Pseudo-Bayesian Learning with Kernel Fourier Transform as Prior

Gaël Letarte[1], Emilie Morvant[2], Pascal Germain[3]

[1] Département d'informatique et de génie logiciel, Université Laval, Québec, Canada

[2] Univ Lyon, UJM-St-Etienne, CNRS , IOGS, Lab Hubert Curien UMR 5516, St-Etienne, France

[3] Équipe-projet Modal, Inria Lille - Nord Europe, Villeneuve d'Ascq, France

Introduction

We revisit **Rahimi and Recht (2007)**'s kernel random Fourier features (RFF) method through the lens of the PAC-Bayesian theory.

New perspective on RFF

- The Fourier transform is a *prior* distribution on trigonometric hypotheses.
- Then we learn a *pseudo-posterior* by minimizing PAC-Bayesian bounds.

Two learning approaches

- A kernel alignment algorithm.
- A landmarks-based similarity measure learning.

Random Fourier Features (RFF)

Let's consider a **translation invariant** kernel $k:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$:

$$k(\mathbf{x},\mathbf{x}')=k(\mathbf{x}-\mathbf{x}')=k(\boldsymbol{\delta})\quad\text{with}\quad \boldsymbol{\delta}:=\mathbf{x}-\mathbf{x}'$$

Denote $p(\boldsymbol{\omega})$ the Fourier transform of $k(\boldsymbol{\delta})$,

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

Thus,

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

Example : Gaussian Kernel

The Gaussian kernel is given by

$$k_\sigma(\mathbf{x},\mathbf{x}')=\exp\big(-\tfrac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{x}'\|^2\big)$$

Its Fourier transform is

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

PAC-Bayesian Theory

Given a data distribution $\mathcal{D}$, a training set $S=\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\}\sim\mathcal{D}^n$, with $\mathbf{x}_i\in\mathbb{R}^d$ and $y_i\in\mathcal{Y}\subset\mathbb{N}$, a loss $\ell:\{-1,1\}^2\to[0,1]$, a predictor $f\in F$:

$$\mathcal{L}_{\mathcal{D}}(f)\;=\;\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\ell\Big(f(\mathbf{x}),y\Big)\qquad\textit{generalization loss}$$

$$\widehat{\mathcal{L}}_S(f)\;=\;\frac{1}{n}\sum_{i=1}^{n}\ell\Big(f(\mathbf{x}_i),y_i\Big)\qquad\textit{empirical loss}$$

PAC-Bayesian Theorem (Lever et al. 2013)

For $t>0$, for any prior π on F, with probability $1-\varepsilon$ on the choice of $S\sim\mathcal{D}^n$, we have for all posterior distribution ρ on F :

$$\mathop{\mathbf{E}}_{f\sim\rho}\mathcal{L}_{\mathcal{D}}(f)\leq\mathop{\mathbf{E}}_{f\sim\rho}\widehat{\mathcal{L}}_S(f)+\frac{1}{t}\left(\mathrm{KL}(\rho\|\pi)+\frac{t^2}{2n}+\ln\frac{1}{\varepsilon}\right)$$

Thus, the **optimal posterior distribution** ρ^* is :

$$\rho^*(f)\;=\;\frac{1}{Z}\pi(f)\,\exp\Big(-t\,\widehat{\mathcal{L}}_S(f)\Big)$$

PAC-Bayesian Theory for RFF

Idea : See the Fourier transform of a kernel as a *prior* over predictors.

$$\begin{aligned}k_p(\mathbf{x}-\mathbf{x}')&=\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p}\cos\big(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')\big)\\&=\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p}h_{\boldsymbol{\omega}}(\boldsymbol{\delta})\qquad\text{with}\qquad h_{\boldsymbol{\omega}}(\boldsymbol{\delta}):=\cos(\boldsymbol{\omega}\cdot\boldsymbol{\delta})\end{aligned}$$

Kernel alignment loss

Loss of a predictor h_ω on $(\boldsymbol{\delta},\lambda)\sim\Delta_{\mathcal{D}}$, given by two draws according to $\mathcal{D}$:

$$\ell\big(h_\omega(\boldsymbol{\delta}),\lambda\big):=\frac{1-\lambda h_\omega(\boldsymbol{\delta})}{2},\qquad\text{with }\lambda:=\begin{cases}1&\text{if }y=y',\\-1&\text{otherwise.}\end{cases}$$

Goal : Learn a posterior q to minimize the loss of k_q.

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

Empirical loss estimation

Given $S\sim\mathcal{D}^n$, an *unbiased* second-order estimator of $\mathcal{L}_{\mathcal{D}}(k_q)$ is :

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

Contact : gael.letarte.1@ulaval.ca

Code : github.com/gletarte/pbrff

NSERC CRSNG APRIORI compute canada | calcul canada

Approach 1 : Kernel Alignment Algorithm

PAC-Bayesian Corollary

For $t>0$ and a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$, we have $\forall q$ on $\mathbb{R}^d$:

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

Consider a uniform prior P on N *random features* drawn according to p, then learn $Q(h_m)$ for $m=1,\ldots,N$:

$$Q(h_m)\;=\;\frac{1}{Z}\exp\Big(-t\,\widehat{\mathcal{L}}_S(h_m)\Big)$$

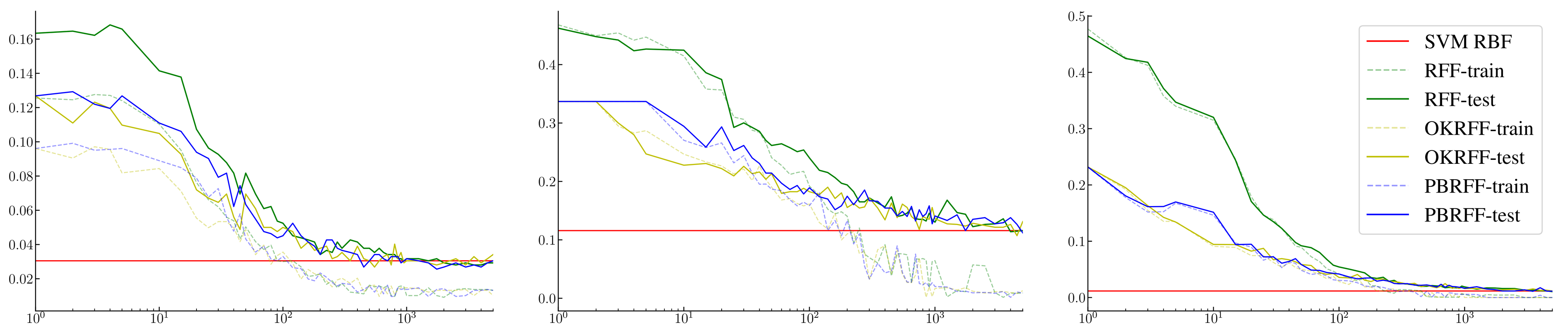
We then sample $D<N$ features according to Q and execute an SVM using the learned kernel

$$\widehat{k}_Q(\mathbf{x},\mathbf{x}'):=\frac{1}{D}\sum_{i=1}^{D}h_i(\mathbf{x}-\mathbf{x}')$$

Also in the paper

A general PAC-Bayesian theorem for $\mathcal{L}_{\mathcal{D}}(k_q)$, where $\mathrm{KL}(q\|p)$ can be replaced by other f-divergences. As a special case, it provides a justification to the *Optimal Kernel (OK) alignment algorithm* of Sinha and Duchi (2016).

Experiments



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

Figure 1: Train and test error of the kernel learning approaches according to the number of random features D.

Approach 2 : Similarity Measure Learning

For a **subset of landmarks** $L=\{(\mathbf{x}_l,y_l)\}_{l=1}^{|L|}\subseteq S$ chosen such that $(\mathbf{x}_i,y_i)\in S$:

$$\mathcal{L}^l_{\mathcal{D}}(k_q):=\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\ell\Big(k_q(\mathbf{x}_l-\mathbf{x}),\lambda(y_l,y)\Big),\;\widehat{\mathcal{L}}^l_S(k_q):=\frac{1}{n-1}\sum_{j=1,j\neq l}^{n}\ell\Big(k_q(\mathbf{x}_l-\mathbf{x}_j),\lambda(y_l,y_j)\Big)$$

PAC-Bayesian Corollary

For $t>0$, $l\in\{1,\ldots,|L|\}$, and a prior distribution p over $\mathbb{R}^d$, with probability $1-\varepsilon$, we have $\forall q$ on $\mathbb{R}^d$:

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

For all $\mathbf{x}_l\in L$, consider a uniform prior P on D *random features* drawn according to p, then learn $Q^l(h^l_m)$

$$Q^l(h^l_m)\;=\;\frac{1}{Z_l}\exp\Big(-t\,\widehat{\mathcal{L}}^l_S(h^l_m)\Big)$$

Then execute an SVM on the following mapping

$$\mathbf{x}\mapsto\Big(\widehat{k}_{Q^1}(\mathbf{x}_1,\mathbf{x}),\widehat{k}_{Q^2}(\mathbf{x}_2,\mathbf{x}),\ldots,\widehat{k}_{Q^{|L|}}(\mathbf{x}_{|L|},\mathbf{x})\Big)$$

Experiments

- Intuition of the method with toy problem :

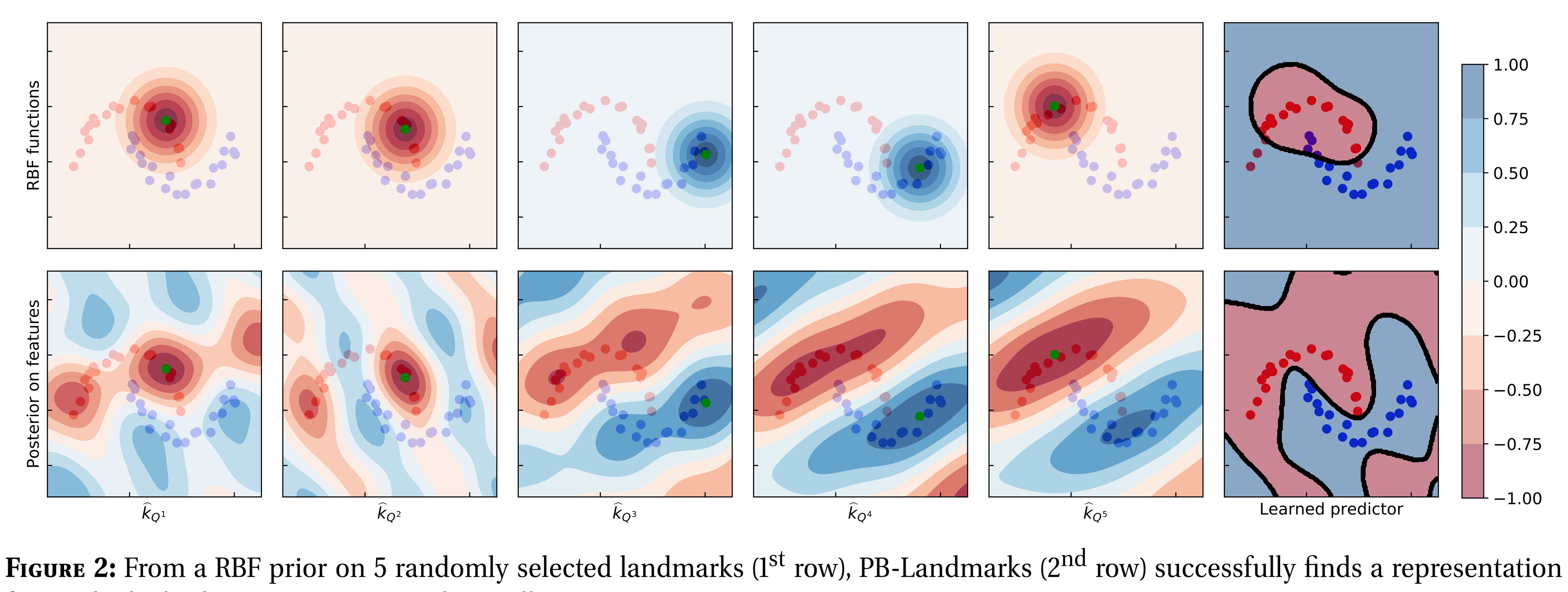


Figure 2: From a RBF prior on 5 randomly selected landmarks (1st row), PB-Landmarks (2nd row) successfully finds a representation from which the linear SVM can predict well.

- Real data results :

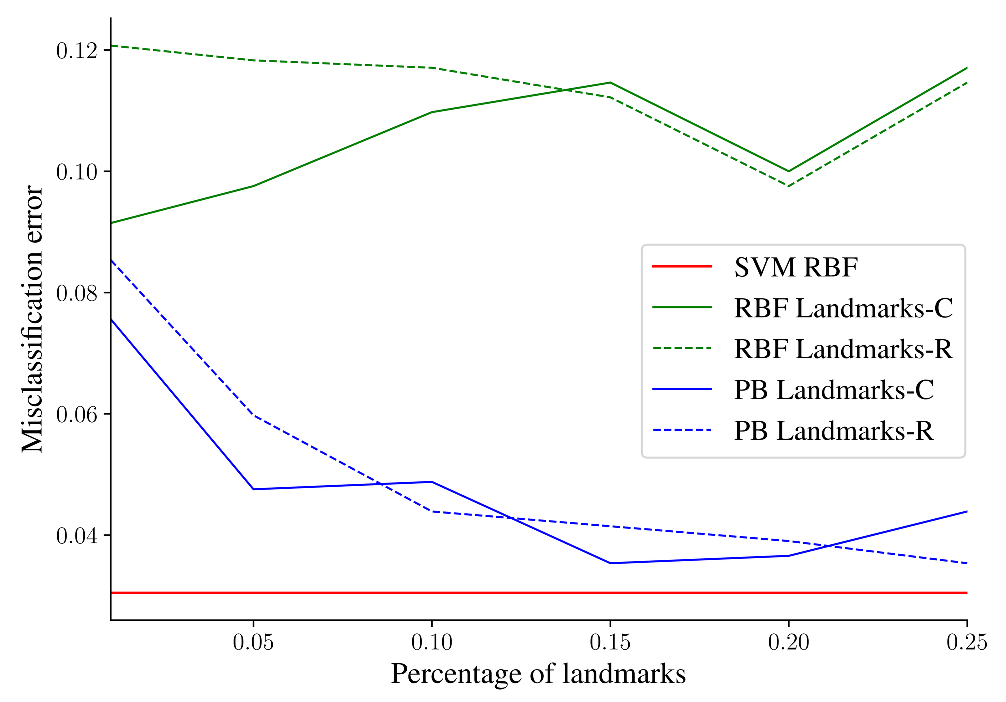


Figure 3: Behavior of the landmarks-based approach according to the percentage of landmarks on the dataset "ads".

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

Dataset		landmarks-based			
	SVM	RBF	PB	$\text{PB}_{L=\sqrt{n}}$	$\text{PB}_{D=64}$
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