper, we study low-rank kernel approximation by finding a set of mutually orthogonal features with nested and compact supports. We first theoretically characterize a condition (based on the Sturm-Liouville problem), which allows us to obtain such features. Then, we propose a novel optimal design method that maximizes the metric entropy among those features. The problem is formulated as a combinatorial optimization with a constraint on the number of features used for approximation. The optimization is generally NP-hard but yields closed-form solutions for specific numbers of features. The algorithm, dubbed entropic optimal features (EOF), can use these features for supervised learning. The construction properties of features (orthogonality, compact support, and nested support) result in a sparse approximation saving dramatically on time and space costs. We establish a generalization bound for EOF that shows with only $\mathcal{O}(N^{\frac{1}{4}})$ features (disregarding logarithmic factors), we can achieve the optimal statistical accuracy (i.e., $\mathcal{O}(1/\sqrt{N})$). Our numerical experiments on benchmark datasets verify the superiority of EOF over the state-of-the-art in kernel approximation. While we postpone the

[1]The authors are with Wm Michael Barnes '64 Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, USA. Correspondence to: Shahin Shahrampour <shahin@tamu.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

exhaustive literature review to Section 6, none of the previous works has approached the problem from the entropy maximization perspective, which is the unique distinction of the current work.

2. Preliminaries on Kernel Methods

Kernel methods map finite-dimensional data to a potentially infinite dimensional feature space. Any element f in the reproducing kernel Hilbert space (RKHS) of k, denoted by $\mathcal{H}_k$, has the following representation:

$$f = \sum_{i=1}^{\infty} \langle f, g_i \rangle_k g_i, \tag{1}$$

where $\langle \cdot, \cdot \rangle_k$ denotes the RKHS inner product induced by k, and $\{g_i\}$ is any orthonormal feature set (i.e., orthonormal basis functions) that spans the space $\mathcal{H}_k$. In general, the kernel trick relies on the observation that the inner product $\langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{x}') \rangle_k = k(\mathbf{x}, \mathbf{x}')$ for $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^D$ (reproducing property), so $k(\mathbf{x}, \mathbf{x}')$ is cheap to compute without the need to calculate the inner product.

When we are concerned with supervised learning, under mild conditions, by the Representer Theorem, it is guaranteed that any solution of the risk minimization problem assumes the form $f(\cdot) = \sum_{i=1}^{N} c_i k(\cdot, \mathbf{x}_i)$, where N is the number of training data points. However, this representation introduces a massive time cost of $\mathcal{O}(N^3)$ and a memory cost of $\mathcal{O}(N^2)$ in the training. Furthermore, the feature space $\{k(\cdot, \mathbf{x}) : \mathbf{x} \in \mathbb{R}^D\}$ may not cover $\mathcal{H}_k$ in an optimal sense. To be more specific, there might be another set of features $\{g_i\}_{i=1}^{M}$ with $M \ll N$ such that $\{k(\cdot, \mathbf{x}) : \mathbf{x} \in \mathbf{X}\} \subset \{g_i\}_{i=1}^{M}$ where $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the input data.

To address the aforementioned problem, (Rahimi & Recht, 2008) propose a random approximation of $k(\mathbf{x}, \mathbf{x}')$

$$k(\mathbf{x}, \mathbf{x}') \approx z^{\mathsf{T}}(\mathbf{x}) z(\mathbf{x}'), \tag{2}$$

where $z^{\mathsf{T}}(\mathbf{x}) = [\zeta_1(\mathbf{x}), \ldots, \zeta_M(\mathbf{x})]$ is a random vector. This decomposes the feature $k(\cdot, \mathbf{x})$ into a linear combination of random low-rank features $\{\zeta_i\}$ to approximate the original target function $\sum_{i=1}^{N} c_i k(\cdot, \mathbf{x}_i)$ by $\sum_{i=1}^{M} \alpha_i \zeta_i$, where $\{\alpha_i\}_{i=1}^{M}$ must be learned (calculated) by data. This idea resolves the computational issue, but due to random selection of the features, the method does not offer the best candidate features for reconstructing the target function.

Furthermore, in supervised learning the goal is to find a mapping from inputs to outputs, and thus, an optimal kernel approximation does not necessarily result in an optimal target function representation. The reason is simply that we require the features that best represent the underlying data model (or target function) rather than the kernel function.

3. Kernel Feature Selection

In this paper, we propose an algorithm that uses a sparse representation to attain a high prediction accuracy with a low computational cost. The key is to find an expansion

$$f = \sum_{i=1}^{\infty} \langle f, g_i \rangle_k g_i, \tag{3}$$

such that features $\{g_i\}$ satisfy the following properties:

1. Compact support: $\mathrm{supt}[g_i]$ is compact.
2. Nested support: $\mathrm{supt}[g_i] = \bigcup_{j \in \mathrm{I}} \mathrm{supt}[g_j]$ for some finite set I.
3. Orthonormality: $\langle g_i, g_j \rangle_k = \delta_{ij}$, where δ_{ij} denotes the Kronecker delta.

Properties 1-2 ensure low time cost for the algorithm by promoting sparsity. To be more specific, given any finite set $\{g_i\}_{i=1}^{M}$ and any data point $\mathbf{x}$, $g_i(\mathbf{x}) = 0$ for a large number of basis functions in $\{g_i\}_{i=1}^{M}$. Property 3 provides a better expansion of $\mathcal{H}_k$.

In general, this problem may be intractable; however, we will prove later in Theorem 2 that when k satisfies the following condition, then a feature set $\{\phi_i\}$ that satisfies properties 1-3 does exist.

Condition 1. *Let kernel k be of the following product form:*

$$k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} p(\min\{x_d, x_d'\}) q(\max\{x_d, x_d'\})$$

where p and q are the independent solutions of the Sturm-Liouville problem on the interval $[a, b]$ for any $a, b \in [-\infty, \infty]$:

$$\frac{d}{dx}\alpha(x)\frac{dy}{dx} + \beta(x)y = 0,$$

and they satisfy the following boundary conditions:

$$c_{11}p'(a) + c_{12}p(a) = 0$$
$$c_{21}q'(b) + c_{22}q(b) = 0$$

with $c_{ij} \geq 0$ for $i, j = 1, 2$ and the operator $\frac{d}{dx}\alpha(x)\frac{d}{dx} + \beta(x)$ is an elliptic operator that satisfies Lax-Milgram Theorem (see Section 6 of (Evans, 2010)).

We provide two commonly used kernels that satisfy condition 1:

$$k(\mathbf{x}, \mathbf{x}') = e^{-\omega \|\mathbf{x} - \mathbf{x}'\|_1}$$
$$k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} [\omega \min\{x_d, x_d'\} + 1].$$

The first one is the Laplace kernel and the second one is the kernel associated to weighted Sobolev space (Dick et al., 2013). Let $z_{l,i} = i2^{-l}$ for any $l, i \in \mathbb{N}$. Then, when the dimension $D = 1$, features associated to Laplace kernel satisfying properties 1-3 are as follows:

$$\phi_{l,i}(x) = \begin{cases} \frac{\sinh \omega |x - z_{l,i+1}|}{\sinh \omega 2^{-l}} & \text{if } x \in (z_{l,i}, z_{l,i+1}] \\ \frac{\sinh \omega |x - z_{l,i-1}|}{\sinh \omega 2^{-l}} & \text{if } x \in [z_{l,i-1}, z_{l,i}] \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and features associated to the weighted Sobolev space kernel are as follows:

$$\phi_{l,i}(x) = \max\left\{0, 1 - \frac{|x - z_{l,i}|}{2^{-l}}\right\}$$

where (l, i) is the index of features. We now start from 1-D kernel to construct a feature space that satisfies properties 1-3:

Theorem 1. *Suppose that k is a kernel that satisfies Condition 1. Let $\mathbf{Z}_l = \{z_{l,i} = i2^{-l} : i = 1, \ldots, 2^l - 1\}$ and let $B_l = \{i = 1, \ldots, 2^l - 1 : i \text{ is odd}\}$. We then define the following function on the interval $[z_{l,i-1}, z_{l,i+1}] = [(i-1)2^{-l}, (i+1)2^{-l}]$:*

$$\phi_{l,i}(x) = \begin{cases} \frac{q(x)p_{l,i+1} - p(x)q_{l,i+1}}{q_{l,i}p_{l,i+1} - p_{l,i}q_{l,i+1}} & \text{if} \quad x \in (z_{l,i},\ z_{l,i+1}] \\ \frac{p(x)q_{l,i-1} - q(x)p_{l,i-1}}{p_{l,i}q_{l,i-1} - q_{l,i}p_{l,i-1}} & \text{if} \quad x \in [z_{l,i-1},\ z_{l,i}] \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

where

$$p_{l,i} = p(z_{l,i}) = p(i2^{-l})$$
$$q_{l,i} = q(z_{l,i}) = q(i2^{-l}).$$

Then, the following feature set is an orthogonal basis of the RKHS of k, $\mathcal{H}_k$, that satisfies property 1-3 on the unit interval $[0, 1]$:

$$\{\phi_{l,i} : l \in \mathbb{N}, i \in B_l\}.$$

The theorem above characterizes the set of features that satisfy Condition 1 when the input is scalar. To extend the idea to D-dimensional space, we only need to take the tensor product form of the 1-dimensional kernel, as described by the consequent theorem:

Theorem 2. *Suppose that k is a kernel that satisfies Condition 1. For any $\mathbf{l} \in \mathbb{N}^D$, we define the Cartesian product of sets as follows:*

$$\mathbf{Z}_\mathbf{l} = \otimes_{d=1}^D \mathbf{Z}_{l_d} = \{\mathbf{z}_{\mathbf{l},\mathbf{i}} = (z_{l_1,i_1}, \cdots, z_{l_D,i_D}) : z_{l_d,i_d} \in \mathbf{Z}_{l_d}\}$$
$$B_\mathbf{l} = \otimes_{d=1}^D B_{l_d} = \{\mathbf{i} \in \mathbb{N}^D : i_d \in B_{l_d}\}.$$

We then define the following function on the hypercube $\otimes_{d=1}^D [z_{l_d,i_d-1}, z_{l_d,i_d+1}] = \otimes_{d=1}^D [(i_d - 1)2^{-l_d}, (i_d + 1)2^{-l_d}]$:

$$\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) = \prod_{d=1}^D \phi_{l_d,i_d}(x_d), \tag{6}$$

where the function ϕ_{l_d,i_d} is defined in Theorem 1. Then the following feature set is an orthogonal basis of the RKHS of k, $\mathcal{H}_k$, that satisfies property 1-3 on the unit cube $[0, 1]^D$:

$$\{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{l} \in \mathbb{N}^D, \mathbf{i} \in B_\mathbf{l}\}.$$

The proof of Theorem 1 is given in the supplementary material. Theorem 2 can be derived from Theorem 1, because the kernel is simply the tensor product of the 1-dimensional kernel in Theorem 1.

Corollary 1. *For any kernel k that satisfies Condition 1, let $\phi_{\mathbf{l},\mathbf{i}}$ be the feature defined in Theorem 2. Then, we have the following expansion for k:*

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{l} \in \mathbb{N}^D} \sum_{\mathbf{i} \in B_\mathbf{l}} \frac{\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}')}{\langle \phi_{\mathbf{l},\mathbf{i}}, \phi_{\mathbf{l},\mathbf{i}} \rangle_k} \tag{7}$$

where $\langle \cdot, \cdot \rangle_k$ is the inner product induced by k.

Proof. We only need to substitute $f(\cdot)$ in equation (3) by $k(\mathbf{x}, \cdot)$, then according to the reproducing property of k we can have the result. □

Corollary 1 is the direct result of Theorem 2. So we can have the following sparse approximation for the kernel function $k(\mathbf{x}, \mathbf{x}')$:

$$k(\mathbf{x}, \mathbf{x}') \approx z^\mathsf{T}(\mathbf{x}) z(\mathbf{x}'),$$

where

$$z(\mathbf{x}) = \left[\frac{\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})}{||\phi_{\mathbf{l},\mathbf{i}}||_k}\right]_{(\mathbf{l},\mathbf{i}) \in S},$$

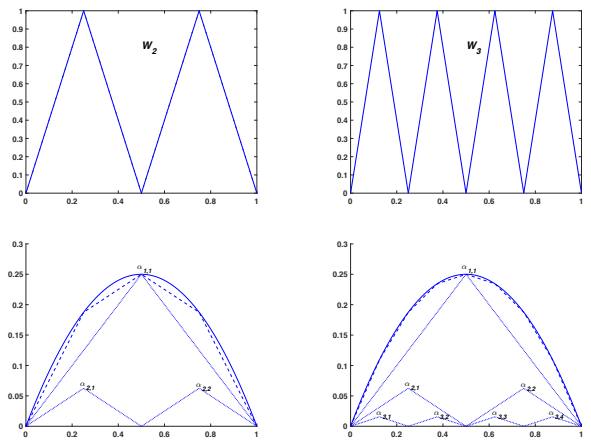
for some set S. Thanks to properties 1-3 (including compact and nested supports for features), the computational advantage of the above approximation is that most entries of $z(\mathbf{x})$ are zero. Therefore, the vector is $z(\mathbf{x})$ is *sparse*, leading to sparse feature matrices.

Remark 1. *Kernel approximation schemes such as random features (Rahimi & Recht, 2008) or Nyström method (Williams & Seeger, 2001; Drineas & Mahoney, 2005) work for a broader class of kernels, but they do not result in sparse approximation. The sparse feature matrices obtained under Condition 1 can speed up the training in supervised learning, as we will see in the numerical experiments (Section 7).*

We now use the RKHS of the following kernel on $[0, 1]$ as an example:

$$k(x, x') = \min\{x, x'\}[1 - \max\{x, x'\}].$$

Figure 1. Top two panels: $\mathbf{W}_2 = \{\phi_{l,i} : l = 2\}$ and $\mathbf{W}_3 = \{\phi_{l,i} : l = 3\}$; lower two panels: nested structure for the representation of a function $f \in \mathcal{H}_k$.



The RKHS associated to k is the first order Sobolev space with zero boundary conditions:

$$\mathcal{H}_k = \left\{ f : \int_0^1 [f'(s)]^2 ds < \infty, f(0) = f(1) = 0 \right\}.$$

In this example, the feature functions given by Theorem 1 coincide with a wavelet basis in $\mathcal{H}_k$. Consider the mother wavelet given by the triangular function:

$$\phi(d) = \max\{0, 1 - |d|\}.$$

Then for any $l \in \mathbb{N}$, $i = 1, \cdots, 2^l - 1$, direct calculations show that

$$\phi_{l,i}(x) = \phi\left(\frac{x - i2^{-l}}{2^{-l}}\right). \tag{8}$$

Now it is easy to verify that the features $\{\phi_{l,i} : l \in \mathbb{N}, i \text{ is odd}\}$ satisfy the desired properties 1-3. Specifically,

1. $\text{supt}[\phi_{l,i}] = [(i-1)2^{-l}, (i+1)2^{-l}]$.
2. $\text{supt}[\phi_{l,i}] = \text{supt}[\phi_{l+1,2i-1}] \cup \text{supt}[\phi_{l+1,2i+1}]$.
3. $\int_0^1 \phi'_{l,i}\phi'_{n,j} ds = 2^{l+1}\delta_{(l,i),(n,j)}$, where $\delta_{(l,i),(n,j)} = 1$ when $(l,i) = (n,j)$ and zero otherwise.

Figure 1 illustrates the compact and nested supports of these wavelet features. The compact support properties can lead to a significant improvement in the time cost. Consider the evaluation of $f(x) = \sum_{|l| \le n} \alpha_{l,i}\phi_{l,i}(x)$. The compact support property implies that $\phi_{l,i}(x) = 0$ for most (l,i)'s, so the computational cost of evaluating $f(x)$ can be much lower than the total number of features. In Section 4.1, we will leverage this property of the basis functions to propose an efficient algorithm for learning. This goal cannot be achieve when the basis functions are not compactly supported.

Figure 2. 2-D tensor product of wavelet features with compact support $\phi_{[1,2],[11]}$ and $\phi_{[1,2],[13]}$

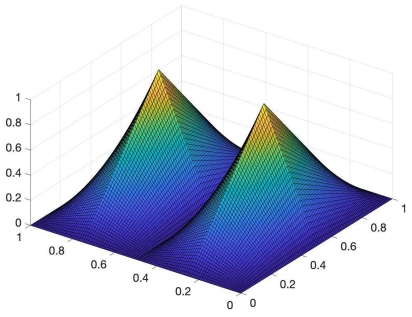


Figure 2 shows the example of the tensor product of the wavelet feature defined in (8). It is a 2-dimensional extension of the wavelet feature. For the general D-dimensional case, according to Theorem 2, the features satisfy properties 1-3 in the RKHS induced by the following kernel:

$$k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} \min\{x_d, x'_d\}[1 - \max\{x_d, x'_d\}],$$

which is the mixed Sobolev space of first order with zero boundary condition on $[0,1]^D$.We refer the reader to (Bungartz & Griebel, 2004) for more details on mixed order Sobolev space.

In view of Theorem 2, we can lift a data point from $\mathbf{x} \in \mathbb{R}^D$ to a finite dimensional space spanned by features with compact and nested supports. As a result, the evaluation of $\mathbf{x}$ on a large number of features is zero, yielding a sparse and efficient representation.

4. Entropic Optimal Design

In the previous section, we provided conditions under which we can find features with compact and nested supports. We now present an optimization criterion to select the best finite set of features with the maximum metric entropy. The intuition behind this choice is that we favor a set of features that are different from each other as much as possible, so that we can reconstruct the underlying model by a moderate amount of features.

To formulate the optimization problem, we need to introduce some notation. First, we introduce the covering number of an operator between two Banach spaces. Let $\varepsilon > 0$ and A, B be Banach spaces with unit balls B_{A} and B_{B}, respectively. The covering number of a bounded linear operator $T : \mathsf{A} \to$

B is defined as

$$\mathcal{N}(T,\varepsilon) := \inf_{n\in\mathbb{N}}\left\{n : \exists\{b_i\in\mathsf{B}\}_{i=1}^n \text{ s.t. } T(B_\mathsf{A})\subseteq\bigcup_{i=1}^n(b_i+\varepsilon B_\mathsf{B})\right\},$$

where $b_i+\varepsilon B_\mathsf{B}$ is a ball of radius ε, centered at b_i. The metric entropy of T is then defined as $\mathsf{Ent}[T,\varepsilon] := \log\mathcal{N}(T,\varepsilon)$. Now, let $\mathcal{H}_k$ be the RKHS associated to kernel k with the inner product $\langle\cdot,\cdot\rangle_k$, and let $\mathcal{P}_S$ be the projection operator from $\mathcal{H}_k$ to the following finite dimensional subspace

$$\mathcal{F}_S = \{\phi_{\mathbf{l},\mathbf{i}} : (\mathbf{l},\mathbf{i})\in S\},$$

where $\phi_{\mathbf{l},\mathbf{i}}$ is defined in Theorem 2 and $\dim(\mathcal{P}_S)=\mathrm{card}(S)$. Our goal is to find the optimal set S^* (with cardinality at most M), whose corresponding feature set maximizes the entropy. This is equivalent to solving the following optimization problem:

$$\sup_S \mathsf{Ent}[\mathcal{P}_S,\varepsilon] \quad \text{s.t.} \quad \mathrm{card}(S)\le M. \tag{9}$$

One can show that the features in $\mathcal{F}_S$ are mutually orthogonal with Hilbert norm:

$$||\phi_{\mathbf{l},\mathbf{i}}||_k =: C_{\mathbf{l},\mathbf{i}}^{-1}, \tag{10}$$

where $C_{\mathbf{l},\mathbf{i}}\to 0$ as $|\mathbf{l}|\to\infty$ (see Lemma 1 in the Supplementary Material). We first multiply $\phi_{\mathbf{l},\mathbf{i}}$ by $C_{\mathbf{l},\mathbf{i}}$ to normalize the feature. Then, for any function $f\in\mathcal{H}_k$, we have that

$$\mathcal{P}_S f = \sum_{(\mathbf{l},\mathbf{i})\in S} C_{\mathbf{l},\mathbf{i}}^2\langle f,\phi_{\mathbf{l},\mathbf{i}}\rangle_k\phi_{\mathbf{l},\mathbf{i}}.$$

As a result, the entropic optimization problem (9) is equivalent to searching an M-dimensional Euclidean space with the largest unit ball, which can be characterized as follows

$$\max_S \sum_{(\mathbf{l},\mathbf{i})\in S} C_{\mathbf{l},\mathbf{i}} \quad \text{s.t. } \mathrm{card}(S)\le M.$$

This optimization problem is called the Knapsack problem and, in general, is NP-hard (Kellerer et al., 2004). However, for some specific values of M, closed form solutions exist. Consider the Laplace kernel here as an example. For Laplace kernel $k(\mathbf{x},\mathbf{x}')=e^{-\omega\|\mathbf{x}-\mathbf{y}\|_1}$, from direct calculation, the constant is:

$$C_{\mathbf{l},\mathbf{i}} = \prod_{d=1}^D\sqrt{\sinh(\omega 2^{-l_d})}.$$

In this case, $C_\mathbf{l}=C_{\mathbf{l},\mathbf{i}}$ is independent of $\mathbf{i}$ and for any $|\mathbf{l}|<|\mathbf{l}'|$, the value $C_\mathbf{l}>C_{\mathbf{l}'}$. Therefore, we can derive that when $M=\mathrm{card}\,(\{\mathbf{l}:|\mathbf{l}|\le n\})$ for some n, the optimal set S_n^* is

$$S_n^* = \{(\mathbf{l},\mathbf{i}) : |\mathbf{l}|\le n, \mathbf{i}\in B_\mathbf{l}\} \tag{11}$$

because for any $C_\mathbf{l}\in S_n^*$ and any $C_{\mathbf{l}'}\notin S_n^*$, $C_\mathbf{l}>C_{\mathbf{l}'}$. It turns out the set S_n^* is equivalent to the Sparse Grid design (Bungartz & Griebel, 2004).

4.1. Algorithm: Entropic Optimal Features

Suppose that the set S_n^* given by equation (11) is the index set associated to the feature set that maximizes the metric entropy optimization problem (9). Then, given a specific input $\mathbf{x}$, we can compute the new feature vector

$$z(\mathbf{x}) = [C_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})]_{(\mathbf{l},\mathbf{i})\in S_n^*} =: [z_{\mathbf{l},\mathbf{i}}(\mathbf{x})]_{(\mathbf{l},\mathbf{i})\in S_n^*}$$

where $C_{\mathbf{l},\mathbf{i}}$ is the coefficient defined in (10), and $z(\mathbf{x})$ satisfies

$$k(\mathbf{x},\mathbf{x}')\approx z(\mathbf{x})^\mathsf{T} z(\mathbf{x}'),$$

in Corollary 1 with $\phi_{\mathbf{l},\mathbf{i}}$ the feature function defined in equation (6). We call $z(\mathbf{x})$ the entropic optimal feature (EOF).

According to properties 1-3, the supports of $\{\phi_{\mathbf{l},\mathbf{i}} : (\mathbf{l},\mathbf{i})\in S_n^*\}$ are either disjoint or nested. Therefore, only a small amount of entries on $z(\mathbf{x})$ are non-zero. To be more specific, given any $\mathbf{l}\in\mathbb{N}^D$ and input $\mathbf{x}$, the supports of $\{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{i}\in B_\mathbf{l}\}$ are disjoint so we can immediately compute the unique non-zero entries of $z_{\mathbf{l},\mathbf{i}}(\mathbf{x})$ (recall the 1-dimensional illustration of disjoint supports in Fig. 1).

Algorithm 1 shows how to explicitly compute the EOF $z(\mathbf{x})$ at a data point $\mathbf{x}$. Note that $\lceil\cdot\rceil,\lfloor\cdot\rfloor$ denote the ceiling and floor operations, respectively.

Algorithm 1 Entropic Optimal Features (EOF)

Input: point $\mathbf{x}$, S_n^*
Initialize $z(\mathbf{x}) = [z_{\mathbf{l},\mathbf{i}}(\mathbf{x})]_{(\mathbf{l},\mathbf{i})\in S_n^*} = 0$
while $|\mathbf{l}|\le n+D-1$ **do**
  for $d=1$ **to** D **do**

$$i_d = \begin{cases}\lceil\frac{x_d}{2^{-l_d}}\rceil \text{ if } \lceil\frac{x_d}{2^{-l_d}}\rceil \text{ is odd}\\ \lfloor\frac{x_d}{2^{-l_d}}\rfloor \text{ if } \lfloor\frac{x_d}{2^{-l_d}}\rfloor \text{ is odd}\end{cases}$$

  end for
  $z_{\mathbf{l},\mathbf{i}}(\mathbf{x}) = C_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})$
end while

The dimension of the vector $z(\mathbf{x})$ given n levels is $\mathcal{O}(2^n n^{D-1})$ (Bungartz & Griebel, 2004). The number of

non-zero elements for $z(\mathbf{x})$ after running Algorithm 1 is:

$$\begin{aligned}\sum_{|\mathbf{l}|\leq n+D-1} 1 &= \sum_{i=D}^{n+D-1} \sum_{|\mathbf{l}|=i} 1 \\ &= \sum_{i=D}^{n+D-1} \binom{i-1}{D-1} \\ &= \binom{n+D-1}{D} = \mathcal{O}(n^D),\end{aligned}$$

which means fraction of non-zeros to the whole vector in $z(\mathbf{x})$ grows with $\mathcal{O}(\frac{n}{2^n})$ as a function of level n.

4.2. Time Complexity of EOF in Regression

Based on above, if we fix M as the size of $z(\mathbf{x})$, the number of non-zero entries on $z(\mathbf{x})$ is $\mathcal{O}(\log^D M)$. Since we evaluate $z(\mathbf{x})$ for each training data, the feature matrix has $\mathcal{O}(N \log^D M)$ non-zero elements, resulting in a training cost of $\mathcal{O}(N \log^{2D} M)$, which is smaller than $\mathcal{O}(NM^2)$ of random features (Rahimi & Recht, 2009), especially when D is moderate. Notice that in the aforementioned time costs, we implicitly assumed $M < N$, where N is the number of training samples. If $M > N$, then approximation of kernel – no matter with what technique – does not lead to any computational advantage.

5. Generalization Bound

In this section, we present the generalization bound for EOF when it is used in supervised learning. Let us define the approximated target function as

$$\hat{f} := \operatorname*{argmin}_{f \in \mathcal{F}_M} \frac{1}{N} \sum_{j=1}^{N} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_k^2,$$

given independent and identically distributed samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathcal{F}_M$ denotes the space spanned by the first M EOFs, L is a loss function, and λ is a tuning parameter that may depend on n. We denote by $R(f) := \mathbb{E}_{\mathbf{x},y}[L(y, f(\mathbf{x}))]$ the true risk. The goal is to bound the generalization error $R(\hat{f}) - \inf_{f \in \mathcal{H}_k} R(f)$.

We use the following assumptions to establish the bound:

Assumption 1. *There exists $f_0 \in \mathcal{H}_k$ so that $\inf_{f \in \mathcal{H}_k} R(f) = R(f_0)$.*

Assumption 2. *The function $m_y(\cdot) := L(y, \cdot)$ is twice differentiable for all y. Furthermore, $m_y(\cdot)$ is strongly convex.*

Assumption 3. *The density function of input $\mathbf{x}$ is uniformly bounded away from infinity. The outputs are uniformly bounded.*

Assumption 1 allows infimum to be achieved in the RKHS. This is not ensured automatically since we deal with a potentially infinite-dimensional RKHS $\mathcal{H}_k$, that is possibly universal (see Remark 2 of (Rudi & Rosasco, 2017)). Assumption 2 is true for common loss functions including least squares for regression ($m_y(y') = (y - y')^2$) and logistic regression for classification ($m_y(y') = \log[1 + \exp(-yy')]$). The bounded output constraint of Assumption 3 is also common in supervised learning.

The generalization bound is given by the following theorem.

Theorem 3. *Suppose Assumptions 1-3 are fulfilled. If the tuning parameter is chosen as $\lambda \sim N^{-1/2}$, then*

$$R(\hat{f}) - \inf_f R(f) \leq \mathcal{O}_p(N^{-1/2}) + CM^{-2} \log^{4D-4} M,$$

for some $C > 0$. The constants may depend on $\|f_0\|_k$.

The theorem above shows that with $\mathcal{O}(N^{\frac{1}{4}})$ EOFs, the optimal statistical accuracy $\mathcal{O}(1/\sqrt{N})$ is achieved up to logarithmic factors. Note that for random features, the number of required features to achieve the optimal rate is $\mathcal{O}(\sqrt{N})$ in the case of ridge regression (Rudi & Rosasco, 2017). So EOF improves the generalization bound in the sense of reducing the number of required features to achieve the optimal accuracy. The bound also holds for strongly convex losses, which can potentially include classification using logistic regression.

6. Related Literature

We provide related works for kernel approximation from different perspectives:

Random Features (Randomized Kernel Approximation): Randomized features was introduced as an elegant approach for Monte Carlo approximation of shift-invariant kernels (Rahimi & Recht, 2008), and it was later extended for Quasi Monte Carlo approximation (Yang et al., 2014). Several methods consider improving the time cost of random features, decreasing it by a linear factor of the input dimension (see e.g., Fast-food (Le et al., 2013; Yang et al., 2015)). Quadrature-based random features are also shown to boost kernel approximation (Munkhoeva et al., 2018). The generalization properties of random features have been studied for ℓ_1-regularized risk minimization (Yen et al., 2014) and ridge regression (Rudi & Rosasco, 2017), improving the initial generalization bound of (Rahimi & Recht, 2009). (Felix et al., 2016) develop orthogonal random features (ORF) to boost the variance of kernel approximation. ORF is shown to provide optimal kernel estimator in terms of mean-squared error (Choromanski et al., 2018). A number of recent works have considered data-dependent sampling of random features to improve kernel approximation. Examples consist of (Yu et al., 2015) on compact nonlinear feature maps, (Yang et al., 2015; Oliva et al., 2016) on approximation of shift-invariant/translation-invariant kernels, and (Agrawal et al., 2019) on data-dependent approximation using greedy approaches (e.g., Frank-Wolfe). Data-dependent

sampling has also been used to improve generalization in supervised learning (Sinha & Duchi, 2016; Shahrampour et al., 2018) through target kernel alignment. Furthermore, (Wang & Shahrampour, 2019) propose task-dependent sampling for trace optimization problems, dimensionality reduction, and correlation analysis.

Deterministic Kernel Approximation: The studies on finding low-rank surrogates for kernels date back two decades (Smola & Schökopf, 2000; Fine & Scheinberg, 2001). As an example, the celebrated Nyström method (Williams & Seeger, 2001; Drineas & Mahoney, 2005) samples a subset of training data for approximating a low-rank kernel matrix. The Nyström method has been further improved in (Zhang et al., 2008) and more recently used for approximation of indefinite kernels (Oglic & Gärtner, 2019). Explicit feature maps have also proved to provide efficient kernel approximation. The works of (Yang et al., 2004; Xu et al., 2006; Cotter et al., 2011) have proposed low-dimensional Taylor expansions of Gaussian kernel for improving the time cost of learning. (Vedaldi & Zisserman, 2012) further study explicit feature maps for additive homogeneous kernels.

Sparse Approximation Using Greedy Methods: Sparse approximation literature has mostly focused on greedy methods. (Vincent & Bengio, 2002) have developed a matching pursuit algorithm where kernels are the dictionary elements. The work of (Nair et al., 2002) focuses on sparse regression and classification models using Mercer kernels, and (Sindhwani & Lozano, 2011) considers sparse regression with multiple kernels. Classical matching pursuit was developed for regression, but further extensions to logistic regression (Lozano et al., 2011) and smooth loss functions (Locatello et al., 2017) have also been studied. (Oglic & Gärtner, 2016) propose a greedy reconstruction technique for regression by empirically fitting squared error residuals. (Shahrampour & Tarokh, 2018) also use greedy methods for sparse approximation using multiple kernels.

Remark 2. *Our approach is radically different from the prior work as we characterize a set of features that maximize the metric entropy. Our feature construction and entropy optimization techniques are novel and have not been explored in the kernel approximation literature.*

7. Numerical Experiments

Benchmark Algorithm: We now compare **EOF** with the following benchmark algorithms on several datasets from the UCI Machine Learning Repository:

1) RKS (Rahimi & Recht, 2009) with approximated Laplace kernel feature $z(\mathbf{x}) = \frac{1}{\sqrt{M}}[\cos(\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_m + b_m)]_{m=1}^M$, where $\{\boldsymbol{\gamma}_m\}_{m=1}^M$ are sampled from a Cauchy distribution multiplied by σ, and $\{b_m\}_{m=1}^M$ are sampled from the uniform distribution on $[0, 2\pi]$.
2) ORF (Felix et al., 2016) with approximated Gaussian kernel feature $z(\mathbf{x}) = \frac{1}{\sqrt{M}}[\cos(\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_m + b_m)]_{m=1}^M$, with $[\boldsymbol{\gamma}_1\ \boldsymbol{\gamma}_2 \cdots \boldsymbol{\gamma}_m] = \sigma\mathbf{SQ}$ where $\mathbf{S}$ is a diagonal matrix, with diagonal entries sampled i.i.d. from the χ-distribution with d degrees and $\mathbf{Q}$ is the orthogonal matrix obtained from the QR decomposition of a matrix $\mathbf{G}$ with normally distributed entries. Note that ORF approximates a Gaussian kernel.
3) LKRF (Sinha & Duchi, 2016) with approximated Laplace kernel feature $z(\mathbf{x}) = \frac{1}{\sqrt{M}}[\cos(\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_m + b_m)]_{m=1}^M$, where first $M_0 > M$ random features are sampled and then re-weighted by solving a kernel alignment optimization. The top M random features would be used in the training.
4) EERF (Shahrampour et al., 2018), with approximated Laplace kernel feature $z(\mathbf{x}) = \frac{1}{\sqrt{M}}[\cos(\mathbf{x}^\mathsf{T}\boldsymbol{\gamma}_m + b_m)]_{m=1}^M$, where first $M_0 > M$ random features are sampled and then re-weighted according to a score function. The top M random features would appear in the training.

Experiment Setup: We also use approximated Laplace kernel feature $z(\mathbf{x}) = [C_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})]_{(\mathbf{l},\mathbf{i})\in S_n^*}$ where $\phi_{\mathbf{l},\mathbf{i}} = \prod_{d=1}^D \phi_{l_d,i_d}$ with ϕ_{l_d,i_d} defined in (4). To determine the value of σ used in **RKS**, **EERF**, **LKRF** and **ORF** we choose the value of σ^{-1} for each dataset to be the mean distance of the 50^{th} ℓ_2 nearest neighbor (Felix et al., 2016). We then calculate the corresponding ω for **EOF** associated to σ. The number of features in **EOF** is a function of dimension D and level n, so it is not possible to calculate them for any M. To resolve this issue, for any given M, we select the set S_n^* defined in (11) that satisfies

$$\operatorname{card}\big(S_{n-1}^*\big) < M \leq \operatorname{card}\big(S_n^*\big)$$

and randomly select M pairs of $(\mathbf{l},\mathbf{i}) \in S_n^*$ to have a random set S_M. We then use the following feature vector:

$$z_M(\mathbf{x}) := [C_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})]_{(\mathbf{l},\mathbf{i})\in S_M}.$$

This is equivalent to randomly select M rows from the feature $z(\mathbf{x}) = [C_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})]_{(\mathbf{l},\mathbf{i})\in S_n^*}$.

We let $M_0 = 10M$ for **LKRF** and **EERF**, then for any M, we compare the performance of different algorithms.

Datasets: In Table 1, we report the number of training samples N_{train} and test samples N_{test} used for each dataset. For the MNIST data set, we map the original $784-$dimensional data to a $32-$dimensional space using an auto-encoder. If the training and test samples are not provided separately for a dataset, we split it randomly. We standardize the data as follows: we scale each input to the unit interval $[0, 1]$ and the responses in regression to be inside $[-1, 1]$.

Comparison: For a fixed number of features, we perform 50 simulation runs for each algorithm on each data set. We then report the average test error (with standard errors) in Fig. 3, where the plot line is the mean error of an algorithm

Figure 3. Comparison of the test error of EOF (this work) versus benchmark algorithms including RKS, EERF, LKRF and ORF.

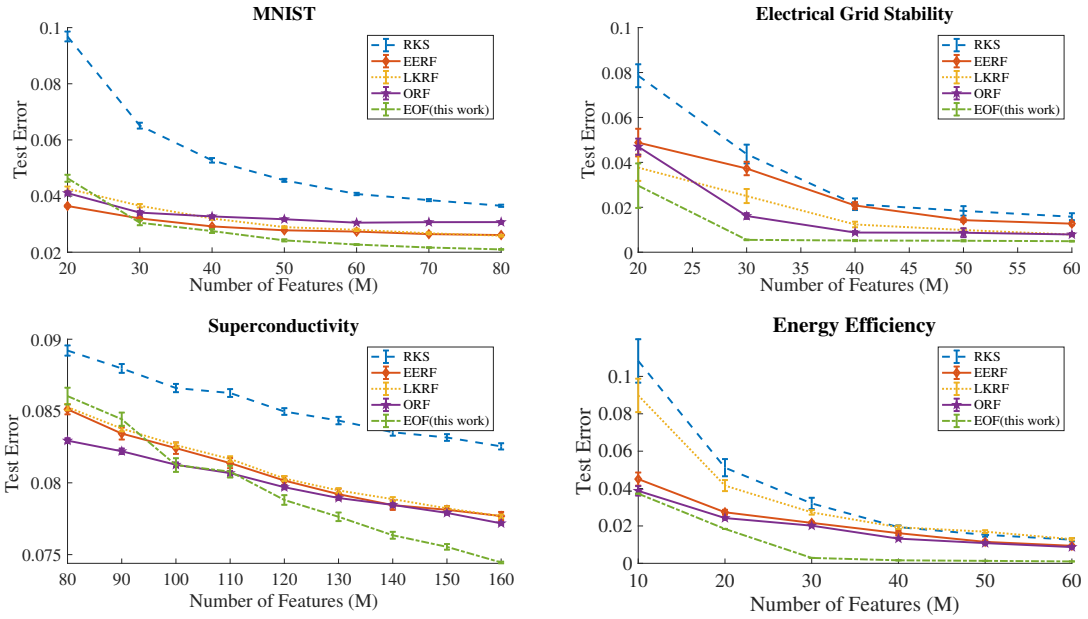


Table 1. Input dimension, number of training samples, and number of test samples are denoted by D, N_{train}, and N_{test}, respectively

DATA SET	TASK	D	N_{TRAIN}	N_{TEST}
MNIST	CLASSIFICATION	32	20000	10000
ELECTRICAL GRIDS STABILITY	CLASSIFICATION	13	7000	3000
SUPERCONDUCTIVITY	REGRESSION	81	15000	6263
ENERGY EFFICIENCY	REGRESSION	8	512	256

and the error bar reflects the standard deviation of the error. Throughout our experiments, we can see that **EOF** consistently improves the test error compared to randomized-feature algorithms. This is specifically visible when the gap between S_M and S_n^* becomes very small and, due to the optimality of S_n^*, **EOF** outperforms any random feature algorithm.

In Table 2, we also compare the time complexity and space complexity. We define the feature matrix

$$F := [z(\mathbf{x}_i)]_{i=1}^N,$$

which is an $M \times N$ matrix with M the number of features and N the number of data. Due to the sparse structure of **EOF**, we can also see that the number of non-zero entries of the F associated to **EOF** is smaller than other methods. When both the dimension D and the size of data N are large, the sparsity of **EOF** becomes more obvious as shown in the case of MNIST. The time cost of running **EOF** is also quite impressive. It is consistently better than **EERF** and **LKRF** and slightly slower than **RKS**. In fact, the major time for **EOF** is spent on feature matrix construction. For random features, due to high efficiency of matrix operations in Matlab, feature construction is fast. However, for **EOF** the feature construction via matrix operations is not possible in an efficient way. We observed that after the feature matrix construction, **EOF** is the fastest method in training. For example, if we only consider the training time (excluding feature construction) as the time cost, in kernel ridge regression on the dataset Superconductivity, the comparison between **RKS** and **EOF** is reported in Table 3, where **EOF** incurs a smaller time cost.

The run time is obtained on a Macbook Pro with a 4-core, 3.3 GHz Intel Core i5 CPU and 8 GB of RAM (2133Mhz).

8. Conclusion

We consider the approximation of kernels that satisfy Condition 1. We construct a set of mutually orthogonal features (with nested and compact supports) for these kernels and select the best M of them that maximize the entropy of the associated projector. The nested and compact support of features greatly reduces the time and space costs for feature matrix operations. The orthogonality and entropic optimality reduce dramatically the error of approximation (as well as generalization). Using our approximation method for supervised learning, we can establish a generalization error bound which indicates that only $\mathcal{O}(N^{\frac{1}{4}})$ features (disregarding the log factors) are needed to achieve the $\mathcal{O}(N^{-\frac{1}{2}})$ optimal accuracy. In terms of generalization, the main advantage of this work is reducing the number of features

Table 2. Time and space complexity comparison. We denote by nnz the number of non-zero elements.

MNIST				
Method	M	M_0	T_{train}	nnz(F)
RKS	80		1.64	1.6×10^6
EERF	80	800	4.43	1.6×10^6
LKRF	80	800	3.07	1.6×10^6
ORF	80		1.21	1.6×10^6
EOF	80	2048	2.45	2.5×10^5

Electrical Grids Stability				
Method	M	M_0	T_{train}	nnz(F)
RKS	60		0.04	4.2×10^5
EERF	60	600	0.14	4.2×10^5
LKRF	60	600	0.13	4.2×10^5
ORF	60		0.06	4.2×10^5
EOF	60	338	0.08	1.3×10^5

Superconductivity				
Method	M	M_0	T_{train}	nnz(F)
RKS	160		0.10	2.4×10^6
EERF	160	1600	0.45	2.4×10^6
LKRF	160	1600	0.37	2.4×10^6
ORF	160		0.13	2.4×10^6
EOF	160	161	0.14	1.2×10^6

Energy Efficiency				
Method	M	M_0	T_{train}	nnz(F)
RKS	60		0.01	6.1×10^3
EERF	60	600	0.05	6.1×10^3
LKRF	60	600	0.06	6.1×10^3
ORF	60		0.02	6.1×10^3
EOF	60	128	0.03	1.0×10^3

Table 3. Comparison on RKS and EOF in pure training excluding feature construction.

	$M = 80$	$M = 100$	$M = 120$	$M = 140$	$M = 160$
RKS	2×10^{-3}	3×10^{-3}	4×10^{-3}	5×10^{-3}	6×10^{-3}
EOF	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}

required to achieve the optimal accuracy (compared to state-of-the-art). Future directions include extending this method to a broader class of kernels.

Acknowledgments

R. Tuo gratefully acknowledges the support of NSF DMS-1914636. The authors would like to thank Soheil Kolouri (HRL) for providing the auto-encoded MNIST dataset.

References

Agrawal, R., Campbell, T., Huggins, J., and Broderick, T. Data-dependent compression of random features for large-scale kernel approximation. In *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1822–1831, 2019.

Avron, H., Kapralov, M., Musco, C., Musco, C., Velingker, A., and Zandieh, A. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 253–262, 2017.

Bungartz, H.-J. and Griebel, M. Sparse grids. *In: Acta Numerica. Vol. 13, pp. 147-269*, 13, 05 2004. doi: 10.1017/S0962492904000182.

Choromanski, K., Rowland, M., Sarlós, T., Sindhwani, V., Turner, R., and Weller, A. The geometry of random features. In *International Conference on Artificial Intelligence and Statistics*, pp. 1–9, 2018.

Cotter, A., Keshet, J., and Srebro, N. Explicit approximations of the gaussian kernel. *arXiv preprint arXiv:1109.4603*, 2011.

Dick, J., Kuo, F., and Sloan, I. High-dimensional integration: The quasi-monte carlo way. *Acta Numerica*, 22, 05 2013. doi: 10.1017/S0962492913000044.

Ding, L. and Zhang, X. Scalable stochastic kriging with markovian covariances, 2018.

Ding, L., Mak, S., and Wu, C.-F. Bdrygp: a new gaussian process model for incorporating boundary information. 08 2019.

Drineas, P. and Mahoney, M. W. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(Dec): 2153–2175, 2005.

Dung, D., Temlyakov, V., and Ullrich, T. Hyperbolic cross approximation. 01 2016.

Evans, L. C. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010. ISBN 9780821849743 0821849743.

Felix, X. Y., Suresh, A. T., Choromanski, K. M., Holtmann-Rice, D. N., and Kumar, S. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pp. 1975–1983, 2016.

Fine, S. and Scheinberg, K. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2(Dec):243–264, 2001.

Joachims, T. Training linear SVM's in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 217–226, 2006.

Kellerer, H., Pferschy, U., and Pisinger, D. *Knapsack Problems*. 01 2004. ISBN 978-3-540-40286-2. doi: 10.1007/978-3-540-24777-7.

Le, Q., Sarlós, T., and Smola, A. Fastfood-approximating kernel expansions in loglinear time. In *International Conference on Machine Learning*, volume 85, 2013.

Locatello, F., Khanna, R., Tschannen, M., and Jaggi, M. A unified optimization view on generalized matching pursuit and frank-wolfe. In *Artificial Intelligence and Statistics*, pp. 860–868, 2017.

Lozano, A., Swirszcz, G., and Abe, N. Group orthogonal matching pursuit for logistic regression. In *Artificial Intelligence and Statistics*, pp. 452–460, 2011.

Munkhoeva, M., Kapushev, Y., Burnaev, E., and Oseledets, I. Quadrature-based features for kernel approximation. In *Advances in Neural Information Processing Systems*, pp. 9147–9156, 2018.

Nair, P. B., Choudhury, A., and Keane, A. J. Some greedy learning algorithms for sparse regression and classification with mercer kernels. *Journal of Machine Learning Research*, 3(Dec):781–801, 2002.

Oglic, D. and Gärtner, T. Greedy feature construction. In *Advances in Neural Information Processing Systems*, pp. 3945–3953, 2016.

Oglic, D. and Gärtner, T. Scalable learning in reproducing kernel krein spaces. In *International Conference on Machine Learning*, pp. 4912–4921, 2019.

Oliva, J. B., Dubey, A., Wilson, A. G., Póczos, B., Schneider, J., and Xing, E. P. Bayesian nonparametric kernel-learning. In *Artificial Intelligence and Statistics*, pp. 1078–1086, 2016.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rahimi, A. and Recht, B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, pp. 1313–1320, 2009.

Rudi, A. and Rosasco, L. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pp. 3218–3228, 2017.

Shahrampour, S. and Tarokh, V. Learning bounds for greedy approximation with explicit feature maps from multiple kernels. In *Advances in Neural Information Processing Systems*, pp. 4695–4706, 2018.

Shahrampour, S., Beirami, A., and Tarokh, V. On data-dependent random features for improved generalization in supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Sindhwani, V. and Lozano, A. C. Non-parametric group orthogonal matching pursuit for sparse learning with multiple kernels. In *Advances in Neural Information Processing Systems*, pp. 2519–2527, 2011.

Sinha, A. and Duchi, J. C. Learning kernels with random features. In *Advances In Neural Information Processing Systems*, pp. 1298–1306, 2016.

Smola, A. J. and Schökopf, B. Sparse greedy matrix approximation for machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 911–918, 2000.

van der Vaart, A. W. and Wellner, J. A. *Weak Convergence and Empirical Processes*. Springer, 1996.

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

Vincent, P. and Bengio, Y. Kernel matching pursuit. *Machine Learning*, 48(1-3):165–187, 2002.

Wang, Y. and Shahrampour, S. A general scoring rule for randomized kernel approximation with application to canonical correlation analysis. *arXiv preprint arXiv:1910.05384*, 2019.

Williams, C. and Seeger, M. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, 2001.

Xu, J.-W., Pokharel, P. P., Jeong, K.-H., and Principe, J. C. An explicit construction of a reproducing gaussian kernel Hilbert space. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, 2006.

Yang, C., Duraiswami, R., and Davis, L. Efficient kernel machines using the improved fast gauss transform. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, pp. 1561–1568, 2004.

Yang, J., Sindhwani, V., Avron, H., and Mahoney, M. Quasi-monte carlo feature maps for shift-invariant kernels. In *International Conference on Machine Learning*, pp. 485–493, 2014.

Yang, Z., Wilson, A., Smola, A., and Song, L. A la carte–learning fast kernels. In *Artificial Intelligence and Statistics*, pp. 1098–1106, 2015.

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

Yu, F. X., Kumar, S., Rowley, H., and Chang, S.-F. Compact nonlinear maps and circulant extensions. *arXiv preprint arXiv:1503.03893*, 2015.

Zaitsev, V. F. and Polyanin, A. D. *Handbook of Exact Solutions for Ordinary Differential Equations*. CRC Press, 2 edition, 2002.

Zhang, K., Tsang, I. W., and Kwok, J. T. Improved Nyström low-rank approximation and error analysis. In *International Conference on Machine Learning*, pp. 1232–1239. ACM, 2008.