TRF: Learning Kernels with Tuned Random Features

Alistair Shilton, Sunil Gupta, Santu Rana, Arun Kumar Venkatesh, Svetha Venkatesh

Applied Artificial Intelligence Institute (A^2I^2), Deakin University, Geelong, Australia
{alistair.shilton, sunil.gupta, santu.rana, aanjanapuravenk, svetha.venkatesh}@deakin.edu.au

Abstract

Random Fourier features (RFF) are a popular set of tools for constructing low-dimensional approximations of translation-invariant kernels, allowing kernel methods to be scaled to big data. Apart from their computational advantages, by working in the spectral domain random Fourier features expose the translation invariant kernel as a density function that may, in principle, be manipulated directly to tune the kernel. In this paper we propose selecting the density function from a reproducing kernel Hilbert space to allow us to search the space of all translation-invariant kernels. Our approach, which we call tuned random features (TRF), achieves this by approximating the density function as the RKHS-norm regularised least-squares best fit to an unknown "true" optimal density function, resulting in a RFF formulation where kernel selection is reduced to regularised risk minimisation with a novel regulariser. We derive bounds on the Rademacher complexity for our method showing that our random features approximation method converges to optimal kernel selection in the large *N,D* limit. Finally, we prove experimental results for a variety of real-world learning problems, demonstrating the performance of our approach compared to comparable methods.

1 Introduction

Kernel based learning is an elegant and powerful family of techniques in machine learning (Cristianini and Shawe-Taylor 2005; Hastie, Tibshirani, and Friedman 2001; Herbrich 2002; Schölkopf and Smola 2001; Shawe-Taylor and Cristianini 2004; Steinwart and Christman 2008; Vapnik 1995; Suykens et al. 2002). Rather than constructing a complex parametric model and then learning its parameters, kernel-based methods encode this complexity into a kernel and then learn a linear (dual) representation using representor theory. A significant theoretical framework has been developed demonstrating the advantages of this approach, backed by substantial experimental evidence. However the computational complexity typically scales as $\mathcal{O}(N^3)$, where N is the training set size, so kernel methods may not scale well for large datasets. Further, kernel selection is often ad-hoc, relying heavily on user knowledge and guesswork (which kernels to consider etc), and can be slow if global optimisation methods such as Bayesian optimisation are used to tune hyper-parameters like weights, length-scales etc.

Random Fourier features (RFF) (Rahimi and Recht 2006; Liu et al. 2020) was originally developed to tackle the problem of computational complexity. RFF-based methods work by approximating the feature map underlying a kernel using a finite (D-) dimensional map obtained by sampling from a density function - typically, but not necessarily (Chang et al. 2017; Liu et al. 2020; Bullins, Zhang, and Zhang 2017), the spectral density corresponding to the kernel by Bochner's theorem. Using this map, the problem is re-cast in approximate feature space and solved in primal form, reducing the typical complexity to $\mathcal{O}(ND^3)$. Building on this, methods have been developed that take advantage of the fact that the feature space is exposed to, in effect, tune the feature map. For example random features may instead be drawn to maximise some criteria, typically kernel alignment (Li et al. 2019a; Yu et al. 2015; Bullins, Zhang, and Zhang 2017; Sinha and Duchi 2016). Alternatively, Fourier kernel learning (FKL) (Băzăvan, Li, and Sminchisescu 2012) directly learns feature weights during training, in effect making the density itself an optimisation parameter. The elegance and directness of FKL make it particularly attractive from a practical standpoint; however, regularising the feature weights using a Euclidean norm effectively casts the density function in reproducing kernel Hilbert space with a delta (diagonal) kernel, which is somewhat restrictive and does not afford the user an opportunity to incorporate their expectations regarding the spectral structure of the optimal kernel.

In this paper we propose an algorithm, tuned random features (TRF), to perform simultaneous regularised risk minimisation and kernel learning in spectral space. Our algorithm incorporates kernel regularisation in spectral domain to prevent kernel over-fitting and uses a *meta-kernel* to allow users to specify the characteristics that we expect the optimal kernel to have, or, more precisely, that we expect the optimal kernel's spectral density to have. To achieve this, we let the density function itself be the parameter (function) to be selected from a reproducing kernel Hilbert space (RKHS), where the meta-kernel defining this RKHS captures the characteristics we expect of the kernel's spectral density. We incorporate kernel selection directly into the regularised empirical risk minimisation problem formulation, with regularisation *towards* a default (reference) kernel. Using representor and RFF theory, we obtain a convex optimisation problem combining kernel

learning - that is, learning a function in reproducing kernel Hilbert space $\mathcal{H}_K$ - and learning the kernel K itself.

We show that TRF is convex and readily trained using gradient-based methods, and demonstrate uniform convergence as $N, D \to \infty$ using Rademacher complexity analysis, given an appropriate "schedule" of hyper-parameters. Experimentally, we have tested TRF on a variety of small and large real-world problems and showed that TRF outperforms comparable methods on most occasions.

1.1 Notation

Column vectors are written in lower-case bold $\mathbf{a}, \mathbf{b}, \ldots$ with elements a_i, so $\mathbf{a} = [a_i]_i$, matrices in upper-case bold $\mathbf{A}, \mathbf{B}, \ldots$ with elements $A_{i,j}$, so $\mathbf{A} = [A_{i,j}]_{i,j}$, and $\mathbf{a}^\dagger = \mathbf{a}^{*\mathrm{T}}$, $\mathbf{A}^\dagger = \mathbf{A}^{*\mathrm{T}}$ is the conjugate transpose. The Hadamard (elementwise) product is denoted $\mathbf{a} \odot \mathbf{b} = [a_i b_i]_i$, the Hadamard power $\mathbf{a}^{\odot b} = [a_i^b]_i$, and the elementwise norm $|\mathbf{a}| = [|a_i|]_i$. $\mathbb{N}_n = \{0, 1, \ldots, n-1\}$ is the integers modulo n. The weighted inner product is $\langle \zeta, \gamma \rangle_\rho = \int \zeta^*(\boldsymbol{\omega}) \gamma(\boldsymbol{\omega}) \rho(\boldsymbol{\omega}) d\boldsymbol{\omega}$ for $\rho : \mathbb{X} \to \mathbb{R}_+$, and the weighted norm $\|\zeta\|_\rho^2 = \langle \zeta, \zeta \rangle_\rho$. The reproducing kernel Hilbert space (RKHS) norm is $\|\cdot\|_{\mathcal{H}_K}$ for kernel $K : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$. We denote by $\mathcal{H}_\kappa^\oplus = \{\sigma \in \mathcal{H}_\kappa : \sigma(\boldsymbol{\omega}) = \sigma(-\boldsymbol{\omega}) \geq 0 \ \forall \boldsymbol{\omega}\}$. The training set is $\mathtt{D} = \{(\mathbf{x}_i, y_i) \in \mathbb{X} \times y_i \in \mathbb{Y} | i \in \mathbb{N}_N\}$ where training pairs $(\mathbf{x}_i, y_i) \sim \mathcal{S}$ are drawn i.i.d. from distribution $\mathcal{S}$. We use N for the size of the training set, d for its input dimension $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^d$, and D for the dimension of the random feature map $\mathbf{z} : \mathbb{R}^d \to \mathbb{C}^D$.

2 Background: Random Fourier Features

As first introduced in (Rahimi and Recht 2006), random Fourier features (RFF) (Liu et al. 2020) allow kernel methods to be scaled to large data by approximating the kernel by $K(\mathbf{x}, \mathbf{x}') \approx \mathbf{z}^\dagger(\mathbf{x})\mathbf{z}(\mathbf{x}')$ for some finite-dimensional $\mathbf{z} : \mathbb{R}^d \to \mathbb{C}^D$. So for example rather than learning a function $f \in \mathcal{H}_K \oplus \mathbb{C}$ for positive-definite kernel K:

$$f(\mathbf{x}) = \textstyle\sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{1}$$

which has a typical complexity $\mathcal{O}(N^3)$ to find $\boldsymbol{\alpha}$ and requires $\mathcal{O}(N^2)$ memory, we may instead learn:

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = \mathbf{w}^\dagger \mathbf{z}(\mathbf{x}) + b$$

which has a typical complexity $\mathcal{O}(ND^2)$ to find $\boldsymbol{\tau}$ and requires $\mathcal{O}(ND)$ memory. By making the complexity linear in N it becomes feasible to scale SVMs (and many other kernel-based methods such as Gaussian Processes) to large datasets.

For translation-invariant kernels (Genton 2001) the basis of random Fourier features is Bochner's theorem (Bochner 1932) - for a continuous, translation invariant, positive definite kernel $K(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$, there exists an even spectral density function $\rho : \mathbb{R}^d \to \mathbb{R}_+$ so that:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= \textstyle\int_{\mathbb{R}^d} e^{i\boldsymbol{\omega}^{\mathrm{T}}(\mathbf{x}-\mathbf{x}')} \rho(\boldsymbol{\omega}) d\boldsymbol{\omega} && \text{(Fourier xform)} \\ &= \textstyle\int_{\mathbb{R}^d} \xi^*(\boldsymbol{\omega}; \mathbf{x}) \xi(\boldsymbol{\omega}; \mathbf{x}') \rho(\boldsymbol{\omega}) d\boldsymbol{\omega} && \text{(Split } \mathbf{x}, \mathbf{x}' \text{ terms)} \\ &= \langle \xi(\cdot; \mathbf{x}), \xi(\cdot; \mathbf{x}') \rangle_\rho && \text{(Inner product in feature space)} \\ &= \mathbb{E}_{\boldsymbol{\omega} \sim \rho} [\xi^*(\boldsymbol{\omega}; \mathbf{x}) \xi(\boldsymbol{\omega}; \mathbf{x}')] && \text{(As an expectation)} \end{aligned} \tag{2}$$

where $\xi(\boldsymbol{\omega}; \mathbf{x}) = e^{i\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}}$, and $\xi(\cdot; \mathbf{x}) = [\xi(\boldsymbol{\omega}; \mathbf{x})]_{\boldsymbol{\omega} \in \mathbb{R}^d}$ is the feature map for kernel K. Thus, re-writing in feature-space form, (1) becomes (Bach 2017, Appendix A):

$$f(\mathbf{x}) = \langle \tau(\cdot), \xi(\cdot; \mathbf{x}) \rangle_\rho + b = \mathbb{E}_{\boldsymbol{\omega} \sim \rho}[\tau^*(\boldsymbol{\omega}) \xi(\boldsymbol{\omega}; \mathbf{x})] + b \tag{3}$$

where $\tau : \mathbb{R}^d \to \mathbb{C}$ is the weight function and $b \in \mathbb{C}$ is the bias. Thus if $\boldsymbol{\omega} \sim \rho$ then $\xi^*(\boldsymbol{\omega}; \mathbf{x}) \xi(\boldsymbol{\omega}; \mathbf{x}')$ and $\tau^*(\boldsymbol{\omega}) \xi(\boldsymbol{\omega}; \mathbf{x})$ are unbiased estimates of $K(\mathbf{x}, \mathbf{x}')$ and $f(\mathbf{x}) - b$, respectively. By sampling $\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{D-1} \sim \rho$ we obtain the Monte-Carlo (MC) approximations of K and f:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &\approx \tilde{K}(\mathbf{x}, \mathbf{x}') = \mathbf{z}^\dagger(\mathbf{x}) \mathbf{z}(\mathbf{x}') \\ f(\mathbf{x}) &\approx \tilde{f}(\mathbf{x}) = \boldsymbol{\tau}^\dagger \mathbf{z}(\mathbf{x}) + b \end{aligned} \tag{4}$$

where $\mathbf{z}(\mathbf{x}) = [e^{i\boldsymbol{\omega}_i^{\mathrm{T}}\mathbf{x}}/\sqrt{D}]_{i \in \mathbb{N}_D}$ is the random feature map and $\boldsymbol{\tau} = [\tau(\boldsymbol{\omega}_i)/\sqrt{D}]_i$ is the weight vector. This approximate feature map has dimension D. By working in the approximate feature space we reduce computational complexity to $\mathcal{O}(ND^2)$ rather than $\mathcal{O}(N^3)$, which is scalable to large N.

It is not necessary to sample from the distribution ρ defined by K (Yang, Sindhwanim, and Mahoney 2014; Avron et al. 2016; Chang et al. 2017; Bullins, Zhang, and Zhang 2017; Liu et al. 2020; Li et al. 2019b). Given a strictly positive, even reference density $\hat{\rho} : \mathbb{R}^d \to \mathbb{R}_+$ (which is associated with a reference kernel $\hat{K}(\mathbf{x}, \mathbf{x}') = \hat{k}(\mathbf{x} - \mathbf{x}')$ via Bochner's theorem), and defining $\mu(\hat{\boldsymbol{\omega}}) = \rho(\hat{\boldsymbol{\omega}})/\hat{\rho}(\hat{\boldsymbol{\omega}})$, (2) and (3) become:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= \langle \mu(\cdot) \xi(\cdot; \mathbf{x}), \xi(\cdot; \mathbf{x}') \rangle_{\hat{\rho}} \\ &= \mathbb{E}_{\hat{\boldsymbol{\omega}} \sim \hat{\rho}} [\mu(\hat{\boldsymbol{\omega}}) \xi^*(\hat{\boldsymbol{\omega}}; \mathbf{x}) \xi(\hat{\boldsymbol{\omega}}; \mathbf{x}')] \\ f(\mathbf{x}) &= \langle \mu(\cdot) \hat{\tau}(\cdot), \xi(\cdot; \mathbf{x}) \rangle_{\hat{\rho}} + b \\ &= \mathbb{E}_{\hat{\boldsymbol{\omega}} \sim \hat{\rho}} [\mu(\hat{\boldsymbol{\omega}}) \hat{\tau}^*(\hat{\boldsymbol{\omega}}) \xi(\hat{\boldsymbol{\omega}}; \mathbf{x})] + b \end{aligned}$$

Here $\mu(\hat{\boldsymbol{\omega}}) \xi^*(\hat{\boldsymbol{\omega}}; \mathbf{x}) \xi(\hat{\boldsymbol{\omega}}; \mathbf{x}')$ and $\mu(\hat{\boldsymbol{\omega}}) \hat{\tau}^*(\hat{\boldsymbol{\omega}}) \xi(\hat{\boldsymbol{\omega}}; \mathbf{x})$ are unbiased estimates of $K(\mathbf{x}, \mathbf{x}')$ and $f(\mathbf{x}) - b$, respectively, when $\hat{\boldsymbol{\omega}} \sim \hat{\rho}$. Hence by sampling $\hat{\boldsymbol{\omega}}_0, \hat{\boldsymbol{\omega}}_1, \ldots, \hat{\boldsymbol{\omega}}_{D-1} \sim \hat{\rho}$:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &\approx \tilde{K}(\mathbf{x}, \mathbf{x}') = (\mathbf{u} \odot \hat{\mathbf{z}}(\mathbf{x}))^\dagger \hat{\mathbf{z}}(\mathbf{x}') \\ f(\mathbf{x}) &\approx \tilde{f}(\mathbf{x}) = (\mathbf{u} \odot \hat{\boldsymbol{\tau}})^\dagger \hat{\mathbf{z}}(\mathbf{x}) + b \\ \hat{K}(\mathbf{x}, \mathbf{x}') &\approx \tilde{\hat{K}}(\mathbf{x}, \mathbf{x}') = \hat{\mathbf{z}}^\dagger(\mathbf{x}) \hat{\mathbf{z}}(\mathbf{x}') \end{aligned} \tag{5}$$

where $\hat{\mathbf{z}}(\mathbf{x}) = [e^{i\hat{\boldsymbol{\omega}}_i^{\mathrm{T}}\mathbf{x}}/\sqrt{D}]_{i \in \mathbb{N}_D}$ is the random feature map, $\hat{\boldsymbol{\tau}} = [\hat{\tau}(\hat{\boldsymbol{\omega}}_i)/\sqrt{D}]_i$ is the weight vector, and we call $\mathbf{u} = [\mu(\hat{\boldsymbol{\omega}}_i)]_i$ the density (ratio) vector. As demonstrated in for example (Li et al. 2019b; Avron et al. 2017), this approach may give a better approximation of K from fewer samples (for example the features $\hat{\boldsymbol{\omega}}_i$ may be placed according to the data dependent empirical ridge leverage score distribution). Table 1 summarises the definitions and notations for RFF for both the standard MC and weighted-MC approaches.

2.1 (Spectral) Kernel Selection

Having exposed the feature map in spectral form, it is natural to ask if kernel selection may be done by tuning the (spectrally sampled) feature map $\mathbf{z} : \mathbb{R}^d \to \mathbb{R}^D$. For example, rather than drawing features $\boldsymbol{\omega}_i$ from ρ or $\hat{\rho}$ we may select them to maximise kernel alignment (Li et al. 2019a; Yu et al. 2015; Bullins, Zhang, and Zhang 2017), (Cristianini et al. 2002; Cortes, Mohri, and Rostamizadeh 2012); or we may give ρ a parametric form such as a mixture of Gaussians (Wilson and Adams 2013) or something more general (Yang

	Standard Form	Modified (weighted) Form
Kernel	$K(\mathbf{x},\mathbf{x}') = \langle \xi(\cdot;\mathbf{x}), \xi(\cdot;\mathbf{x}') \rangle_\rho$	$K(\mathbf{x},\mathbf{x}') = \langle \mu(\cdot)\xi(\cdot;\mathbf{x}), \xi(\cdot;\mathbf{x}') \rangle_{\hat{\rho}}$
Reference Kernel	–	$\hat{K}(\mathbf{x},\mathbf{x}') = \langle \xi(\cdot;\mathbf{x}), \xi(\cdot;\mathbf{x}') \rangle_{\hat{\rho}}$
Function Form	$f(\mathbf{x}) = \langle \tau(\cdot), \xi(\cdot;\mathbf{x}) \rangle_\rho + b$	$f(\mathbf{x}) = \langle \mu(\cdot)\hat{\tau}(\cdot), \xi(\cdot;\mathbf{x}) \rangle_{\hat{\rho}} + b$
Feature map	$\xi(\cdot;\mathbf{x}) = [e^{i\boldsymbol{\omega}^\mathrm{T}\mathbf{x}}]_{\boldsymbol{\omega}\in\mathbb{R}^d}$	$\xi(\cdot;\mathbf{x}) = [e^{i\hat{\boldsymbol{\omega}}^\mathrm{T}\mathbf{x}}]_{\hat{\boldsymbol{\omega}}\in\mathbb{R}^d}$
Weight function	$\tau : \mathbb{R}^d \to \mathbb{C}, \tau \in L_{2,\rho}$	$\hat{\tau} : \mathbb{R}^d \to \mathbb{C}, \mu(\cdot)\hat{\tau}(\cdot) \in L_{2,\hat{\rho}}$
Feature weights	–	$\mu(\hat{\boldsymbol{\omega}}) = \frac{\rho(\hat{\boldsymbol{\omega}})}{\hat{\rho}(\hat{\boldsymbol{\omega}})}$
	Monte-Carlo Approximate	Weighted Monte-Carlo Approximation
Kernel	$K(\mathbf{x},\mathbf{x}') \approx \tilde{K}(\mathbf{x},\mathbf{x}') = \mathbf{z}^\dagger(\mathbf{x})\mathbf{z}(\mathbf{x}')$	$K(\mathbf{x},\mathbf{x}') \approx \tilde{K}(\mathbf{x},\mathbf{x}') = (\mathbf{u}\odot\hat{\mathbf{z}}(\mathbf{x}))^\dagger \hat{\mathbf{z}}(\mathbf{x}')$
Reference Kernel	–	$\hat{K}(\mathbf{x},\mathbf{x}') \approx \tilde{\hat{K}}(\mathbf{x},\mathbf{x}') = \hat{\mathbf{z}}^\dagger(\mathbf{x})\hat{\mathbf{z}}(\mathbf{x}')$
Function Form	$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = \boldsymbol{\tau}^\dagger\mathbf{z}(\mathbf{x}) + b$	$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = (\mathbf{u}\odot\hat{\boldsymbol{\tau}})^\dagger \hat{\mathbf{z}}(\mathbf{x}) + b$
Feature map	$\mathbf{z}(\mathbf{x}) = \left[\frac{1}{\sqrt{D}} e^{i\boldsymbol{\omega}_i^\mathrm{T}\mathbf{x}}\right]_{i\in\mathbb{N}_D}$	$\hat{\mathbf{z}}(\mathbf{x}) = \left[\frac{1}{\sqrt{D}} e^{i\hat{\boldsymbol{\omega}}_i^\mathrm{T}\mathbf{x}}\right]_{i\in\mathbb{N}_D}$
Weight vector	$\boldsymbol{\tau} = \left[\frac{1}{\sqrt{D}}\tau(\boldsymbol{\omega}_i)\right]_{i\in\mathbb{N}_D}$	$\hat{\boldsymbol{\tau}} = \left[\frac{1}{\sqrt{D}}\hat{\tau}(\hat{\boldsymbol{\omega}}_i)\right]_{i\in\mathbb{N}_D}$
Density vector	–	$\mathbf{u} = [\mu(\hat{\boldsymbol{\omega}}_i)]_{i\in\mathbb{N}_D}$
Feature space inner product	$\langle \zeta,\gamma \rangle_\rho = \int_{\mathbb{R}^d} \zeta^*(\boldsymbol{\omega})\gamma(\boldsymbol{\omega})\rho(\boldsymbol{\omega})\,d\boldsymbol{\omega}$	$\langle \zeta,\gamma \rangle_{\hat{\rho}} = \int_{\mathbb{R}^d} \zeta^*(\hat{\boldsymbol{\omega}})\gamma(\hat{\boldsymbol{\omega}})\hat{\rho}(\hat{\boldsymbol{\omega}})\,d\hat{\boldsymbol{\omega}}$
Samples	$\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{D-1} \sim \rho$	$\hat{\boldsymbol{\omega}}_0, \hat{\boldsymbol{\omega}}_1, \ldots, \hat{\boldsymbol{\omega}}_{D-1} \sim \hat{\rho}$

Figure 1: Summary of RFF and related notations. In this table $f \in \mathcal{H}_K \oplus \mathbb{C}$ is the function we wish to learn, where K is a translation invariant kernel with corresponding density ρ as per (2) (Bochner's theorem). The upper-left quadrant shows the spectral form of f, and the upper-right quadrant the modified (weighted) spectral form using reference kernel $\hat{K}$. The bottom-left shows the Monte-Carlo approximation with D samples, and the bottom right the weighted-Monte-Carlo approximation, where we use $\tilde{f}$ and $\tilde{K}$ to indicate RFF approximations of f and K, respectively.

et al. 2015) to obtain a kernel mixture that may be tuned. Alternatively, Fourier kernel learning (FKL) (Băzăvan, Li, and Sminchisescu 2012) proposes directly selecting $\mathbf{u} \in \mathbb{R}_+^D$ during training with a regularisation term $\|\mathbf{u}\|_2^2$.

In the present paper we propose selecting (learning) a weight function $\mu \in \mathcal{H}_\kappa^\oplus$ in spectral space, where κ is a kernel (we call κ a *meta-kernel*) that defines the characteristics we expect of the spectral density $\rho(\cdot) = \mu(\cdot)\hat{\rho}(\cdot)$ without restricting it's exact form. Interestingly we note that FKM is a variant (special case) of our method where we use the meta-kernel $\kappa(\boldsymbol{\omega},\boldsymbol{\omega}') = \mathbb{1}_{\boldsymbol{\omega}=\boldsymbol{\omega}'}$ and substitute $\hat{\mu}(\cdot) = 0$ (so $\hat{\mathbf{v}} = \tilde{\hat{\mathbf{v}}} = \mathbf{0}$ - see sections 3.1, 3.2).

For completeness we note standard approaches to kernel selection and tuning, which typically involve selecting a finite set of test kernels and using grid-search, Bayesian optimisation or similar methods to tune parameters (e.g. length-scale); or multi-kernel learning (Gönen and Alpaydin 2011). While powerful, these ad-hoc approaches are often computationally expensive and restrict the search space to the span of a small, pre-defined set of kernels. Alternatively, hyper-kernel methods (Ong, Williamson, and Smola 2003; Ong, Smola, and Williamson 2005) select K from a hyper-RKHS defined by a hyper-kernel $\underline{K}$. However, hyper-kernel methods tend to be computationally complex, scaling as $\mathcal{O}(N^3)$ or worse.

3 Tuned Random Features

In this section we introduce our method, tuned random features (TRF), for combining kernel selection and regularised risk minimisation using random Fourier features. We begin by constructing a regularised empirical risk minimisation formulation in Fourier feature space. We then apply modified random Fourier features techniques to make the formulation practically attainable.

3.1 Learning the Kernel in the Spectral Domain

Given a training set $\mathtt{D} = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{Y} | i \in \mathbb{N}_N\}$ of N samples $(\mathbf{x}_i, y_i) \sim \mathcal{S}$ drawn i.i.d. from a distribution $\mathcal{S}$, our goal is to select both $f \in \mathcal{H}_K \oplus \mathbb{C}$, where $\mathcal{H}_K$ is the reproducing kernel Hilbert space defined by translation-invariant kernel K, and the kernel K defining the hypothesis space $\mathcal{H}_K$ itself to minimise the regularised empirical risk:

$$R_N(f) = \frac{1}{N}\sum_i \ell(f(\mathbf{x}_i), y_i) + \frac{\lambda}{2}\|f\|_{\mathcal{H}_K}^2$$

where ℓ is a loss function whose form depends on the problem being solved. Equivalently, in modified Fourier feature space (Table 1, upper-right quadrant), we may minimise:

$$R_N(\hat{\tau}, b) = \frac{1}{N}\sum_i \ell\left(\langle \mu(\cdot)\hat{\tau}(\cdot), \xi(\cdot;\mathbf{x}_i)\rangle_{\hat{\rho}} + b, y_i\right) + \frac{\lambda}{2}\langle \mu(\cdot)\hat{\tau}(\cdot), \hat{\tau}(\cdot)\rangle_{\hat{\rho}} \quad (6)$$

To incorporate kernel selection into this formulation we propose letting $\mu \in \mathcal{H}_\kappa^\oplus = \{\sigma \in \mathcal{H}_\kappa : \sigma(\hat{\boldsymbol{\omega}}) = \sigma(-\hat{\boldsymbol{\omega}}) \geq 0\ \forall \hat{\boldsymbol{\omega}} \in \mathbb{R}^D\}$ be an even, strictly positive function for some meta-kernel κ defining the spectral characteristics we expect of $\rho(\cdot) = \mu(\cdot)\hat{\rho}(\cdot)$ and hence indirectly, by Bochner's theorem, our expectations of the kernel K. In modified Fourier

feature space (Table 1, top-right), we aim to find:

$$f(\mathbf{x}) = \langle \mu(\cdot)\hat{\tau}(\cdot), \xi(\cdot;\mathbf{x})\rangle_{\hat{\rho}} + b \tag{7}$$

where $\hat{\tau} : \mathbb{R}^d \to \mathbb{C}$, $b \in \mathbb{C}$, $\mu \in \mathcal{H}_\kappa^\oplus$ minimise the *tuned regularised empirical risk*:

$$\begin{aligned} Q_N(\hat{\tau}, b, \mu) = \tfrac{1}{N}\textstyle\sum_i \ell\Big(\langle \mu(\cdot)\hat{\tau}(\cdot), \xi(\cdot;\mathbf{x}_i)\rangle_{\hat{\rho}} + b, y_i\Big) \\ + \tfrac{\lambda}{2}\langle \mu(\cdot)\hat{\tau}(\cdot), \hat{\tau}(\cdot)\rangle_{\hat{\rho}} + \tfrac{\Lambda}{2}\|\mu - \hat{\mu}\|^2_{\mathcal{H}_\kappa} \end{aligned} \tag{8}$$

and:

$$\hat{\mu}(\cdot) = \operatorname*{argmin}_{\hat{\mu}\in\mathcal{H}_\kappa^\oplus} \textstyle\int_{\mathbb{R}^d} (\hat{\mu}(\hat{\boldsymbol{\omega}}) - 1)^2 \hat{\rho}(\hat{\boldsymbol{\omega}})\, d\hat{\boldsymbol{\omega}}$$

is the function in $\mathcal{H}_\kappa$ that most closely approximates $\hat{\mu}(\cdot) = 1$ in the least-squares sense.[1] In this formulation $\hat{\rho}$ is defined by the (fixed) reference kernel $\hat{K}$, and $\mu \in \mathcal{H}_\kappa^\oplus$ is characterised by the positive definite meta-kernel κ. As per our definitions in previous section and Table 1, the kernel K is defined by the density function $\rho(\cdot) = \mu(\cdot)\hat{\rho}(\cdot)$, so minimising Q_N for μ optimises ρ and hence tunes K to minimise the tuned regularised empirical risk. Note that:

1. The first two terms in the tuned regularised empirical risk (8) are the standard empirical risk and RKHS-norm regularisation terms as per (6).
2. The final term is a regularisation term designed to prevent over-fitting of ρ (and hence K). It is designed to regularise *toward* $\mu = \hat{\mu} \approx 1$ in the limit $\Lambda \to \infty$, which corresponds to $\rho \approx \hat{\rho}$ and hence $K \approx \hat{K}$. It follows that $\hat{K}$ acts as a default (fallback) kernel in the strong regularisation limit. Note that if we regularised using $\frac{\Lambda}{2}\|\mu\|^2_{\mathcal{H}_\kappa}$ then $\mu \to 0$ in the limit, which corresponds to $K = 0$ and is therefore unhelpful.

To finish, we further simplify the tuned regularised empirical risk minimisation problem by defining $\eta(\cdot) = \hat{\tau}(\cdot)\mu(\cdot)$. In terms of η, b in this paper we aim to find:

$$\boxed{f(\mathbf{x}) = \langle \eta(\cdot), \xi(\cdot;\mathbf{x})\rangle_{\hat{\rho}} + b} \tag{9}$$

The variables $\eta : \mathbb{R}^d \to \mathbb{C}$ and $b \in \mathbb{C}$ minimise the tuned regularised empirical risk minimisation problem (8), which we re-write in terms of η as follows:

$$\boxed{Q_N(\eta, b) = \tfrac{1}{N}\textstyle\sum_i \ell\Big(\langle \eta(\cdot), \xi(\cdot;\mathbf{x}_i)\rangle_{\hat{\rho}} + b, y_i\Big) + \tfrac{\lambda}{2} r(\eta)} \tag{10}$$

In this expression r is a regulariser of the form:

$$\boxed{r(\eta) = \min_{\mu\in\mathcal{H}_\kappa^\oplus} \|\eta(\cdot)\|^2_{\frac{\hat{\rho}(\cdot)}{\mu(\cdot)}} + \tfrac{\Lambda}{\lambda}\|\mu - \hat{\mu}\|^2_{\mathcal{H}_\kappa}} \tag{11}$$

where $\|\cdot\|^2_\rho = \langle\cdot,\cdot\rangle_\rho$. Thus we see that the tuned empirical risk minimisation formulation can be written as a novel form of regularised risk minimisation.

[1] We cannot guarantee $\hat{\mu} = 1$ in general. If the feature map $\boldsymbol{\varphi}_\kappa : \mathbb{R}^d \to \mathbb{R}^m$ associated with $\kappa(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\omega}}') = \boldsymbol{\varphi}_\kappa^{\mathrm{T}}(\hat{\boldsymbol{\omega}})\boldsymbol{\varphi}_\kappa(\hat{\boldsymbol{\omega}}')$ by Mercer's theorem includes a constant term then, as $\hat{\mu}(\cdot) = \hat{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\cdot)$ by definition, we can always select $\hat{\mathbf{v}}$ so that $\hat{\mu}(\cdot) = \hat{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\cdot) = 1 \in \mathcal{H}_\kappa^\oplus$. For example if $\kappa(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\omega}}') = (1 + \hat{\boldsymbol{\omega}}^{\mathrm{T}}\hat{\boldsymbol{\omega}}')^2$ and $d = 2$ then $\boldsymbol{\varphi}_\kappa(\boldsymbol{\omega}) = [1; \sqrt{2}\omega_0; \sqrt{2}\omega_1; \omega_0^2; \omega_1^2; \sqrt{2}\omega_0\omega_1]$ and $\hat{\mu}(\cdot) = \hat{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\cdot) = 1 \in \mathcal{H}_\kappa^\oplus$ when $\hat{\mathbf{v}} = [1; \mathbf{0}]$. However if κ is an RBF kernel then $1 \notin \mathcal{H}_\kappa$, though it may be approximated to arbitrary accuracy w.r.t. $\hat{\rho}$ as the RBF kernel is universal.

3.2 Learning the Kernel via Tuned Random Features

We now apply modified random Fourier features to (8)/(10). Selecting a reference kernel $\hat{K}$ and using Table 1, we see that, in terms of the weight vector $\hat{\boldsymbol{\tau}}$, density vector $\mathbf{u}$ and random feature map $\hat{\mathbf{z}} : \mathbb{R}^d \to \mathbb{C}^D$, the trained machine may be approximated using:

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = (\mathbf{u} \odot \hat{\boldsymbol{\tau}})^\dagger \hat{\mathbf{z}}(\mathbf{x}) + b$$

where $\mathbf{u} \geq \mathbf{0}$ (this is a relaxation of $\mu \in \mathcal{H}_\kappa^\oplus$ as we do not require that $\mu(\hat{\boldsymbol{\omega}}) \geq 0$ for $\hat{\boldsymbol{\omega}} \notin \{\hat{\boldsymbol{\omega}}_0, \hat{\boldsymbol{\omega}}_1, \ldots, \hat{\boldsymbol{\omega}}_{D-1}\}$), and $\hat{\boldsymbol{\tau}}$, b and $\mu \in \{\sigma \in \mathcal{H}_\kappa : \sigma(\hat{\boldsymbol{\omega}}_i) = u_i \geq 0 \forall i\}$ minimise:

$$\begin{aligned} \tilde{Q}_N(\hat{\boldsymbol{\tau}}, b, \mathbf{u}, \mu) = \tfrac{1}{N}\textstyle\sum_i \ell\Big((\mathbf{u}\odot\hat{\boldsymbol{\tau}})^\dagger \hat{\mathbf{z}}(\mathbf{x}_i) + b, y_i\Big) \\ + \tfrac{\lambda}{2}(\mathbf{u}\odot\hat{\boldsymbol{\tau}})^\dagger\hat{\boldsymbol{\tau}} + \tfrac{\Lambda}{2}\|\mu - \hat{\mu}\|^2_{\mathcal{H}_\kappa} \end{aligned} \tag{12}$$

To be useful in practice the final term must be approximated in terms of the random Fourier features representation. To this end we note that we can write $\mu, \hat{\mu} \in \mathcal{H}_\kappa^\oplus$ in feature space form $\mu(\cdot) = \mathbf{v}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\cdot)$ and $\hat{\mu}(\cdot) = \hat{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\cdot)$, where $\mathbf{v}, \hat{\mathbf{v}} \in \mathbb{R}^m$ and $\boldsymbol{\varphi}_\kappa : \mathbb{R}^d \to \mathbb{R}^m$ is the feature map associated with κ through Mercer's theorem. Define:

$$\begin{aligned} \mathbf{v} &= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^m} \tfrac{1}{2}\textstyle\int_{\mathbb{R}^d}\big(\mathbf{v}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\hat{\boldsymbol{\omega}}) - \mu(\boldsymbol{\omega})\big)^2\hat{\rho}(\hat{\boldsymbol{\omega}})\, d\hat{\boldsymbol{\omega}} \\ \hat{\mathbf{v}} &= \operatorname*{argmin}_{\hat{\mathbf{v}}\in\mathbb{R}^m} \tfrac{1}{2}\textstyle\int_{\mathbb{R}^d}\big(\hat{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\hat{\boldsymbol{\omega}}) - 1\big)^2\hat{\rho}(\hat{\boldsymbol{\omega}})\, d\hat{\boldsymbol{\omega}} \end{aligned} \tag{13}$$

Note that the definition of $\mathbf{v}$ is tautological, while the definition of $\hat{\mathbf{v}}$ defines the sense in which $\hat{\mu} \approx 1$. We approximate these as $\tilde{\mu}(\hat{\boldsymbol{\omega}}) = \tilde{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\hat{\boldsymbol{\omega}})$ and $\tilde{\tilde{\mu}}(\hat{\boldsymbol{\omega}}) = \tilde{\tilde{\mathbf{v}}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\hat{\boldsymbol{\omega}})$, respectively, where, recalling that $u_i = \mu(\hat{\boldsymbol{\omega}}_i)$:

$$\begin{aligned} \tilde{\mathbf{v}} &= \operatorname*{argmin}_{\tilde{\mathbf{v}}\in\mathbb{R}^m} \tfrac{1}{2}\textstyle\sum_i\big(\tilde{\mathbf{v}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\hat{\boldsymbol{\omega}}_i) - u_i\big)^2 + \tfrac{\gamma}{2}\|\tilde{\mathbf{v}}\|_2^2 \\ \tilde{\tilde{\mathbf{v}}} &= \operatorname*{argmin}_{\tilde{\tilde{\mathbf{v}}}\in\mathbb{R}^m} \tfrac{1}{2D}\textstyle\sum_i\Big(\tilde{\tilde{\mathbf{v}}}^{\mathrm{T}}\boldsymbol{\varphi}_\kappa(\hat{\boldsymbol{\omega}}_i) - 1\Big)^2 + \tfrac{\gamma}{2}\|\tilde{\tilde{\mathbf{v}}}\|_2^2 \end{aligned} \tag{14}$$

That is, we replace μ and $\hat{\mu}$ with the regularised least-squares approximations obtained from the training sets $\{(\hat{\boldsymbol{\omega}}_i, u_i) : i \in \mathbb{N}_D\}$ and $\{(\hat{\boldsymbol{\omega}}_i, 1) : i \in \mathbb{N}_D\}$, respectively, where the regularisation terms are included to ensure uniform convergence $\tilde{\mu} \to \mu$, $\tilde{\tilde{\mu}} \to \hat{\mu}$ in the limit $D \to \infty$. Hence:

$$\begin{aligned} \tilde{\mathbf{v}} &= \Big(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \gamma D\mathbf{I}\Big)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{u} \\ \tilde{\tilde{\mathbf{v}}} &= \Big(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \gamma D\mathbf{I}\Big)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{1} \end{aligned}$$

where $\boldsymbol{\Phi} = [\varphi_{\kappa j}(\hat{\boldsymbol{\omega}}_i)]_{ij}$. Subsequently, using the Woodbury matrix identity, $(\tilde{\mathbf{v}} - \tilde{\tilde{\mathbf{v}}})^{\mathrm{T}}(\tilde{\mathbf{v}} - \tilde{\tilde{\mathbf{v}}}) = \mathbf{u}^{\mathrm{T}}\mathbf{H}_\gamma\mathbf{u}$, where $\mathbf{H}_\gamma = (\boldsymbol{\Gamma} + \gamma D\mathbf{I})^{-1}\boldsymbol{\Gamma}(\boldsymbol{\Gamma} + \gamma D\mathbf{I})^{-1}$ and $\boldsymbol{\Gamma} = \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}} = [\kappa(\hat{\boldsymbol{\omega}}_i, \hat{\boldsymbol{\omega}}_j)]_{i,j}$. Recalling that the RKHS norm $\|\cdot\|_{\mathcal{H}_\kappa}$ corresponds to the Euclidean norm in feature space, we see that $\|\mu - \hat{\mu}\|^2_{\mathcal{H}_\kappa} = \|\mathbf{v} - \hat{\mathbf{v}}\|_2^2 \approx \|\tilde{\mathbf{v}} - \tilde{\tilde{\mathbf{v}}}\|_2^2$, and so we approximate:

$$\begin{aligned} &\|\mu - \hat{\mu}\|^2_{\mathcal{H}_\kappa} \approx (\mathbf{u} - \mathbf{1})^{\mathrm{T}}\mathbf{H}_\gamma(\mathbf{u} - \mathbf{1}) \\ &\text{where:}\quad \mathbf{H}_\gamma = (\boldsymbol{\Gamma} + \gamma D\mathbf{I})^{-1}\boldsymbol{\Gamma}(\boldsymbol{\Gamma} + \gamma D\mathbf{I})^{-1} \\ &\qquad\quad \boldsymbol{\Gamma} = [\kappa(\hat{\boldsymbol{\omega}}_i, \hat{\boldsymbol{\omega}}_j)]_{i,j\in\mathbb{N}_D} \end{aligned} \tag{15}$$

With this approximation we may re-write the approximated

modified regularised empirical risk (12) entirely in terms of (finite dimensional) modified random Fourier features. To further simplify let $\mathbf{w} = \mathbf{u} \odot \hat{\boldsymbol{\tau}} = [\eta(\hat{\boldsymbol{\omega}}_i)]_{i \in \mathbb{N}_D}$, so:

$$\boxed{f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = \mathbf{w}^\dagger \hat{\mathbf{z}}(\mathbf{x}) + b} \tag{16}$$

where $\mathbf{w} \in \mathbb{C}^D$ and $b \in \mathbb{C}$ minimise the tuned random features (TRF) objective, substituting (15) into (12):

$$\boxed{\tilde{Q}_N(\mathbf{w}, b) = \tfrac{1}{N}\textstyle\sum_i \ell\big(\mathbf{w}^\dagger \hat{\mathbf{z}}(\mathbf{x}_i) + b, y_i\big) + \tfrac{\lambda}{2}\tilde{r}(\mathbf{w})} \tag{17}$$

where $\tilde{r}$ is a regulariser of the form:

$$\boxed{\tilde{r}(\mathbf{w}) = \min_{\mathbf{u} \in \mathbb{R}_+^D} \mathbf{w}^\dagger \mathrm{diag}(\mathbf{u})^{-1}\mathbf{w} + \tfrac{\Lambda}{\lambda}(\mathbf{u}-\mathbf{1})^{\mathrm{T}}\mathbf{H}_\gamma(\mathbf{u}-\mathbf{1})} \tag{18}$$

Note that (16)-(18) are the random Fourier features approximation of (9)-(11). In the next section we will show that $\tilde{r}$ is in fact convex, and moreover the gradient of $\tilde{r}$ is element-wise positive. Thus when training with gradient-descent (or similar) the effect of $\tilde{r}$ is to apply adaptive regularisation to each compound weight component $\mathbf{w}$.

4 Theoretical Analysis

In this section we analyse the properties of the TRF formulation from a theoretical standpoint. We first analyse the properties of the TRF regulariser function $\tilde{r}$ and demonstrate that it is a convex regularisation function with a straightforward gradient. Subsequently we analyse the Rademacher complexity of the formulation and give bounds to demonstrate uniform convergence both in terms of N (the training set size) and D (the number of random features).

4.1 Properties of the TRF Regulariser $\tilde{r}$

In the previous section we demonstrated how tuned regularised empirical risk minimisation using random Fourier features could be reduced to a regularised empirical risk minimisation problem (17) where the density vector $\mathbf{u}$ only appears in the regulariser $\tilde{r}$ defined by (18). For convenience, we refactor the regulariser $\tilde{r}$ as:

$$\tilde{r}(\mathbf{w}) = \rho(\mathbf{u}^\star(\mathbf{w}); \mathbf{w}) \tag{19}$$

where $\mathbf{u}^\star(\mathbf{w}) = \mathrm{argmin}_{\mathbf{u} \in \mathbb{R}_+^D} \rho(\mathbf{u}; \mathbf{w})$, and:

$$\rho(\mathbf{u}; \mathbf{w}) = \mathbf{w}^\dagger \mathrm{diag}(\mathbf{u})^{-1}\mathbf{w} + \tfrac{\Lambda}{\lambda}(\mathbf{u}-\mathbf{1})^{\mathrm{T}}\mathbf{H}_\gamma(\mathbf{u}-\mathbf{1}) \tag{20}$$

The first term, which will dominate when $\Lambda \ll \lambda$, will tend to *push* the density vector $\mathbf{u}^\star \to \infty$, so $\tilde{r}(\mathbf{w}) \to 0$; while the second term, which will dominate when $\Lambda \gg \lambda$, will tend to *pull* toward $\mathbf{u} = \mathbf{1}$, so $\rho \approx \hat{\rho}$ and hence $K \approx \hat{K}$. Applying first-order optimality conditions we see that these opposing influences cancel out at the optimal $\mathbf{u}^\star = \mathbf{u}^\star(\mathbf{w})$:

$$\nabla_{\mathbf{u}} \rho(\mathbf{u}^\star; \mathbf{w}) = -\left|\mathbf{w} \odot \mathbf{u}^{\star \odot -1}\right|^{\odot 2} + 2\tfrac{\Lambda}{\lambda}\mathbf{H}_\gamma(\mathbf{u}^\star - \mathbf{1}) = \mathbf{0} \tag{21}$$

As shown in the supplementary material:

- The regularisation function $\tilde{r}$ is convex and has gradient:
$$\nabla_{\mathbf{w}} \tilde{r}(\mathbf{w}) = 2\mathbf{w} \odot \mathbf{u}^{\star \odot -1}$$
- The optimal density vector satisfies $\mathbf{u}^\star(\mathbf{w}) \geq \mathbf{1}$.

The convexity of the regulariser $\tilde{r}$ means that, if the loss function ℓ is convex, then so too is the tuned regularised empirical risk $\tilde{Q}_N$, which is helpful for training.

$\boldsymbol{\Gamma}$-Spectrum	Polynomial $\hat{\Delta}_i = \mathcal{O}(Di^{-\nu})$	Exponential $\hat{\Delta}_i = \mathcal{O}(De^{-ci})$
Parameter λ	$\lambda = \Omega(N^{\epsilon-\phi})$ $\lambda = \mathcal{O}(N^{-\epsilon})$ $\lambda = \mathcal{O}(D^{-\frac{1}{2}(\nu+1)})$	$\lambda = \Omega(N^{\epsilon-\phi})$ $\lambda = \mathcal{O}(N^{-\epsilon})$ $\lambda = \mathcal{O}(\frac{1}{\sqrt{D}}e^{-\frac{cD}{2}})$
Parameter Λ	$\Lambda = \Omega(N^{\epsilon-\phi})$ $\Lambda = \mathcal{O}(N^{-\epsilon})$	$\Lambda = \Omega(N^{\epsilon-\phi})$ $\Lambda = \mathcal{O}(N^{-\epsilon})$
Parameter γ	$\gamma = \Omega(D^{\epsilon-1})$ $\gamma = \mathcal{O}(D^{-\epsilon})$	$\gamma = \Omega(D^{\epsilon-1})$ $\gamma = \mathcal{O}(D^{-\epsilon})$

Figure 2: Summary of convergence characteristics for kernels with polynomially and exponentially decaying eigenspectra. In this table $\hat{\Delta}_0 \geq \hat{\Delta}_1 \geq \ldots \geq \hat{\Delta}_{D-1}$ are the eigenvalues of the random feature Gram matrix $\boldsymbol{\Gamma}$ for meta-kernel κ; $0 < \epsilon \ll 1$; $\nu \geq 1$ and $c \geq 1$ are constants characterising κ; $\phi = \frac{1}{10}$ if ℓ is Lipschitz, $\phi = \frac{1}{6}$ if ℓ is quadratic; and the hyper-parameter recommendations are designed to ensure uniform convergence as $N, D \to \infty$.

4.2 Convergence and Complexity

While we formulate TRF as a combination of regularised empirical risk minimisation and kernel learning for a given dataset $\mathtt{D}$, the underlying goal remains to minimise the *actual* risk on the distribution $\mathcal{S}$, where $\mathtt{D} \sim \mathcal{S}^N$. Precisely, defining (using the notation of Figure 3):

$$\tilde{f}^\star = \operatorname*{argmin}_{\tilde{f} \in \mathcal{W}} \tilde{Q}_N(\tilde{f}), \quad f^\star = \operatorname*{argmin}_{f \in \mathcal{H}_K \oplus \mathbb{C}} Q_{\mathcal{S}}(f)$$
$$\text{where} : \mathcal{W} \subset \left\{ \mathbf{w}^\dagger \hat{\mathbf{z}}(\cdot) + b \,\middle|\, \mathbf{w} \in \mathbb{C}^D, b \in \mathbb{C} \right\}$$

we aim to show $\tilde{f}^\star \to f^\star$ with high probability as $N, D \to \infty$ (i.e. uniform convergence (Menon and Williamson 2018; Bartlett and Mendelson 2002)). To this end, as illustrated in Figure 3, we may split the analysis of TRF on to two axis, specifically the asymptotic behaviour as $N \to \infty$, and the asymptotic behaviour as $D \to \infty$. Then:

1. We show that $\tilde{Q}_N(\tilde{f}) \to \tilde{Q}_{\mathcal{S}}(\tilde{f})$ with high probability in the limit $N \to \infty$ for all D, $\tilde{f} \in \mathcal{W}$.
2. We characterise the schedule of hyper-parameters λ, Λ, γ with respect to N, D so that this convergence is guaranteed and $\lambda \to 0$ as $N \to \infty$ for all D, $\tilde{f} \in \mathcal{W}$.
3. We show that $\mathcal{W} \to \mathcal{H}_{\hat{K}} \oplus \mathbb{C}$ as $D \to \infty$.

Combining these results, we find that $\tilde{f}^\star \to f^\star$ with high probability as $N, D \to \infty$; first, as $N \to \infty$, TRF converges to an unregularised RFF approximation of actual risk minimisation, and then, as $D \to \infty$, $\mathcal{W} \to \mathcal{H}_{\hat{K}} \oplus \mathbb{C}$ and noting that $\tilde{Q}_{\mathcal{S}} = Q_{\mathcal{S}}$ (each is independent of the range of $\tilde{f}$ or f), we obtain the desired result.

4.3 Convergence as $N \to \infty$

In this section we derive a bound on the Rademacher complexity $R_N(\mathcal{W})$ of the (hypothesis space $\mathcal{W}$ of the) TRF formulation and derive schedules for λ and Λ such that $\lambda \to 0$ and $R_N(\mathcal{W}) \to 0$ as $N \to \infty$. Given this, it is then trivial

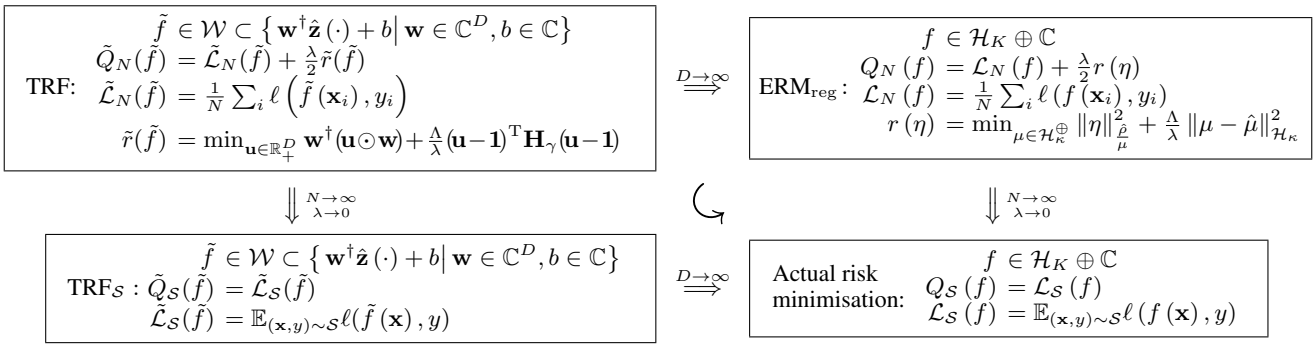

Figure 3: Connection between TRF (top left), tuned regularised risk minimisation (top right), TRF in the limit $N \to \infty$ (bottom left) and actual risk minimisation (bottom right).

use standard Rademacher complexity based uniform convergence bounds (Menon and Williamson 2018; Bartlett and Mendelson 2002) to guarantee that $\tilde{Q}_N(\tilde{f}) \to \tilde{Q}_{\mathcal{S}}(\tilde{f})$ with high probability for a variety of loss function families. We require the following assumptions:

1. Either ℓ is L-Lipschitz and sub-differentiable, or $\ell(\breve{y}, y) = \frac{1}{2}(\breve{y} - y)^2$ is quadratic and $\mathbb{Y} = [-m, m] \subset \mathbb{R}$.
2. $\frac{\lambda}{\Lambda}\hat{\Delta}^{-1/2}_{\gamma,\min} \to C < \infty$ as $D \to \infty$, where $\hat{\Delta}_{\gamma,\min}$ is the minimum eigenvalue of $\mathbf{H}_\gamma$.

A detailed derivation of our complexity bound is given in the supplementary and summarised here. As a first step, we show that $\mathcal{W} \subset \{\mathbf{w}^\dagger\hat{\mathbf{z}}(\cdot) + b : \tilde{r}(\mathbf{w}) \leq R\}$, where (see theorems 11 and 12 in the supplementary):

$$R < a_D\lambda^{-p_a} + b_D\lambda^{-p_b} + c_D\lambda^{-p_c}$$

for positive, finite-valued sequences a_D, b_D, c_D and positive exponents p_a, p_b, p_c; and $p_a < p_b < p_c \leq 5$ in the case where ℓ is Lipschitz, $p_a < p_b < p_c \leq 3$ if ℓ is quadratic; and moreover the Rademacher complexity of $\mathcal{W}$ is bounded as (supplementary, theorem 7):

$$\mathcal{R}_N(\mathcal{W}) \leq \sqrt{R/N} + e_D R/\sqrt{N}$$

where e_D is a positive, finite-valued sequence. Hence $\mathcal{R}_N(\mathcal{W}) \to 0$ as $N \to \infty$ if $\lambda, \Lambda = \Omega(N^{\epsilon-\phi})$ (we require $\frac{\Lambda}{\lambda} \to E < \infty$ as $N \to \infty$) for $0 < \epsilon \ll 1$, and $\phi = \frac{1}{10}$ if ℓ is Lipschitz, and $\phi = \frac{1}{6}$ if ℓ is quadratic.

The proviso $\frac{\lambda}{\Lambda}\hat{\Delta}^{-1/2}_{\gamma,\min} \to C < \infty$ as $D \to \infty$ leads us to analyse the eigenspectrum of $\mathbf{H}_\gamma$ (supplementary, Lemma 5). Denoting the eigenspectrum of the random-feature Gram matrix $\mathbf{\Gamma}$ by $\hat{\Delta}_0 \geq \hat{\Delta}_1 \geq \ldots \geq \hat{\Delta}_{D-1}$, we show that:

$$\hat{\Delta}^{-1}_{\gamma,\min} \leq D\begin{cases} 4\hat{\Delta}_0/D & \text{if } \gamma \leq \gamma_\perp \\ \hat{\Delta}_{D-1}/D + 2\gamma + \gamma^2/(\hat{\Delta}_{D-1}/D) & \text{if } \gamma \geq \gamma_\perp \end{cases}$$

where $\gamma_\perp = \frac{1}{D}(\hat{\Delta}_0\hat{\Delta}_{D-1})^{1/2}$. Then, for example, for kernels with polynomially decaying eigenvalues, we have $\hat{\Delta}_i = \mathcal{O}(Di^{-\nu})$ for some $\nu \geq 1$,[2] so, in this case, $\frac{\lambda}{\Lambda}\hat{\Delta}^{-1/2}_{\gamma,\min} \to C < \infty$ as $D \to \infty$ if $\Lambda = \mathcal{O}(1)$ (w.r.t. D) and:

$$\lambda = \mathcal{O}(1/\sqrt{D}) \text{ if } \gamma \leq \gamma_\perp,\ \lambda = \mathcal{O}(1/\sqrt{D^{\nu+1}}) \text{ otherwise}$$

Finally, we note that $\gamma_\perp = \mathcal{O}(D^{-\frac{1}{2}(\nu+1)})$. We will show in our analysis of the $D \to \infty$ limit that $\gamma = \Omega(1/D)$ is necessary to ensure convergence $\tilde{Q}_N \to Q_N$, so in most cases $\gamma > \gamma_\perp$ for large D, so $\lambda = \mathcal{O}(\frac{1}{\sqrt{D^{\nu+1}}})$ is required to ensure uniform convergence. See Table 2 for a summary of scheduling requirements for hyper-parameters λ, Λ, γ.

4.4 Convergence in μ and $\hat{\mu}$

The role of γ in the TRF formulation is to regularise the approximation of μ and $\hat{\mu}$, defined by (13), via the regularised least-squares approximation $\tilde{\mu}$ and $\tilde{\hat{\mu}}$, defined by (14). Thus the scheduling of γ as $D \to \infty$ is pivotal in determining whether $\tilde{\mu} \to \mu$ and $\tilde{\hat{\mu}} \to \hat{\mu}$ uniformly as $D \to \infty$. For regularised least-squares, it is well-known that $\gamma = \Omega(D^{\epsilon-1})$, $\gamma = \mathcal{O}(D^{-\epsilon})$, where $0 < \epsilon \ll 1$, is sufficient to ensure $\tilde{\mu} \to \mu$ and $\tilde{\hat{\mu}} \to \hat{\mu}$ with high probability as $D \to \infty$. As noted previously, this implies that $\gamma \geq \gamma_\perp$ as $D \to \infty$.

4.5 Convergence as $D \to \infty$

Our goal is to show that $\mathcal{W} \to \mathcal{H}_K \oplus \mathbb{C}$ as $D \to \infty$. Recall that $\hat{K}$ is a translation invariant kernel with a corresponding, strictly positive, even density function $\hat{\rho}$; and likewise K is a translation invariant kernel with a strictly positive and even density function $\rho = \mu(\cdot)\hat{\rho}(\cdot)$, where $\mu \in \mathcal{H}_\kappa^\oplus$. Thus, ignoring the inner-product (which is different for $\mathcal{H}_K$ and $\mathcal{H}_{\hat{K}}$), $\mathcal{H}_K$ and $\mathcal{H}_{\hat{K}}$ actually represent the same set of functions:

$$\begin{aligned}\mathcal{H}_K &= \left\{\langle\eta(\cdot), \xi(\cdot, \mathbf{x})\rangle_\rho\right\} = \left\{\left\langle\tfrac{\rho(\cdot)}{\hat{\rho}(\cdot)}\eta(\cdot), \xi(\cdot, \mathbf{x})\right\rangle_{\hat{\rho}}\right\} \\ &= \left\{\langle\hat{\eta}(\cdot), \xi(\cdot, \mathbf{x})\rangle_{\hat{\rho}}\right\} = \mathcal{H}_{\hat{K}}\end{aligned}$$

[2] For example all ν-times continuously differentiable meta-kernels have polynomially decaying eigenvalues (Wathen and Zhu 2015, Theorem 1), so this includes the RBF kernel, Matérn kernel of order $\geq \frac{3}{2}$ etc. See results in (Braun 2005; Williamson, Smola, and Schölkopf 2001; Wathen and Zhu 2015) for further discussion. In the supplementary we also give results for meta-kernels with exponentially decaying eigenspectra.

so it suffices to show that $\mathcal{W} \to \mathcal{H}_{\hat{K}} \oplus \mathbb{C}$. Moreover, recalling that $\mathcal{H}_{\tilde{K}} = \{\mathbf{w}^\dagger \hat{\mathbf{z}}(\cdot) | \mathbf{w} \in \mathbb{R}^D\}$ with feature map $\hat{\mathbf{z}}$:

$$\mathcal{W} = \left\{\mathbf{w}^\dagger \hat{\mathbf{z}}(\cdot) + b\right\} = \mathcal{H}_{\tilde{K}} \oplus \mathbb{C}$$

so it suffices to show that $\hat{\tilde{K}} \to \hat{K}$ as $D \to \infty$. The convergence properties of RFF kernel approximations has been widely studied (Li et al. 2019b; Rahimi and Recht 2006; Liu et al. 2020; Yang, Sindhwanim, and Mahoney 2014). In particular, (Sutherland and Schneider 2015) present a range of convergence bounds showing that $\|\hat{\tilde{K}} - \hat{K}\|_\infty \to 0$ with high probability as $D \to \infty$, any one of which suffices to show that $\mathcal{W} \to \mathcal{H}_K \oplus \mathbb{C}$ with high probability as $D \to \infty$, and hence, when combined with our previous analysis, that $\tilde{f}^\star \to f^\star$ with high probability as $N, D \to \infty$ so long as λ, Λ and γ are scheduled appropriately (see Table 2).

5 Training Considerations

In this section we consider the question of training. Recall that the regulariser $\tilde{r}$ may be rewritten as $\tilde{r}(\mathbf{w}) = \rho(\mathbf{u}^\star(\mathbf{w}); \mathbf{w})$ as per (19)-(20). In section 4.1 we showed that $\mathbf{u}^\star(\mathbf{w}) \geq \mathbf{1}$ for all $\mathbf{w}$, which allows us to re-write our optimisation problem - that is, minimising $\tilde{Q}_N$ as defined by (17)-(18) - as a constrained minimisation problem including $\mathbf{u}$:

$$\begin{aligned} \underset{\mathbf{w},\mathbf{u},b:\mathbf{u}\geq\mathbf{1}}{\operatorname{argmin}} \; & \tilde{T}_N = \tfrac{1}{N}\textstyle\sum_i \ell\left(\mathbf{w}^\dagger \hat{\mathbf{z}}\left(\mathbf{x}_i\right) + b, y_i\right) + \ldots \\ \ldots + \tfrac{\lambda}{2}\mathbf{w}^\dagger & \mathrm{diag}\left(\mathbf{u}\right)^{-1}\mathbf{w} + \tfrac{\Lambda}{2}\left(\mathbf{u}-\mathbf{1}\right)^{\mathrm{T}}\mathbf{H}_\gamma\left(\mathbf{u}-\mathbf{1}\right) \end{aligned} \tag{22}$$

the gradients (assuming ℓ is differentiable - more generally we may use subgradient methods) of which are:

$$\begin{aligned} \nabla_{\mathbf{w}}\tilde{T}_N &= \tfrac{1}{N}\textstyle\sum_i \ell'\left(\mathbf{w}^\dagger\hat{\mathbf{z}}(\mathbf{x}_i) + b, y_i\right)\hat{\mathbf{z}}(\mathbf{x}_i) + \lambda\mathbf{u}^{\odot-1}\odot\mathbf{w} \\ \nabla_b\tilde{T}_N &= \tfrac{1}{N}\textstyle\sum_i \ell'\left(\mathbf{w}^\dagger\hat{\mathbf{z}}(\mathbf{x}_i) + b, y_i\right) \\ \nabla_{\mathbf{u}}\tilde{T}_N &= \Lambda\mathbf{H}_\gamma\left(\mathbf{u}-\mathbf{1}\right) - \tfrac{\lambda}{2}\left|\mathbf{w}\right|^{\odot 2}\odot\mathbf{u}^{\odot-2} \end{aligned}$$

where we denote by ℓ' the gradient of ℓ with respect to its first argument. As $\mathbf{u} \geq \mathbf{1}$ we see that all three gradients are well-defined and well-behaved, making gradient-based approaches well-suited to the task of minimising $\tilde{T}_N$. We have chosen to use Adam (Kingma and Ba 2015) in our experiments, with an additional clipping step after each iteration to enforce the constraint $\mathbf{u} \geq \mathbf{1}$. An alternative approach is to minimise (17) with a loss-specific algorithm (e.g. Pegasos (Shalev-Shwartz, Singer, and Srebro 2007; Menon and Williamson 2018; Jumutc and Suykens 2013) if ℓ is a hinge loss) and minimise (18) at each step as an *inner loop* using a gradient-based approach (or use bi-quadratic optimisation). However our initial experiments showed that a simple, single-layer gradient-based approach was significantly faster and had a more predictable running time, so we focus on it exclusively.

6 Experimental Results

In this section we present experimental results for TRF applied to classification and regression problems for small and medium sized datasets. For classification we use hinge loss $\ell(\breve{y}, y) = (1 - \breve{y}y)_+$, and for regression $\ell(\breve{y}, y) = \frac{1}{2}(\breve{y} - y)^2$. Experiments were run on a Ubuntu 4.15 server with 72 x86_64 cores and 754 GB of memory running SVMHeavy 8.

Dataset	SVM-RBF	SVM-MKL	FKL	**TRF**
Biodeg	13.9(2.8)	12.3(2.1)	12.0(2.3)	**11.2(3.5)**
Car	1.15(0.4)	0.81(0.2)	0.75(0.1)	**0.73(0.1)**
Contra.	40.1(4.8)	31.6(4.8)	35.0(3.7)	**29.5(3.1)**
Fertility	**9(4.2)**	**9(4.2)**	**9(4.2)**	**9(4.2)**
Ionosph.	5.92(2.5)	4.22(1.4)	**3.99(1.6)**	14.7(7.4)
Sonar	27.1(16)	20.5(2.7)	21.1(4.5)	**15.5(3.2)**
a8a	15.7(1.0)	13.2(2.4)	12.9(2.7)	**12.6(2.7)**

Dataset	SVM-RBF	SVM-MKL	FKL	**TRF**
Airfoil	0.48(0.30)	−(−)	0.46(0.02)	**0.43(0.02)**
Auto	0.18(0.04)	0.17(0.04)	0.17(0.03)	**0.16(0.03)**
Boston	0.43(0.13)	**0.38(0.07)**	**0.38(0.13)**	**0.38(0.14)**
Slump	0.028(0.02)	0.020(0.01)	0.20(0.01)	**0.018(0.01)**
Yatch	0.17(0.07)	**0.05(0.03)**	0.07(0.06)	0.16(0.12)

Figure 4: Classification (top, misclassification error %) and Regression (bottom, RMSE error) results on 20% test set with 5 repeats. TRF is our method, SVM-RBF is the SVM with RBF kernel, and SVM-MKL is the SVM with MKL kernel.

For our TRF method we use an RBF meta-kernel κ with length-scale l, and an RBF reference kernel $\hat{K}$ with fixed length-scale 1. Hyper-parameters were selected to minimise 10-fold cross-validation error on the training set, with $\lambda, \Lambda, \gamma, l \in [0.01, 100]$, using Bayesian optimisation with GP-UCB acquisition function (Srinivas et al. 2012) with a budget of 105 evaluations (5 in the initial random set). This was repeated for $D = 50, 100, 200, 400, 800, 1600$ random features to find D to minimise 10-fold cross validation error.

Our baselines were SVM (ϵ-SV regression and C-SV classification) with RBF and MKL kernel $K_{\mathrm{MKL}} = m_0 K_{\mathrm{RBF}} + m_1 K_{\mathrm{MAT}} + m_2 K_{\mathrm{poly}}$ (K_{RBF}, K_{MAT} and K_{poly} are RBF, $\frac{3}{2}$-Matérn and 2^{nd}-order polynomial kernels, respectively); and FKL (Băzăvan, Li, and Sminchisescu 2012). All hyper-parameters, including C, ϵ (for regression), m_0, m_1, m_2, and the length-scales for K_{RBF} and K_{MAT} were chosen using Bayesian optimisation with GP-UCB acquisition function to minimise 10-fold cross-validation error on the training set.

All datasets are taken from the UCI repository (Dua and Graff 2017), normalised so $\mathbf{x}_i$ lies in the unit hypersphere and split randomly 80% training and 20% testing. All experiments were repeated 5 times to generate error bars. Results for regression and classification are shown in Table 4. Note that our method outperforms the baseline in most cases.

7 Conclusion

We have introduced the tuned random features (TRF) algorithm that combines kernel learning and regularised risk minimisation in the spectral domain, allowing the kernel to be selected automatically from the set of all translation-invariant kernels. We have shown that TRF training may be done via a simple gradient-based approach on the convex objective. We have also analysed the convergence properties of TRF as $N, D \to \infty$, and, using Rademacher complexity analysis, proved that TRF converges uniformly in the limit. Finally, we have demonstrated the effectiveness of TRF on a range of real regression and classification datasets.

Acknowledgments

This research was partially funded by the Australian Government through the Australian Research Council (ARC). Prof Venkatesh is the recipient of an ARC Australian Laureate Fellowship (FL170100006).

References

Avron, H.; Kapralov, M.; Musco, C.; Musco, C.; Velingker, A.; and Zandieh, A. 2017. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *International Conference on Machine Learning*, 253–262.

Avron, H.; Sindhwani, V.; Yang, J.; and Mahoney, M. W. 2016. Quasi-Monte Carlo Feature Maps for Shift-Invariant Kernels. *Journal of Machine Learning Research*, 17(120): 1–38.

Bach, F. 2017. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19): 1–53.

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3: 463–482.

Băzăvan, E. G.; Li, F.; and Sminchisescu, C. 2012. Fourier kernel learning. In *European Conference on Computer Vision*, 459–473. Springer.

Bochner, S. 1932. *Vorlesungen über Fouriersche Integrale*. Leipzig.

Braun, M. L. 2005. *Spectral properties of the kernel matrix and their relation to kernel methods in machine learning*. Ph.D. thesis, Universitäts-und Landesbibliothek Bonn.

Bullins, B.; Zhang, C.; and Zhang, Y. 2017. Not-so-random features. *arXiv preprint arXiv:1710.10230*.

Chang, W.-C.; Li, C.-L.; Yang, Y.; and Poczos, B. 2017. Data-driven random fourier features using stein effect. *arXiv preprint arXiv:1705.08525*.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2012. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13: 795–828.

Cristianini, N.; and Shawe-Taylor, J. 2005. *An Introductino to Support Vector Machines and other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge University Press.

Cristianini, N.; Shawe-Taylor, J.; Elisseeff, A.; and Kandola, J. 2002. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, 367–373.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. *University of California, Irvine, School of Information and Computer Sciences*, http://archive.ics.uci.edu/ml.

Genton, M. G. 2001. Classes of Kernels for Machine Learning: A Statistics Perspective. *Journal of Machine Learning Research*, 2: 299–312.

Geršgorin, S. 1931. Über die Abgrenzung der Eigenwerte einer Matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, 749–754.

Gönen, M.; and Alpaydin, E. 2011. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research*, 12: 2211–2268.

Hastie, T. J.; Tibshirani, R. J.; and Friedman, J. H. 2001. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York: Springer.

Herbrich, R. 2002. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press.

Horn, R. A.; and Johnson, C. R. 2013. *Matrix Analysis*. Cambridge University Press, 2nd edition.

Jumutc, V.; and Suykens, J. A. K. 2013. Weighted coordinate-wise Pegasos. In *Proc. of the 5th International Confence on Pattern Recognition and Machine Intelligence*, volume 8251, 262–269.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015*.

Li, C.-L.; Chang, W.-C.; Mroueh, Y.; Yang, Y.; and Poczos, B. 2019a. Implicit Kernel Learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007–2016.

Li, Z.; Ton, J.-F.; Oglic, D.; and Sejdinovic, D. 2019b. Towards a unified analysis of random Fourier features. In *Proceedings of the 36th International Conference on Machine Learning*, 3905–3914.

Liu, F.; Huang, X.; Chen, Y.; and Suykens, J. A. K. 2020. Random features for kernel approximation: A survey in algorithms, theory, and beyond. *arXiv preprint arXiv:2004.11154*.

Menon, A. K.; and Williamson, R. C. 2018. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, 107–118. PMLR.

Ong, C. S.; Smola, A. J.; and Williamson, R. C. 2005. Learning the Kernel with Hyperkernels. *Journal of Machine Learning Research*, 6: 1043–1071.

Ong, C. S.; Williamson, R. C.; and Smola, A. J. 2003. Hyperkernels. In *Advances in neural information processing systems*, 495–502.

Rahimi, A.; and Recht, B. 2006. Random Features for Large-Scale Kernel Machines. In Platt, J. C.; Koller, D.; Singer, Y.; and Roweis, S. T., eds., *Advances in Neural Information Processing Systems 20*, 1177–1184.

Rasmussen, C. E.; and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Schölkopf, B.; and Smola, A. J. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, Massachusetts: MIT Press. ISBN 0262194759.

Shalev-Shwartz, S.; Singer, Y.; and Srebro, N. 2007. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the 24th International Conference on Machine learning (ICML07)*, 807–814.

Shawe-Taylor, J.; and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Sinha, A.; and Duchi, J. C. 2016. Learning kernels with random features. In *Proceedings of NIPS*.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. W. 2012. Information-Theoretic Regret Bounds for Gaussian

Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5): 3250–3265.

Steinwart, I.; and Christman, A. 2008. *Support Vector Machines*. Springer.

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

Suykens, J. A. K.; Van Gestel, T.; De Brabanter, J.; De Moor, B.; and Vandewalle, J. 2002. *Least Squares Support Vector Machines*. New Jersey: World Scientific Publishing.

Vapnik, V. 1995. *Statistical Learning Theory*. New York: Springer-Verlag.

Wathen, A. J.; and Zhu, S. 2015. On spectral distribution of kernel matrices related to radial basis functions. *Numerical Algorithms*, 70: 709–726.

Williams, C.; and Shawe-taylor, J. S. 2003. The Stability of Kernel Principal Components Analysis and its Relation to the Process Eigenspectrum. In Becker, S.; Thrun, S.; and Obermayer, K., eds., *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, 383–390. MIT Press.

Williamson, R. C.; Smola, A. J.; and Schölkopf, B. 2001. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47(6): 2516–2532.

Wilson, A. G.; and Adams, R. P. 2013. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the International Conference on Machine Learning*, 1067–1075.

Yang, J.; Sindhwanim, H., Vikasand Avron; and Mahoney, M. 2014. Quasi-Monte Carlo Feature Maps for Shift-Invariant Kernels. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, 485–493.

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

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

Zhu, H.; Williams, C. K. I.; Rohwer, R. J.; and Morciniec, M. 1998. Gaussian regression and optimal finite dimensional linear models. *Neural Networks and Machine Learning*.