

Pseudo-Bayesian Learning with Kernel Fourier Transform as Prior

Gaël Letarte, Emilie Morvant, Pascal Germain

▶ **To cite this version:**

Gaël Letarte, Emilie Morvant, Pascal Germain. Pseudo-Bayesian Learning with Kernel Fourier Transform as Prior. The 22nd International Conference on Artificial Intelligence and Statistics, Apr 2019, Naha, Japan. hal-01908555v2

HAL Id: hal-01908555
https://hal.science/hal-01908555v2

Submitted on 19 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Pseudo-Bayesian Learning with Kernel Fourier Transform as Prior

Gaël Letarte[1] gael.letarte.1@ulaval.ca
Emilie Morvant[2] emilie.morvant@univ-st-etienne.fr
Pascal Germain[3] pascal.germain@inria.fr
[1] Département d'informatique et de génie logiciel, Université Laval, Québec, Canada
[2] Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, Saint-Etienne, France
[3] Équipe-projet Modal, Inria Lille - Nord Europe, Villeneuve d'Ascq, France

Abstract

We revisit Rahimi and Recht (2007)'s kernel random Fourier features (RFF) method through the lens of the PAC-Bayesian theory. While the primary goal of RFF is to approximate a kernel, we look at the Fourier transform as a *prior* distribution over trigonometric hypotheses. It naturally suggests learning a *posterior* on these hypotheses. We derive generalization bounds that are optimized by learning a *pseudo-posterior* obtained from a closed-form expression. Based on this study, we consider two learning strategies: The first one finds a compact landmarks-based representation of the data where each landmark is given by a distribution-tailored similarity measure, while the second one provides a PAC-Bayesian justification to the kernel alignment method of Sinha and Duchi (2016).

1 INTRODUCTION

Kernel methods (Shawe-Taylor and Cristianini, 2004), such as support vector machines (Boser et al., 1992; Vapnik, 1998), map data in a high dimension space in which a linear predictor can solve the learning problem at hand. The mapping space is not directly computed and the linear predictor is represented implicitly thanks to a kernel function. This is the powerful kernel trick: the kernel function computes the scalar product between two data points in this high dimension space. However, kernel methods notoriously suffer from two drawbacks. On the first hand, computing all the scalar products for all the learning samples is costly: $O(n^2)$ for many kernel-based methods, where n is the number of training data point. On the other hand, one has to select a kernel function adapted to the learning problem for the algorithm to succeed.

The first of these drawbacks has motivated the development of approximation methods making kernel methods more scalable, such as Nyström approximation (Williams and Seeger, 2001; Drineas and Mahoney, 2005) that constructs a low-rank approximation of the Gram matrix[1] and is data dependent, or random Fourier features (RFF) (Rahimi and Recht, 2007) that approximates the kernel with random features based on the Fourier transform and is not data dependent (a comparison between the two approaches have been conducted by Yang et al., 2012). In this paper, we revisit the latter technique.

We start from the observation that a predictor based on kernel Fourier features can be interpreted as a weighted combination of those features according to a data independent distribution defined by the Fourier transform. We introduce an original viewpoint, where this distribution is interpreted as a *prior distribution* over a space of weak hypotheses—each hypothesis being a simple trigonometric function obtained by the Fourier decomposition. This suggests that one can improve the approximation by adapting this distribution in regards to data points: we aim at learning a *posterior distribution*. By this means, our study proposes strategies to learn a representation to the data. While this representation is not as flexible and powerful than the ones that can be learned by deep neural networks (Goodfellow et al., 2016), we think that it is worthwhile to study this strategy to eventually solve the second drawback of kernel methods that currently heavily rely on the kernel choice. This in mind, while the majority of work related to random Fourier fea-

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

[1] The Gram matrix is the $n \times n$ matrix constituted by all the kernel values computed on the learning samples.

tures focus on the study and improvement of the kernel approximation, we propose here a reinterpretation in the light of the PAC-Bayesian theory (McAllester, 1999; Catoni, 2007). We derive generalization bounds that can be straightforwardly optimized by learning a *pseudo-posterior* thanks to a closed-form expression.

The rest of the paper is organized as follows. Section 2 recalls the RFF setting. Section 3 expresses the Fourier transform as a prior leading *(i)* to a first PAC-Bayesian analysis and a landmarks-based algorithm in Section 4, *(ii)* to another PAC-Bayesian analysis in Section 5 allowing to justify the kernel alignment method of Sinha and Duchi (2016) and to propose a greedy kernel learning method. Then Section 6 provides experiments to show the usefulness of our work.

2 RANDOM FOURIER FEATURES

Problem setting. Consider a classification problem where we want to learn a predictor $f : \mathbb{R}^d \to Y$, from a d-dimensional space to a discrete output space (*e.g.*, $Y = \{0, 1, \ldots, |Y|-1\}$). The learning algorithm is given a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}^n$ of n *i.i.d.* samples, where $\mathcal{D}$ denotes the data generating distribution over $\mathbb{R}^d \times Y$. We consider a positive-semidefinite (PSD) kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to [-1, 1]$. Kernel machines learn predictors of the form

$$f(\mathbf{x}) \;=\; \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x})\,, \tag{1}$$

by optimizing the values of vector $\boldsymbol{\alpha} \in \mathbb{R}^n$.

Fourier features. When n is large, running a kernel machine algorithm (like SVM or kernel ridge regression) is expensive in memory and running time. To circumvent this problem, Rahimi and Recht (2007) introduced the *random Fourier features* as a way to approximate the value of a *shift-invariant kernel*, *i.e.*, relying on the value of $\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}' \in \mathbb{R}^d$, which we write

$$k(\boldsymbol{\delta}) = k(\mathbf{x} - \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$$

interchangeably. Let the distribution $p(\boldsymbol{\omega})$ be the Fourier transform of the shift-invariant kernel k,

$$p(\boldsymbol{\omega}) \;=\; \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} k(\boldsymbol{\delta})\, e^{-i\,\boldsymbol{\omega}\cdot\boldsymbol{\delta}} d\,\boldsymbol{\delta}\,. \tag{2}$$

Now, by writing k as the inverse of the Fourier transform p, and using trigonometric identities, we obtain:

$$\begin{aligned} k(\mathbf{x} - \mathbf{x}') = \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) e^{i\,\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')} d\,\boldsymbol{\omega} \;&=\; \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} e^{i\,\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')} \\ &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \Big[\cos\big(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')\big) + i \sin\big(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')\big)\Big] \\ &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \cos\big(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')\big)\,. \end{aligned} \tag{3}$$

Rahimi and Recht (2007) suggest expressing the above $\cos\big(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')\big)$ as a product of two features. One way to achieve this is to map every input example into

$$\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x}) \;=\; \big(\cos(\boldsymbol{\omega}\cdot\mathbf{x}),\; \sin(\boldsymbol{\omega}\cdot\mathbf{x})\big)\,. \tag{4}$$

The random variable $\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x}) \cdot \mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x}')$, with $\boldsymbol{\omega}$ drawn from p, is an unbiased estimate of $k(\mathbf{x} - \mathbf{x}')$. Indeed, we recover from Equation (3) and Equation (4):

$$\begin{aligned} &\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x}) \cdot \mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x}') \\ &\quad= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \Big[\cos(\boldsymbol{\omega}\cdot\mathbf{x})\cos(\boldsymbol{\omega}\cdot\mathbf{x}') + \sin(\boldsymbol{\omega}\cdot\mathbf{x})\sin(\boldsymbol{\omega}\cdot\mathbf{x}')\Big] \\ &\quad= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \cos\big(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')\big)\,. \end{aligned}$$

To reduce the variance in the estimation of $k(\mathbf{x}-\mathbf{x}')$, the idea is to sample D points *i.i.d.* from p: $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_D$. Then, each training sample $\mathbf{x}_i \in \mathbb{R}^d$ is mapped to a new feature vector in $\mathbb{R}^{2D}$:

$$\boldsymbol{\phi}(\mathbf{x}_i) = \frac{1}{\sqrt{D}}\Big(\cos(\boldsymbol{\omega}_1\cdot\mathbf{x}_i)\,,\;\ldots\,,\;\cos(\boldsymbol{\omega}_D\cdot\mathbf{x}_i)\,,\;\sin(\boldsymbol{\omega}_1\cdot\mathbf{x}_i)\,,\;\ldots\,,\;\sin(\boldsymbol{\omega}_D\cdot\mathbf{x}_i)\Big)\,. \tag{5}$$

Thus, we have $k(\mathbf{x} - \mathbf{x}') \approx \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}')$ when D is "large enough". This provides a decomposition of the PSD kernel k that differs from the classical one (as discussed in Bach, 2017). By learning a linear predictor on the transformed training set $S \mapsto \{(\boldsymbol{\phi}(\mathbf{x}_i), y_i)\}_{i=1}^n$ through an algorithm like a linear SVM, we recover a predictor equivalent to the one learned by a kernelized algorithm. That is, we learn a weight vector $\mathbf{w} = (w_1, \ldots, w_{2D}) \in \mathbb{R}^{2D}$ and we predict the label of a sample $\mathbf{x} \in \mathbb{R}^d$ by computing, in place of Equation (1),

$$f(\mathbf{x}) \;=\; \sum_{j=1}^{2D} w_j\, \phi_j(\mathbf{x})\,. \tag{6}$$

3 THE FOURIER TRANSFORM AS A PRIOR DISTRIBUTION

As described in the previous section, the random Fourier features trick has been introduced to reduce the running time of kernel learning algorithms. Consequently, most of the subsequent work study and/or improve the properties of the kernel approximation (*e.g.*, Yu et al., 2016; Rudi and Rosasco, 2017; Bach, 2017; Choromanski et al., 2018) with some notable exceptions, as the *kernel learning* algorithms of Yang et al. (2015), Sinha and Duchi (2016), and Oliva et al. (2016), that we discuss and relate to our approach in Section 5.

We aim at reinterpreting the Fourier transform—*i.e.*, the distribution p of Equation (2)—as a *prior distribution* over the feature space. It can be seen as an alternative representation of the prior knowledge that is

encoded in the choice of a specific kernel function, that we denote k_p from now on. In accordance with Equation (3), each feature obtained from a vector $\boldsymbol{\omega} \in \mathbb{R}^d$ can be seen as a hypothesis

$$h_{\boldsymbol{\omega}}(\boldsymbol{\delta}) \coloneqq \cos(\boldsymbol{\omega} \cdot \boldsymbol{\delta}) \,.$$

Henceforth, the kernel is interpreted as a predictor performing a p-weighed aggregation of weak hypotheses. This alternative interpretation of distribution p as a prior over hypotheses naturally suggests to *learn a posterior distribution* over the same hypotheses. That is, we seek a distribution q giving rise to a new kernel

$$k_q(\boldsymbol{\delta}) \coloneqq \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} h_{\boldsymbol{\omega}}(\boldsymbol{\delta}) \,.$$

In order to assess the quality of the kernel k_q, we define a loss function based on the consideration that its output should be high when two samples share the same label, and low otherwise. Hence, we evaluate the kernel on two samples $(\mathbf{x}, y) \sim \mathcal{D}$ and $(\mathbf{x}', y') \sim \mathcal{D}$ through the linear loss

$$\ell\big(k_q(\boldsymbol{\delta}), \lambda\big) \;\coloneqq\; \frac{1 - \lambda\, k_q(\boldsymbol{\delta})}{2} \,, \tag{7}$$

where $\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}'$ denotes a pairwise distance and λ denotes the pairwise similarity measure:

$$\lambda = \lambda(y, y') \coloneqq \begin{cases} 1 & \text{if } y = y', \\ -1 & \text{otherwise.} \end{cases}$$

Furthermore, we define the *kernel alignment* generalization loss $\mathcal{L}_\Delta(k_q)$ on a "pairwise" probability distribution Δ, defined over $\mathbb{R}^d \times [-1, 1]$ as

$$\mathcal{L}_\Delta(k_q) \;\coloneqq \mathop{\mathbf{E}}_{(\boldsymbol{\delta},\lambda)\sim\Delta} \ell\big(k_q(\boldsymbol{\delta}), \lambda\big) \,. \tag{8}$$

Note that any data generating distribution $\mathcal{D}$ over input-output spaces $\mathbb{R}^d \times Y$ automatically gives rise to a "pairwise" distribution $\Delta_\mathcal{D}$. By a slight abuse of notation, we write $\mathcal{L}_\mathcal{D}(k_q)$ the corresponding generalization loss, and the associated *kernel alignment* empirical loss is defined as

$$\widehat{\mathcal{L}}_S(k_q) \coloneqq \frac{1}{n(n-1)} \sum_{i,j=1, i\neq j}^{n} \ell\big(k_q(\boldsymbol{\delta}_{ij}), \lambda_{ij}\big) \,, \tag{9}$$

where for a pair of examples $\{(\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j)\} \in S^2$ we have $\boldsymbol{\delta}_{ij} \coloneqq (\mathbf{x}_i - \mathbf{x}_j)$ and $\lambda_{ij} \coloneqq \lambda(y_i, y_j)$.

Starting from this reinterpretation of the Fourier transform, we provide in the rest of the paper two PAC-Bayesian analyses. The first one (Section 4) is obtained by combining n PAC-Bayesian bounds: instead of considering all the possible pairs of data points, we fix one point and we study the generalization ability for all the pairs involving it. The second analysis (Section 5) is based on the fact that the loss can be expressed as a second-order U-statistics.

4 PAC-BAYESIAN ANALYSIS AND LANDMARKS

Due to the linearity of the loss function ℓ, we can rewrite the loss of k_q as the q-average loss of every hypothesis. Indeed, Equation (8) becomes

$$\begin{aligned} \mathcal{L}_\mathcal{D}(k_q) \;&=\; \mathop{\mathbf{E}}_{(\boldsymbol{\delta},\lambda)\sim\Delta_\mathcal{D}} \ell\Big(\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} h_{\boldsymbol{\omega}}(\boldsymbol{\delta}), \lambda\Big) \\ &=\; \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} \mathop{\mathbf{E}}_{(\boldsymbol{\delta},\lambda)\sim\Delta_\mathcal{D}} \ell(h_{\boldsymbol{\omega}}(\boldsymbol{\delta}), \lambda) \;=\; \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} \mathcal{L}_\mathcal{D}(h_{\boldsymbol{\omega}}) \,. \end{aligned}$$

The above q-expectation of losses $\mathcal{L}_\mathcal{D}(h_{\boldsymbol{\omega}})$ turns out to be the quantity bounded by most PAC-Bayesian generalization theorems (sometimes referred as the *Gibbs risk* in the literature), excepted that such results usually apply to the loss over samples instead of distances. Hence, we use PAC-Bayesian bounds to obtain generalization guarantees on $\mathcal{L}_\mathcal{D}(k_q)$ from its empirical estimate of Equation (9), that we can rewrite as

$$\widehat{\mathcal{L}}_S(k_q) = \frac{1}{n^2 - n} \sum_{i,j=1; i\neq j}^{n} \ell\Big(\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} h_{\boldsymbol{\omega}}(\boldsymbol{\delta}), \lambda_{ij}\Big) = \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} \widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}}) \,.$$

However the *classical* PAC-Bayesian theorems cannot be applied directly to bound $\mathcal{L}_\mathcal{D}(k_q)$, as the empirical loss $\widehat{\mathcal{L}}_S(k_q)$ would require to be computed from *i.i.d.* observations of $\Delta_\mathcal{D}$. Instead, the empirical loss involves *dependent* samples, as it is computed from $n^2 - n$ pairs formed by n elements from $\mathcal{D}$.

4.1 First Order KL-Bound

A straightforward approach to apply *classical* PAC-Bayesian results is to bound separately the loss associated with each training sample. That is, for each $(\mathbf{x}_i, y_i) \in S$, we define

$$\mathcal{L}^i_\mathcal{D}(h_{\boldsymbol{\omega}}) \coloneqq \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell\Big(h_{\boldsymbol{\omega}}(\mathbf{x}_i - \mathbf{x}), \lambda(y_i, y)\Big) \,, \tag{10}$$

$$\text{and } \widehat{\mathcal{L}}^i_S(h_{\boldsymbol{\omega}}) \coloneqq \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} \ell\Big(h_{\boldsymbol{\omega}}(\mathbf{x}_i - \mathbf{x}_j), \lambda(y_i, y_j)\Big) \,.$$

Thus, the next theorem gives a generalization guarantee on $\mathcal{L}^i_\mathcal{D}(k_q)$ relying namely on the empirical estimate $\widehat{\mathcal{L}}^i_S(k_q)$ and the Kullback-Leibler divergence $\mathrm{KL}(q\|p) = \mathbf{E}_{\boldsymbol{\omega}\sim q} \ln \frac{q(\boldsymbol{\omega})}{p(\boldsymbol{\omega})}$ between the prior p and the learned posterior q. Note that the statement of Theorem 1 is obtained straightforwardly from Alquier et al. (2016, Theorem 4.1 and Lemma 1), but can be recovered easily from Lever et al. (2013).

Theorem 1. *For $t > 0$, $i \in \{1, \ldots, n\}$, and a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$ over the choice of $S \sim \mathcal{D}^n$, we have for all q on $\mathbb{R}^d$:*

$$\mathcal{L}^i_\mathcal{D}(k_q) \le \widehat{\mathcal{L}}^i_S(k_q) + \frac{1}{t}\left(\mathrm{KL}(q\|p) + \frac{t^2}{2(n-1)} + \ln\frac{1}{\varepsilon}\right) .$$

By the union bound, and using the fact that $\mathcal{L}_{\mathcal{D}}(k_q) = \mathbf{E}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}}\,\mathcal{L}^i_{\mathcal{D}}(k_q)$, we prove the following corollary in the supplementary material.

Corollary 2. *For $t>0$ and a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$ over the choice of $S\sim\mathcal{D}^n$, we have for all q on $\mathbb{R}^d$:*

$$\mathcal{L}_{\mathcal{D}}(k_q) \leq \widehat{\mathcal{L}}_S(k_q)+\frac{2}{t}\left(\mathrm{KL}(q\|p)+\frac{t^2}{2(n-1)}+\ln\frac{n+1}{\varepsilon}\right).$$

Pseudo-Posterior for KL-bounds. Since the above result is valid for any distribution q, one can compute the bound for any learned posterior distribution. Note that the bound promotes the minimization of a trade-off—parameterized by a constant t—between the empirical loss $\widehat{\mathcal{L}}_S(k_q)$ and the KL-divergence between the prior p and the posterior q:

$$\widehat{\mathcal{L}}_S(k_q)+\frac{2}{t}\,\mathrm{KL}(q\|p)\,.$$

It is well-known that for fixed t, p and S, the minimum bound value is obtained with the *pseudo-Bayesian* posterior q^*, such that for $\boldsymbol{\omega}\in\mathbb{R}^d$,

$$q^*(\boldsymbol{\omega}) \;=\; \frac{1}{Z}\,p(\boldsymbol{\omega})\,\exp\left(-\tau\,\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}})\right), \qquad (11)$$

where $\tau := \frac{1}{2}t$ and Z is a normalization constant.[2] Note also Corollary 2's bound converges to the generalization loss $\mathcal{L}_{\mathcal{D}}(k_q)$ at rate $O\left(\sqrt{\frac{\ln n}{n}}\right)$ for the parameter choice $t=\sqrt{n\ln n}$.

Due to the continuity of the feature space, the pseudo-posterior of Equation (11) is hard to compute. To estimate it, one may make use of Monte Carlo (*e.g.*, Dalalyan and Tsybakov, 2012) or variational Bayes methods (*e.g.*, Alquier et al., 2016). In this work, we explore a simpler method: we work solely from a discrete probability space.

4.2 Landmarks-Based Learning

We now propose to leverage on the fact that Theorem 1 bounds the kernel function for the distances to a single data point, instead of learning a kernel globally for every data point as in Corollary 2. We thus aim at learning a collection of kernels (which we can also interpret as similarity functions) for a subset of the training points. We call *landmarks* these training points. The aim of this approach is to learn a new representation of the input space, mapping the datapoints into compact feature vectors, from which we can learn a simple predictor.

Concretely, along with the learning sample S of n examples *i.i.d.* from $\mathcal{D}$, we consider a landmarks sample $L=\{(\mathbf{x}_l,y_l)\}_{l=1}^{n_L}$ of n_L points *i.i.d.* from $\mathcal{D}$, and a *prior* Fourier transform distribution p. For each landmark $(\mathbf{x}_l,y_l)\in L$, let sample D points from p, denoted $\Omega^L=\{\boldsymbol{\omega}^l_m\}_{m=1}^D\sim p^D$. Then, consider a uniform distribution P on the discrete hypothesis set Ω^L, such that $P(\boldsymbol{\omega}^l_m)=\frac{1}{D}$ and $h^l_m(\boldsymbol{\delta}):=\cos(\boldsymbol{\omega}^l_m\cdot\boldsymbol{\delta})$. We aim at learning a set of kernels $\{\widehat{k}_{Q^l}\}_{l=1}^{n_L}$, where each $\widehat{k}_{Q^l}$ is obtained from a distinct $\mathbf{x}_l\in L$ with a fixed parameter $\beta>0$, by computing the pseudo-posterior distribution Q^l given by

$$Q^l_m \;=\; \frac{1}{Z_l}\,\exp\left(-\beta\sqrt{n}\,\widehat{\mathcal{L}}^l_S(h^l_m)\right), \qquad (12)$$

for $m=1,\ldots,D$; Z_l being the normalization constant. Note that Equation (12) gives the minimum of Theorem 1 with $t=\beta\sqrt{n}$. That is, $\beta=1$ corresponds to the regime where the bound converges. Moreover, similarly to Corollary 2, generalization guarantees are obtained simultaneously for the n_L computed distributions thanks to the union bound and Theorem 1. Thus, with probability $1-\varepsilon$, for all $\{Q^l\}_{l=1}^{n_L}$:

$$\mathcal{L}^l_{\mathcal{D}}(\widehat{k}_{Q^l}) \leq \mathcal{L}_S(\widehat{k}_{Q^l})+\frac{1}{t}\left(\mathrm{KL}(Q^l\|P)+\frac{t^2}{2(n-1)}+\ln\frac{n_L}{\varepsilon}\right),$$

where $\mathrm{KL}(Q^l\|P)=\ln D+\sum_{j=1}^D Q^l_j\ln Q^l_j$.

Once all pseudo-posterior are computed thanks to Equation (12), our landmarks-based approach is to map samples $\mathbf{x}\in\mathbb{R}^d$ to n_L similarity features:

$$\boldsymbol{\psi}(\mathbf{x}) := \left(\widehat{k}_{Q^1}(\mathbf{x}_1-\mathbf{x}),\ldots,\widehat{k}_{Q^{n_L}}(\mathbf{x}_{n_L}-\mathbf{x})\right), \qquad (13)$$

and to learn a linear predictor on the transformed training set. Note that, this mapping is not a kernel map anymore and is somehow similar to the mapping proposed by Balcan et al. (2008b,a); Zantedeschi et al. (2018) for a similarity function that is more general than a kernel but fixed for each landmark.

5 LEARNING KERNEL (REVISITED)

In this section, we present PAC-Bayesian theorems that directly bound the *kernel alignment* generalization loss $\mathcal{L}_{\mathcal{D}}(k_q)$ on a "pairwise" probability distribution $\Delta_{\mathcal{D}}$—as defined by Equation (8)—even if the empirical loss $\widehat{\mathcal{L}}_{\mathcal{D}}(k_q)$ is computed on dependent samples. These bounds suggest a *kernel alignment* (or *kernel learning*) strategy similar to the one of Sinha

[2] This trade-off is the same one involved in some other PAC-Bayesian bounds for *i.i.d.* data (*e.g.*, Catoni, 2007). As discussed in Zhang (2006); Grünwald (2012); Germain et al. (2016), there is a similarity between the minimization of such PAC-Bayes bounds and the Bayes update rule.

and Duchi (2016). We stress that our guarantees hold solely for the kernel alignment loss, but not for the predictor trained with this kernel. Hence, our proposed algorithm learns a kernel independently of the prediction method to be used downstream. This is in contrast with the *one-step* frameworks of Yang et al. (2015) and Oliva et al. (2016), which learn a mixture of random kernel features in a *fully* Bayesian way; they rely on a data-generating model, whereas our approach assumes only that the observations are *i.i.d.*

5.1 Second Order KL-bound

The following result is based on the fact that $\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}}) := \frac{1}{n^2-n}\sum_{i\neq j}^{n} \ell(h_{\boldsymbol{\omega}}(\boldsymbol{\delta}_{ij}), \lambda_{ij})$ is an *unbiased* second-order estimator of $\mathbf{E}_{(\boldsymbol{\delta},\lambda)\sim\Delta_{\mathcal{D}}} \ell(h_{\boldsymbol{\omega}}(\boldsymbol{\delta}), \lambda)$, allowing us to build on the PAC-Bayesian analysis for U-statistics of Lever et al. (2013, Theorem 7). Indeed, the next theorem gives a generalization guarantee on the kernel alignment loss $\mathcal{L}_{\mathcal{D}}(k_q)$.

Theorem 3 (Lever et al. 2013)**.** *For $t > 0$ and a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$ over the choice of $S \sim \mathcal{D}^n$, we have for all q on $\mathbb{R}^d$:*

$$\mathcal{L}_{\mathcal{D}}(k_q) \leq \widehat{\mathcal{L}}_S(k_q) + \frac{1}{t}\left(\mathrm{KL}(q\|p) + \frac{t^2}{2n} + \ln\frac{1}{\epsilon}\right).$$

Except for some constant terms, the above Theorem 3 is similar to Corollary 2. Indeed, both are minimized by the same pseudo-posterior q^* (Equation 11, with $\tau := t$ for Theorem 3). Interestingly, we get rid of the $\ln(n+1)$ term of Corollary 2, making in Theorem 3's bound to converge at rate $O(\frac{1}{\sqrt{n}})$ when $t = \sqrt{n}$.

5.2 Second Order Bounds for f-Divergences

In the following, we build on a recent result of Alquier and Guedj (2018) to express a new family of PAC-Bayesian bounds for our dependent samples, where the KL term is replaced by other f-divergences.

Given a convex function f such that $f(1)=0$, a f-divergence is given by $D_f(q\|p) := \mathbf{E}_{\boldsymbol{\omega}\sim p} f\left(\frac{q(\boldsymbol{\omega})}{p(\boldsymbol{\omega})}\right)$. The following theorem applies to f-divergences such that $f(x) = x^\mu - 1$.

Theorem 4. *For $\mu > 1$ and a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$ over the choice of $S \sim \mathcal{D}^n$, we have for all q on $\mathbb{R}^d$:*

$$\mathcal{L}_{\mathcal{D}}(k_q) \leq \widehat{\mathcal{L}}_S(k_q) + \begin{cases} \left(\frac{1}{2\sqrt{n}}\right)^{\mu-1}\left(D_\mu(q\|p)+1\right)^{\frac{1}{\mu}}\left(\frac{1}{\varepsilon}\right)^{1-\frac{1}{\mu}} & \text{if } 1<\mu\leq 2, \\ \left(\frac{1}{4n}\right)^{1-\frac{1}{\mu}}\left(D_\mu(q\|p)+1\right)^{\frac{1}{\mu}}\left(\frac{1}{\varepsilon}\right)^{1-\frac{1}{\mu}} & \text{if } \mu>2, \end{cases}$$

where $D_\mu(q\|p) := \mathbf{E}_{\boldsymbol{\omega}\sim p}\left(\frac{q(\boldsymbol{\omega})}{p(\boldsymbol{\omega})}\right)^\mu - 1.$

Proof. Let $\mathcal{M}_\mu := \mathbf{E}_{\boldsymbol{\omega}\sim p}\mathbf{E}_{S'\sim D^n}\left|\mathcal{L}_{\mathcal{D}}(h_{\boldsymbol{\omega}}) - \widehat{\mathcal{L}}_{S'}(h_{\boldsymbol{\omega}})\right|^\mu$.

We start from Alquier and Guedj (2018, Theorem 1):

$$\mathcal{L}_{\mathcal{D}}(k_q) \leq \widehat{\mathcal{L}}_S(k_q) + \left(\frac{\mathcal{M}_\mu}{\varepsilon}\right)^{1-\frac{1}{\mu}}\left(D_\mu(q\|p)+1\right)^{\frac{1}{\mu}}. \quad (14)$$

Let us show $\mathcal{M}_\mu \leq \left(\frac{1}{2\sqrt{n}}\right)^\mu$ for $1 < \mu \leq 2$:

$$\begin{aligned} \mathcal{M}_\mu &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p}\mathop{\mathbf{E}}_{S'\sim D^n}\left[\left(\mathcal{L}_{\mathcal{D}}(h_{\boldsymbol{\omega}}) - \widehat{\mathcal{L}}_{S'}(h_{\boldsymbol{\omega}})\right)^2\right]^{\frac{\mu}{2}} \\ &\leq \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p}\left[\mathop{\mathbf{E}}_{S'\sim D^n}\left(\mathcal{L}_{\mathcal{D}}(h_{\boldsymbol{\omega}}) - \widehat{\mathcal{L}}_{S'}(h_{\boldsymbol{\omega}})\right)^2\right]^{\frac{\mu}{2}} \quad (15) \\ &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p}\left[\mathop{\mathbf{Var}}_{S'\sim D^n}\left(\mathcal{L}_{S'}(h_{\boldsymbol{\omega}})\right)\right]^{\frac{\mu}{2}} \\ &\leq \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p}\left[\frac{1}{4n}\right]^{\frac{\mu}{2}} = \left[\frac{1}{4n}\right]^{\frac{\mu}{2}}. \quad (16) \end{aligned}$$

Line (15) is obtained by Jensen's inequality (since $0 < \frac{\mu}{2} \leq 1$), and the inequality of Line (16) is proven by Lemma 6 of the supplementary material. Note that the latter is based on the Efron-Stein inequality and Boucheron et al. (2013, Corollary 3.2).

The first case of Theorem 4 statement ($1 < \mu \leq 2$) is obtained by inserting Line (16) in Equation (14). The second case ($\mu > 2$) is obtained by upper-bounding $\mathcal{M}_\mu$ by $\mathcal{M}_2 = \frac{1}{4n}$, as $|\mathcal{L}_{\mathcal{D}}(h_{\boldsymbol{\omega}}) - \widehat{\mathcal{L}}_{S'}(h_{\boldsymbol{\omega}})| \leq 1$. □

As a particular case, with $\mu = 2$, we obtain from Theorem 4 a bound that relies on the chi-square divergence $\chi^2(q\|p) = \mathbf{E}_{\boldsymbol{\omega}\sim p}\left(\frac{q(\boldsymbol{\omega})}{p(\boldsymbol{\omega})}\right)^2 - 1$.

Corollary 5. *Given a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$ over the choice of $S \sim \mathcal{D}^n$, we have for all q on $\mathbb{R}^d$:*

$$\mathcal{L}_{\mathcal{D}}(k_q) \leq \widehat{\mathcal{L}}_S(k_q) + \sqrt{\frac{\chi^2(q\|p)+1}{4\,n\,\varepsilon}}.$$

It is noteworthy that the above result looks alike other PAC-Bayesian bounds based on the chi-square divergence in the *i.i.d.* setting, as the one of Honorio and Jaakkola (2014, Lemma 7), Bégin et al. (2016, Corollary 10) or Alquier and Guedj (2018, Corollary 1). Interestingly, the latter has been introduced to handle unbounded (possibly heavy-tailed) losses, and one could also extend our Corollary 5 to this setting.

5.3 PAC-Bayesian Interpretation of Kernel Alignment Optimization

Sinha and Duchi (2016) propose a kernel learning algorithm that weights random kernel features. To do so, their algorithm solves a *kernel alignment* problem. As explained below, this method is coherent with the PAC-Bayesian theory exposed by our current work.

Kernel alignment algorithm. Let us consider a Fourier transform distribution p, from which N points are sampled, denoted $\Omega = \{\boldsymbol{\omega}_m\}_{m=1}^N \sim p^N$. Then, consider a uniform distribution P on the discrete hypothesis set Ω, such that $P(\boldsymbol{\omega}_m) = \frac{1}{N}$ and $h_m(\boldsymbol{\delta}) \coloneqq \cos(\boldsymbol{\omega}_m \cdot \boldsymbol{\delta})$. Given a dataset S, and constant parameters $\mu > 1$, $\rho > 0$, the optimization algorithm proposed by Sinha and Duchi solves the following problem.

$$\underset{Q \in \mathbb{R}_+^N}{\text{maximize}} \sum_{i=1}^n \sum_{j=1}^n \lambda_{ij} \sum_{m=1}^N Q_m h_m(\boldsymbol{\delta}_{ij}) \,, \tag{17}$$

$$\text{such that} \sum_{m=1}^N Q_m = 1 \text{ and } D_\mu(Q\|P) \leq \rho \,. \tag{18}$$

The iterative procedure proposed by Sinha and Duchi finds an ϵ-suboptimal solution to the above problem in $O(N \log(\frac{1}{\epsilon}))$ steps. The solution provides a learned kernel $\widehat{k}_Q(\boldsymbol{\delta}) \coloneqq \frac{1}{N} \sum_{m=1}^N Q_m h_m(\boldsymbol{\delta})$.

Sinha and Duchi propose to use the above alignment method to reduce the number of features needed compared to the classical RFF procedure (as described in Section 2). Albeit this method is a kernel learning one, empirical experiments show that with a large number of random features, the classical RFF procedure achieves as good prediction accuracy. However, one can draw (with replacement) $D < N$ features from Ω according to Q. For a relatively small D, learning a linear predictor on the random feature vector (such as the one presented by Equation 5) obtained from Q achieves better results than the classical RFF method on the same number D of random features.

PAC-Bayesian interpretation. The optimization problem of Equations (17–18) deals with the same trade-off as the one promoted by Theorem 4. Indeed, maximizing Equation (17) amounts to minimizing $\widehat{\mathcal{L}}_S(k_q)$, and the constraint of Equation (18) controls the f-divergence $D_\mu(Q\|P)$, which is the same complexity measure involved in Theorem 4. Furthermore, the empirical experiments performed by Sinha and Duchi (2016) focus on the χ^2-divergence (case μ=2), which corresponds to tackling the trade-off expressed by Corollary 5.

5.4 Greedy Kernel Learning

The method proposed by Sinha and Duchi (2016) can easily be adapted to minimize the bound of Theorem 3 instead of the bound of Theorem 4. We describe this kernel learning procedure below.

Given a Fourier transform prior distribution p, let sample N points $\Omega = \{\boldsymbol{\omega}_m\}_{m=1}^N \sim p^N$. Let $P(\boldsymbol{\omega}_m) = \frac{1}{N}$ and $h_m(\boldsymbol{\delta}) \coloneqq \cos(\boldsymbol{\omega}_m \cdot \boldsymbol{\delta})$. Given a dataset S, and constant parameters $\beta > 0$, compute the following pseudo-posterior for $m = 1, \ldots, N$:

$$Q_m = \frac{1}{Z} \exp\Big(-\beta \sqrt{n}\, \widehat{\mathcal{L}}_S(h_m) \Big) \,. \tag{19}$$

Then, we sample with replacement $D < N$ features from Ω according to the pseudo-posterior Q. The sampled features are used to map every $\mathbf{x} \in \mathbb{R}^d$ of the training set into a new vector $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{2D}$ according to Equation (5). The latter transformed dataset is then given as input to a linear learning procedure.

In summary, this learning method is strongly inspired by the one described in Section 5.3, but the posterior computation phase is faster, as we benefit from a closed-form expression (Equation 19). Once $\widehat{\mathcal{L}}_S(h_m)$ is computed for all h_m,[3] we can vary the parameter β and get a new posterior in $O(N)$ steps.

6 EXPERIMENTS

All experiments use a Gaussian (*a.k.a.* RBF) kernel of variance σ^2: $k_\sigma(\mathbf{x}, \mathbf{x}') = \exp\big(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}'\|^2 \big)$, for which the Fourier transform is given by

$$p_\sigma(\boldsymbol{\omega}) = \left(\frac{\sigma^2}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{1}{2}\sigma^2 \|\boldsymbol{\omega}\|^2} = \mathcal{N}(\boldsymbol{\omega}; \mathbf{0}, \tfrac{1}{\sigma^2}\mathbf{I}) \,. \tag{20}$$

Apart from the toy experiment of Figure 1, the experiments on real data are conducted by splitting the available data into a training set, a validation set and a test set. The kernel parameter σ is chosen among $\{10^{-7}, 10^{-6}, \ldots, 10^2\}$ by running an RBF SVM on the training set and keeping the parameter having the best accuracy score on the validation set. That is, this σ defines the prior distribution given by Equation (20) for all our pseudo-Bayesian methods. Unless otherwise specified, all the other parameters are selected using the validation set. More details about the experimental procedure are given in the supplementary material.

6.1 Landmarks-Based Learning

Toy experiment. To get some insight from the landmarks-based procedure of Section 4.2, we generate a 2D dataset S_{toy}, illustrated by Figure 1. We randomly select five training points L=$\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \subset S_{\text{toy}}$, and compare two procedures, described below.

RBF-Landmarks: Learn a linear SVM on the *empirical kernel map* given by the five RBF kernels centered on L. That is, each $\mathbf{x} \in S_{\text{toy}}$ is mapped such that

$$\mathbf{x} \mapsto \Big(k_\sigma(\mathbf{x}_1, \mathbf{x}), k_\sigma(\mathbf{x}_2, \mathbf{x}), k_\sigma(\mathbf{x}_3, \mathbf{x}), k_\sigma(\mathbf{x}_4, \mathbf{x}), k_\sigma(\mathbf{x}_5, \mathbf{x}) \Big) \,.$$

[3] We show in supplementary material (section A.2) that each $\widehat{\mathcal{L}}_S(h_m)$ can be computed in $O(n)$ steps.

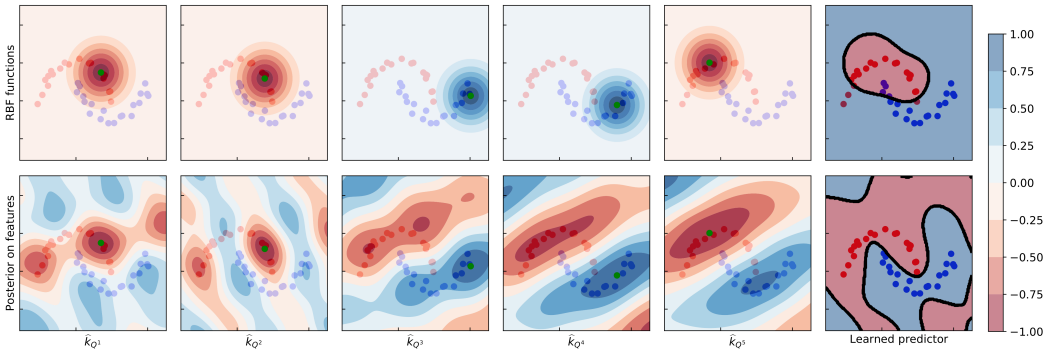


Figure 1: First row shows selected RBF-Landmarks kernel outputs, while second row shows the corresponding learned similarity measures on random Fourier features (PB-Landmarks). The rightmost column displays the classification learned by a linear SVM over the mapped dataset.

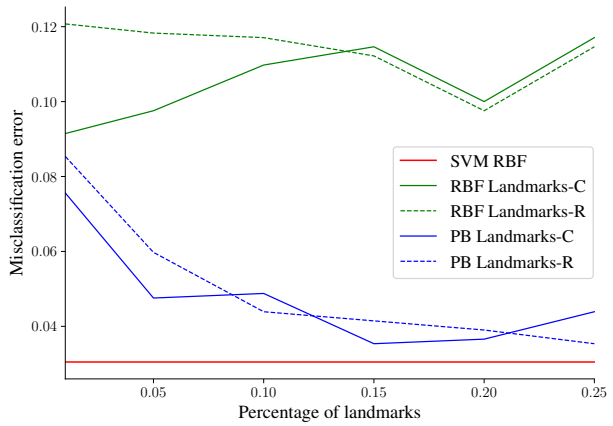


Figure 2: Behavior of the Landmarks-based approach according to the percentage of training points selected as landmarks on the dataset "ads".

Dataset	SVM	Landmarks-based			
		RBF	PB	$\text{PB}_{\beta=1}$	$\text{PB}_{D=64}$
ads	3.05	10.98	**4.88**	5.12	5.00
adult	19.70	19.60	**17.99**	**17.99**	**17.99**
breast	4.90	6.99	3.50	3.50	**2.80**
farm	11.58	17.47	15.73	**14.19**	15.73
mnist17	0.34	0.74	0.42	**0.32**	**0.32**
mnist49	1.16	2.26	**1.80**	2.09	2.50
mnist56	0.55	**0.97**	1.06	1.55	1.03

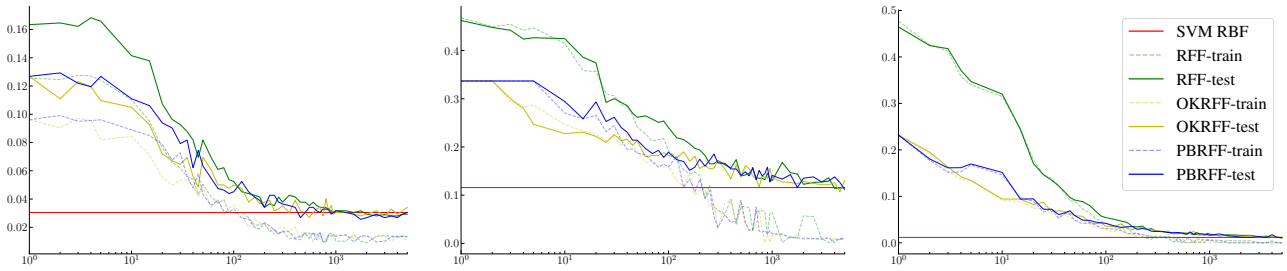
Table 1: Test error of the landmarks-based approach.

PB-Landmarks: Generate 20 random samples according to the Fourier transform of Equation (20). For every landmark of L, learn a *similarity measure* thanks to Equation (12) (with $\beta = 1$), minimizing the PAC-Bayesian bound. We thus obtain five posterior distributions Q^1, Q^2, Q^3, Q^4, Q^5, and learn a linear SVM on the mapped training set obtained by Equation (13).

Hence, the RBF-Landmarks method corresponds to the prior, from which we learn a posterior by landmarks by the PB-Landmarks procedure. Right-most plots of Figure 1 show that the PB-Landmarks setting successfully finds a representation from which the linear SVM can predict well.

Experiments on real data. We conduct similar experiments as the above one on seven real binary classification datasets. Figure 2 studies the behavior of the approaches according to the number of selected landmarks. We select a percentage of the training points as landmarks (from 1% to 25%), and we compare the classification error of a linear SVM on the mapping obtained by the original RBF functions (as in the RBF-Landmarks method above), with the mapping obtained by learning the landmarks posterior distributions (PB-Landmark method). We also compare the case where the landmarks are selected at random among the training data (curves postfixed "-R"), to another scenario where we use the centroids obtained with a k-Means clustering as landmarks (curves postfixed "-C"). Note that, the latter case is not rigorously backed by our PAC-Bayesian theorems, since the choice of landmarks is now dependent of the whole observed training set. The results show that the classification error of both cases are similar, but the clustering strategy leads to a more stable behavior, probably since the landmarks are more representative of the original space. Moreover, the pseudo-Bayesian method improves the results on almost all datasets.

Table 1 compares the error rate of an SVM (trained along with the full Gram matrix and a properly selected σ on the validation set) with four landmarks-based approaches: *(RBF)* the landmarks are RBF kernel of parameter σ; *(PB)* the PB-Landmarks approach where the number of features per landmarks D and



(a) ads (b) farm (c) mnist49

Figure 3: Train and test error of the kernel learning approaches according to the number of random features D.

the β parameter are selected using the validation set; *($PB_{\beta=1}$)* the PB-Landmarks approach where β=1 is fixed and D is selected by validation; and *($PB_{D=64}$)* the PB-Landmarks approach where D=64 is fixed and β is selected by validation. For all landmarks-based approaches, we select the landmarks by clustering, and use 10% of the training set size as the number of landmarks; we want to study the methods in the regime where it provides relatively compact representations. We observe that learning the posterior improves the RBF-Landmarks (except on "mnist56") and that the validation of both β and D parameters are not mandatory to obtain satisfactory results. The SVM RBF is better than all landmarks-based approaches on 4 datasets out of 7, but requires a far less compact representation of the data as it uses the full Gram matrix.

6.2 Greedy Kernel Learning

Figure 3 presents a study of the kernel learning method detailed in Section 5.4, inspired from the one of Sinha and Duchi (2016). We first generate N=20000 random features according to p_σ as given by Equation (4), and we learn a posterior using two strategies: *(OKRFF)* the original optimized kernel of Sinha and Duchi given by Equations (17-18), where ρ is selected on the validation set; and *(PBRFF)* the pseudo-posterior given by Equation (19) where β is selected on the validation set. For both obtained posteriors, we subsample an increasing number of features $D \in [1, 5000]$ to create the mapping given by Equation (5), on which we learn a linear SVM. We also compare to *(RFF)* the standard random Fourier features as described in Section 2, with D randomly selected features according to the prior p_σ.

We see that our PBRFF approach behaves similarly as OKRFF, with a slight advantage for the latter. However, we recall that computing the posterior of former method is faster. Both kernel learning methods have better accuracy than the classical RFF algorithm for a small number of random features, and similar ones for a large number of random features.

7 CONCLUSION & PERSPECTIVES

We elaborated an original viewpoint of the random Fourier features, proposed by Rahimi and Recht (2007) to approximate a kernel. By looking at the Fourier transform as a prior distribution over trigonometric functions, we present two kinds of generalization theorems that bound a kernel alignment loss. Based on classical first-order PAC-Bayesian results, we derived a landmarks-based strategy that learns a compact representation of the data. Then, we proposed two second-order generalization bounds. The first one is based on the U-statistic theorem of Lever et al. (2013). The second one is a new PAC-Bayesian theorem for f-divergences (replacing the usual KL-divergence term). We show that the latter bound provides a theoretical justification to the kernel alignment method of Sinha and Duchi (2016), and we also empirically evaluate a similar but simpler algorithm where the alignment distribution is obtained by the PAC-Bayesian pseudo-posterior closed-form expression.

Our current guarantees hold solely for the kernel alignment loss, and not for the predictor trained with this kernel. An important research direction is to extend the guarantees to the final predictor, which could in turn be the bedrock of a new one-step learning procedure (in the vein of Yang et al., 2015; Oliva et al., 2016). Other research directions include the study of the RKHS associated with the learned kernel, and the extension of our study to wavelet transforms (Mallat, 2008). Furthermore, considering the Fourier transform of a kernel as a (pseudo-)Bayesian prior might lead to other original contributions. Among them, we foresee new perspectives on representation and metric learning, namely for unsupervised learning.

Acknowledgments. P. Germain wants to thank Francis Bach for insightful preliminary discussions. This work was supported in part by the French Project APRIORI ANR-18-CE23-0015 and in part by NSERC. This research was enabled in part by support provided by Compute Canada (www.computecanada.ca).

References

Pierre Alquier and Benjamin Guedj. Simpler PAC-Bayesian bounds for hostile data. *Machine Learning*, 107(5), 2018.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *Journal of Machine Learning Research*, 17, 2016.

Francis R. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18, 2017.

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. Improved guarantees for learning via similarity functions. In *COLT*, 2008a.

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1-2):89–112, 2008b.

Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. PAC-Bayesian bounds based on the Rényi divergence. In *AISTATS*, 2016.

Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, 1992.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities : a nonasymptotic theory of independence*. Oxford university press, 2013. ISBN 978-0-19-953525-5.

Olivier Catoni. *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*, volume 56. Inst. of Mathematical Statistic, 2007.

Krzysztof Choromanski, Mark Rowland, Tamás Sarlós, Vikas Sindhwani, Richard E. Turner, and Adrian Weller. The geometry of random features. In *AISTATS*, 2018.

Arnak S. Dalalyan and Alexandre B. Tsybakov. Sparse regression learning by aggregation and langevin monte-carlo. *J. Comput. Syst. Sci.*, 78(5), 2012.

Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(Dec), 2005.

Pascal Germain, Francis R. Bach, Alexandre Lacoste, and Simon Lacoste-Julien. PAC-Bayesian theory meets Bayesian inference. In *NIPS*, 2016.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Peter Grünwald. The safe Bayesian - learning the learning rate via the mixability gap. In *ALT*, 2012.

Jean Honorio and Tommi S. Jaakkola. Tight bounds for the expected risk of linear classifiers and pac-bayes finite-sample guarantees. In *AISTATS*, 2014.

Guy Lever, François Laviolette, and John Shawe-Taylor. Tighter PAC-Bayes bounds through distribution-dependent priors. *Theor. Comput. Sci.*, 473, 2013.

Stéphane Mallat. *A Wavelet Tour of Signal Processing, 3rd Edition*. Academic Press, 2008.

David McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3), 1999.

Junier B Oliva, Avinava Dubey, Andrew G Wilson, Barnabás Póczos, Jeff Schneider, and Eric P Xing. Bayesian nonparametric kernel-learning. In *AISTATS*, 2016.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *NIPS*, 2017.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Aman Sinha and John C. Duchi. Learning kernels with random features. In *NIPS*, 2016.

Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *NIPS*. 2001.

Tianbao Yang, Yu-feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *NIPS*. 2012.

Zichao Yang, Andrew Gordon Wilson, Alexander J. Smola, and Le Song. A la carte - learning fast kernels. In *AISTATS*, 2015.

Felix X. Yu, Ananda Theertha Suresh, Krzysztof Marcin Choromanski, Daniel N. Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *NIPS*, 2016.

Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. Fast and provably effective multi-view classification with landmark-based svm. In *ECML-PKDD*, 2018.

Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Trans. Information Theory*, 52(4), 2006.

A Supplementary Material

A.1 Mathematical Results

Corollary 2. *For $t > 0$ and a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$ over the choice of $S \sim \mathcal{D}^n$, we have for all q on $\mathbb{R}^d$:*

$$\mathcal{L}_{\mathcal{D}}(k_q) \leq \widehat{\mathcal{L}}_S(k_q) + \frac{2}{t}\left(\mathrm{KL}(q\|p) + \frac{t^2}{2(n-1)} + \ln\frac{n+1}{\varepsilon}\right).$$

Proof. We want to bound

$$\begin{aligned}\mathcal{L}_{\mathcal{D}}(k_q) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\ \mathop{\mathbf{E}}_{(\mathbf{x}',y')\sim\mathcal{D}}\ \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} \ell\Big(h_{\boldsymbol{\omega}}(\mathbf{x}-\mathbf{x}'), \lambda(y,y')\Big)\\ &= \mathop{\mathbf{E}}_{(\mathbf{x}',y')\sim\mathcal{D}} \mathcal{L}'_{\mathcal{D}}(k_q)\,,\end{aligned}$$

where $\mathcal{L}'_{\mathcal{D}}(k_q)$ is the alignment loss of the kernel k_q centered on $(\mathbf{x}', y') \sim \mathcal{D}$ (see Equation (10)).

Let $t > 0$ and p a distribution on $\mathbb{R}^d$. By applying the PAC-Bayesian theorem, with $\varepsilon_0 \in (0,1)$, we have

$$\Pr_{S\sim\mathcal{D}^n}\left(\forall q \text{ on } \mathbb{R}^d : \mathcal{L}_{\mathcal{D}}(k_q) \leq \frac{1}{n}\sum_{i=1}^n \mathcal{L}^i_{\mathcal{D}}(k_q) + \frac{1}{t}\left[\mathrm{KL}(q\|p) + \frac{t^2}{2n} + \ln\frac{1}{\varepsilon_0}\right]\right) \geq 1-\epsilon_0\,.$$

Moreover, we have that for each $i \in \{1,\ldots,n\}$, with a $\varepsilon_i \in (0,1)$, we have

$$\Pr_{S\sim\mathcal{D}^n}\left(\forall q \text{ on } \mathbb{R}^d : \mathcal{L}^i_{\mathcal{D}}(k_q) \leq \widehat{\mathcal{L}}^i_S(k_q) + \frac{1}{t}\left[\mathrm{KL}(q\|p) + \frac{t^2}{2(n-1)} + \ln\frac{1}{\varepsilon_i}\right]\right) \geq 1-\epsilon_i\,.$$

By combining above probabilistic results with $\varepsilon_0 = \varepsilon_1 = \cdots = \varepsilon_n = \frac{\varepsilon}{n+1}$, we obtain that, with probability at least $1-\varepsilon$,

$$\begin{aligned}\mathcal{L}_{\mathcal{D}}(k_q) &= \mathop{\mathbf{E}}_{(\mathbf{x}',y')\sim\mathcal{D}} \mathcal{L}'_{\mathcal{D}}(k_q)\\ &\leq \frac{1}{n}\sum_{i=1}^n\left[\widehat{\mathcal{L}}^i_S(k_q) + \frac{1}{t}\left[\mathrm{KL}(q\|p) + \frac{t^2}{2(n-1)} + \ln\frac{n+1}{\varepsilon}\right]\right] + \frac{1}{t}\left[\mathrm{KL}(q\|p) + \frac{t^2}{2n} + \ln\frac{n+1}{\varepsilon}\right]\\ &= \widehat{\mathcal{L}}_S(k_q) + \frac{1}{t}\left[\mathrm{KL}(q\|p) + \frac{t^2}{2(n-1)} + \ln\frac{n+1}{\varepsilon}\right] + \frac{1}{t}\left[\mathrm{KL}(q\|p) + \frac{t^2}{2n} + \ln\frac{n+1}{\varepsilon}\right]\\ &\leq \widehat{\mathcal{L}}_S(k_q) + \frac{2}{t}\left[\mathrm{KL}(q\|p) + \frac{t^2}{2(n-1)} + \ln\frac{n+1}{\varepsilon}\right].\end{aligned}$$

□

Lemma 6. *For any data-generating distribution $\mathcal{D}$:*

$$\mathop{\mathbf{Var}}_{S'\sim\mathcal{D}^n}\left(\mathcal{L}_{S'}(h_{\boldsymbol{\omega}})\right) \leq \frac{1}{4n}\,.$$

Proof. Given $S' = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}^n$, we denote

$$\mathcal{F}_{\boldsymbol{\omega}}(S') \coloneqq \mathcal{F}_{\boldsymbol{\omega}}\big((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\big) \coloneqq \mathcal{L}_{S'}(h_{\boldsymbol{\omega}}) = \frac{1}{n(n-1)}\sum_{i\neq j}^n \ell\Big(h_{\boldsymbol{\omega}}(\mathbf{x}_i-\mathbf{x}_j), \lambda(y_i,y_j)\Big)\,.$$

The function $\mathcal{F}_{\boldsymbol{\omega}}$ above has the *bounded differences property*. That is, for each $i \in \{1,\ldots,n\}$:

$$\sup_{S',\mathbf{x}^*\in\mathbb{R}^d,y^*\in Y}\left|\mathcal{F}_{\boldsymbol{\omega}}\big((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\big) - \mathcal{F}_{\boldsymbol{\omega}}\big((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{i-1},y_{i-1}),(\mathbf{x}^*,y^*),(\mathbf{x}_{i+1},y_{i+1}),\ldots,(\mathbf{x}_n,y_n)\big)\right| \leq \frac{1}{n},$$

Thus, we apply the Efron-Stein inequality (following Boucheron et al., 2013, Corollary 3.2) to obtain

$$\underset{S'\sim\mathcal{D}^n}{\mathbf{Var}}\left(\mathcal{F}_{\boldsymbol{\omega}}(S')\right) \leq \frac{1}{4}\sum_{i=1}^{n}\left(\frac{1}{n}\right)^2 = \frac{1}{4n}\,.$$

□

A.2 Kernel Alignment Loss Computation

The kernel learning algorithms presented in Section 5 require to compute the empirical kernel alignment loss for each hypothesis $h_{\boldsymbol{\omega}}$, given by

$$\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}}) = \frac{1}{n(n-1)}\sum_{i\neq j}^{n}\ell(h_{\boldsymbol{\omega}}(\boldsymbol{\delta}_{ij}),\lambda_{ij})\,. \tag{21}$$

A naive implementation of Equation (21) would need $O(n^2)$ steps. Propositions 7 and 8 below show how to rewrite Equation (21) in a form that needs $O(n)$ steps. Proposition 7 is dedicated to the binary classification, and is equivalent to the computation method proposed by Sinha and Duchi (2016). By Proposition 8, we extend the result to the multi-classification case.

Proposition 7 (Binary classification). *When $S = (\mathbf{x}_i, y_i)_{i=1}^n \in (\mathbb{R}^d \times \{-1,1\})^n$, we have*

$$\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}}) = \frac{n}{2(n-1)} - \frac{1}{2n(n-1)}\left[\left(\sum_{i=1}^{n} y_i\cos(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2 + \left(\sum_{i=1}^{n} y_i\sin(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2\right].$$

That is, in the binary classification case ($y\in\{-1,1\}$), one can compute the empirical alignment loss $\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}})$ in $O(n)$ steps.

Proof. Using the cosine trigonometric identity

$$\begin{aligned}
\sum_{i\neq j}^{n}\lambda_{ij}h_{\boldsymbol{\omega}}(\mathbf{x}_i-\mathbf{x}_j) &= \sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j\cos(\boldsymbol{\omega}\cdot(\mathbf{x}_i-\mathbf{x}_j)) - n\\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j\left(\cos(\boldsymbol{\omega}\cdot\mathbf{x}_i)\cos(\boldsymbol{\omega}\cdot\mathbf{x}_j) + \sin(\boldsymbol{\omega}\cdot\mathbf{x}_i)\sin(\boldsymbol{\omega}\cdot\mathbf{x}_j)\right) - n\\
&= \left(\sum_{i=1}^{n} y_i\cos(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2 + \left(\sum_{i=1}^{n} y_i\sin(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2 - n
\end{aligned}$$

Thus,

$$\begin{aligned}
\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}}) &= \frac{1}{n(n-1)}\sum_{i\neq j}^{n}\ell(h_{\boldsymbol{\omega}}(\boldsymbol{\delta}_{ij}),\lambda_{ij})\\
&= \frac{1}{n(n-1)}\sum_{i\neq j}^{n}\frac{1-\lambda_{ij}h_{\boldsymbol{\omega}}(\boldsymbol{\delta}_{ij})}{2}\\
&= \frac{1}{2} - \frac{1}{2n(n-1)}\sum_{i\neq j}^{n}\lambda_{ij}h_{\boldsymbol{\omega}}(\mathbf{x}_i-\mathbf{x}_j)\\
&= \frac{1}{2} - \frac{1}{2n(n-1)}\left[\left(\sum_{i=1}^{n} y_i\cos(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2 + \left(\sum_{i=1}^{n} y_i\sin(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2 - n\right]\\
&= \frac{n}{2(n-1)} - \frac{1}{2n(n-1)}\left[\left(\sum_{i=1}^{n} y_i\cos(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2 + \left(\sum_{i=1}^{n} y_i\sin(\boldsymbol{\omega}\cdot\mathbf{x}_i)\right)^2\right].
\end{aligned}$$

□

Proposition 8 (Multi-class classification)**.** *When* $S = (\mathbf{x}_i, y_i)_{i=1}^n \in (\mathbb{R}^d \times \{1, \dots, L\})^n$*, we have*

$$\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}}) = \frac{n}{2(n-1)} - \frac{1}{2n(n-1)} \left[2 \sum_{y=1}^{L} (c_y^2 + s_y^2) - \left(\sum_{y=1}^{L} c_y \right)^2 - \left(\sum_{y=1}^{L} s_y \right)^2 \right],$$

with

$$c_y \coloneqq \sum_{\mathbf{x} \in S_y} \cos(\boldsymbol{\omega} \cdot \mathbf{x}) \qquad \textit{and} \qquad s_y \coloneqq \sum_{\mathbf{x} \in S_y} \sin(\boldsymbol{\omega} \cdot \mathbf{x}) \,.$$

That is, in the multi-class classification case with L classes ($y \in \{1, \dots, L\})^n$), one can compute the empirical alignment loss $\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}})$ in $O(n)$ steps.

Proof.

$$\begin{aligned}
\sum_{i \neq j}^{n} \lambda_{ij} h_{\boldsymbol{\omega}}(\mathbf{x}_i - \mathbf{x}_j) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{ij} \cos(\boldsymbol{\omega} \cdot (\mathbf{x}_i - \mathbf{x}_j)) - n \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} (2I[y_i = y_j] - 1) \cos(\boldsymbol{\omega} \cdot (\mathbf{x}_i - \mathbf{x}_j)) - n \\
&= 2 \sum_{i=1}^{n} \sum_{j=1}^{n} I[y_i = y_j] \cos(\boldsymbol{\omega} \cdot (\mathbf{x}_i - \mathbf{x}_j)) - \sum_{i=1}^{n} \sum_{j=1}^{n} \cos(\boldsymbol{\omega} \cdot (\mathbf{x}_i - \mathbf{x}_j)) - n
\end{aligned}$$

Let's denote $S_y \coloneqq \{\mathbf{x}_i | (\mathbf{x}_i, y) \in S\}$. We have

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{j=1}^{n} I[y_i = y_j] \cos(\boldsymbol{\omega} \cdot (\mathbf{x}_i - \mathbf{x}_j)) &= \sum_{y=1}^{L} \sum_{\mathbf{x} \in S_y} \sum_{\mathbf{x}' \in S_y} \cos(\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{x}')) \\
&= \sum_{y=1}^{L} \left[\left(\sum_{\mathbf{x} \in S_y} \cos(\boldsymbol{\omega} \cdot \mathbf{x}) \right)^2 + \left(\sum_{\mathbf{x} \in S_y} \sin(\boldsymbol{\omega} \cdot \mathbf{x}) \right)^2 \right],
\end{aligned}$$

and

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \cos(\boldsymbol{\omega} \cdot (\mathbf{x}_i - \mathbf{x}_j)) = \left(\sum_{y=1}^{L} \sum_{\mathbf{x} \in S_y} \cos(\boldsymbol{\omega} \cdot \mathbf{x}) \right)^2 + \left(\sum_{y=1}^{L} \sum_{\mathbf{x} \in S_y} \sin(\boldsymbol{\omega} \cdot \mathbf{x}) \right)^2 .$$

Thus, we can rewrite

$$\sum_{i \neq j}^{n} \lambda_{ij} h_{\boldsymbol{\omega}}(\mathbf{x}_i - \mathbf{x}_j) = 2 \sum_{y=1}^{L} (c_y^2 + s_y^2) - \left(\sum_{y=1}^{L} c_y \right)^2 - \left(\sum_{y=1}^{L} s_y \right)^2 - n \,.$$

Therefore,

$$\begin{aligned}
\widehat{\mathcal{L}}_S(h_{\boldsymbol{\omega}}) &= \frac{1}{n(n-1)} \sum_{i \neq j}^{n} \ell(h_{\boldsymbol{\omega}}(\boldsymbol{\delta}_{ij}), \lambda_{ij}) \\
&= \frac{1}{n(n-1)} \sum_{i \neq j}^{n} \frac{1 - \lambda_{ij} h_{\boldsymbol{\omega}}(\boldsymbol{\delta}_{ij})}{2} \\
&= \frac{1}{2} - \frac{1}{2n(n-1)} \sum_{i \neq j}^{n} \lambda_{ij} h_{\boldsymbol{\omega}}(\mathbf{x}_i - \mathbf{x}_j) \\
&= \frac{1}{2} - \frac{1}{2n(n-1)} \left[2 \sum_{y=1}^{L} (c_y^2 + s_y^2) - \left(\sum_{y=1}^{L} c_y \right)^2 - \left(\sum_{y=1}^{L} s_y \right)^2 - n \right] \\
&= \frac{n}{2(n-1)} - \frac{1}{2n(n-1)} \left[2 \sum_{y=1}^{L} (c_y^2 + s_y^2) - \left(\sum_{y=1}^{L} c_y \right)^2 - \left(\sum_{y=1}^{L} s_y \right)^2 \right] .
\end{aligned}$$

□

A.3 Experiments

Implementation details. The code used to run the experiments is available at:

https://github.com/gletarte/pbrff

In Section 6 we use the following datasets:

ads http://archive.ics.uci.edu/ml/datasets/Internet+Advertisements
The first 4 features which have missing values are removed.

adult https://archive.ics.uci.edu/ml/datasets/Adult

breast https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).

farm https://archive.ics.uci.edu/ml/datasets/Farm+Ads

mnist http://yann.lecun.com/exdb/mnist/
As Sinha and Duchi (2016), binary classification tasks are compiled with the following digits pairs: 1 vs. 7, 4 vs. 9, and 5 vs. 6.

We split the datasets into training and testing sets with a 75/25 ratio except for adult which has a training/test split already computed. We then use 20% of the training set for validation. Table 2 presents an overview. We use the following parameter values range for selection on the validation set:

- $C \in \{10^{-5}, 10^{-4}, \ldots, 10^{4}\}$
- $\sigma \in \{10^{-7}, 10^{-6}, \ldots, 10^{2}\}$
- $\rho \in \{10^{-4}N, 10^{-3}N, \ldots, 10^{0}N\}$
- $\beta \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$
- $D \in \{8, 16, 32, 64, 128\}$

Dataset	n_{train}	n_{valid}	n_{test}	d
ads	1967	492	820	1554
adult	26048	6513	16281	108
breast	340	86	143	30
farm	2485	622	1036	54877
mnist17	9101	2276	3793	784
mnist49	8268	2068	3446	784
mnist56	7912	1979	3298	784

Table 2: Datasets overview.

Supplementary experiments. Figures 4 and 6 present extra results obtained for the *landmark-based learning* experiments (Subsection 6.1). Figure 5 gives extra results for the *greedy kernel learning* experiment (Subsection 6.2).

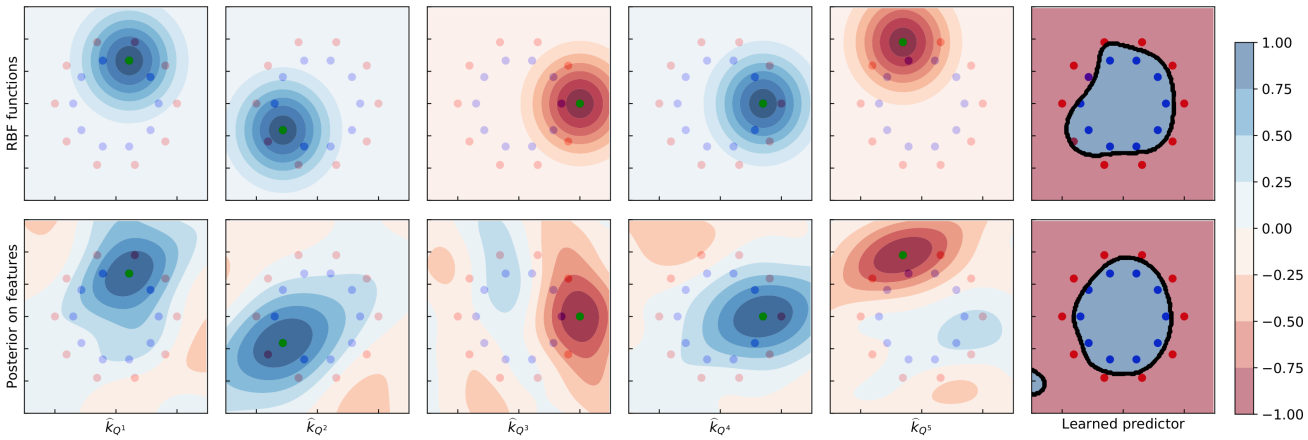
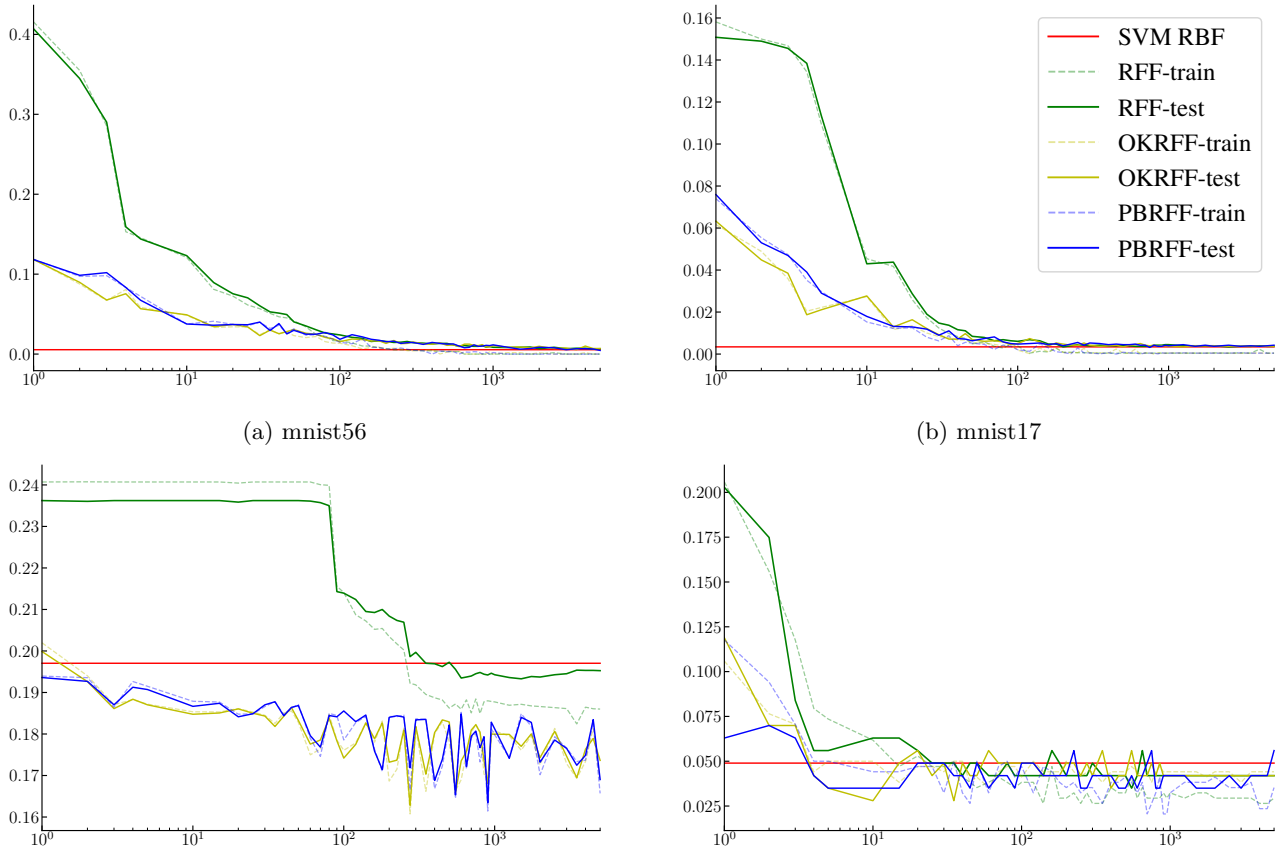


Figure 4: Repetition of Figure 1's experiment, with another toy dataset. First row shows selected RBF-Landmarks kernel outputs, while second row shows the corresponding learned similarity measures on random Fourier features (PB-Landmarks). The rightmost column displays the classification learned by a linear SVM over the mapped dataset.



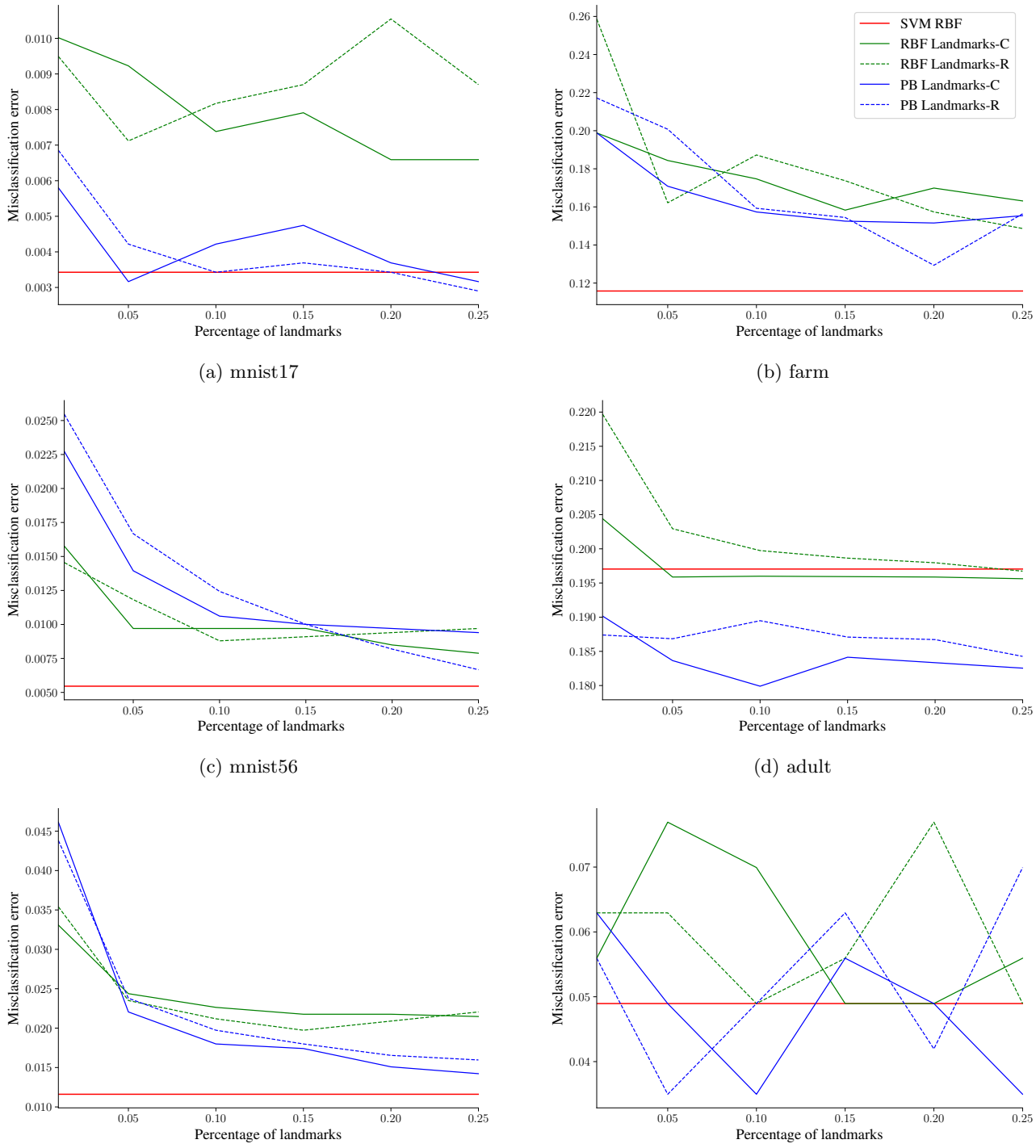
(a) mnist56

(b) mnist17

(c) adult

(d) breast

Figure 5: Train and test error of the kernel learning approaches according to the number of random features D on the remaining 4 datasets (not reported by Figure 3).



(a) mnist17

(b) farm

(c) mnist56

(d) adult

(e) mnist49

(f) breast

Figure 6: Behavior of the landmarks-based approach according to the percentage of training points selected as landmarks on the remaining 6 datasets (not reported by Figure 2).