Wavelet-based Bayesian approximate kernel method for high-dimensional data analysis

Wenxing Guo, School of Mathematics, Statistics and Actuarial Science, University of Essex & Department of Mathematical and Statistical Sciences, University of Alberta

Xueying Zhang, Department of Mathematical and Statistical Sciences, University of Alberta

Bei Jiang, Department of Mathematical and Statistical Sciences, University of Alberta

Linglong Kong, Department of Mathematical and Statistical Sciences, University of Alberta

Yaozhong Hu, Department of Mathematical and Statistical Sciences, University of Alberta

Accepted for publication in **Computational Statistics**

Abstract

Kernel methods are often used for nonlinear regression and classification in statistics and machine learning because they are computationally cheap and accurate. The wavelet kernel functions based on wavelet analysis can efficiently approximate any nonlinear functions. In this article, we construct a novel wavelet kernel function in terms of random wavelet bases and define a linear vector space that captures nonlinear structures in reproducing kernel Hilbert spaces (RKHS). Based on the wavelet transform, the data are mapped into a low-dimensional randomized feature space and convert kernel function into operations of a linear machine. We then propose a new Bayesian approximate kernel model with the random wavelet expansion and use the Gibbs sampler to compute the model's parameters. Finally, some simulation studies and two real datasets analyses are carried out to demonstrate that the proposed method displays good stability, prediction performance compared to some other existing methods.

Keywords

Kernel method; Wavelet transform; Randomized feature; Bayesian kernel model

1 Introduction

Machine learning problems with observations substantially smaller than the number of available variables, known as large p small n problem, are widespread and full of challenges. It is normal to use principal component analysis to lower the dimensions (Wang and Du 2000) or to use variable selection to reduce the number of variables (Chakraborty et al. 2012). Variable selection is well-developed for linear regression models, but it might not be practical or applicable in some situations. In this article, we focus on using nonlinear regression models to handle the large p small n problem in the reproducing kernel Hilbert spaces (RKHS) (Aronszajn 1950). Since support vector machine (SVM) (Vapnik 1999) was proposed, kernel supervised learning methods in RKHS have been widely used. Generalized kernel models (Zhang et al. 2011) are extensions of the generalized linear models induced by a reproducing kernel in the feature space. A usual way to train a nonlinear support vector machine is to approximate the factorization of the kernel matrix and process the columns of the factor matrix as features in a linear machine (deCoste and Mazzoni 2003). Bayesian approaches are applied to nonlinear classification and regression. Bayesian binary classification models in RKHS are proposed to analyze microarray data and produce smaller classification errors than some existing classification methods (Mallick et al. 2005). Bayesian approximate kernel regression model for nonlinear regression (Crawford et al. 2018) performs well in genomic selection and association mapping.

In the present work, we approximate the kernel function by factoring the kernel function itself. This method maps high-dimensional data into low-dimensional randomized feature space. Rahimi and Recht (2007) introduces that the kernel in the models can be approximated by random Fourier features. Inspired by their work, we construct a novel wavelet kernel function in terms of random wavelet bases. We then develop a new Bayesian approximate kernel model using the random wavelet bases. The experiment results indicate that random wavelet method yield higher accuracy in solving classification and regression problems.

1.1 Review of reproducing kernel Hilbert space

There is an issue that for many well-adopted kernels, the dimension of the Hilbert space is infinite (Wahba 1990). When training the dataset, it is preferred to solve an optimization problem in a finite-dimensional space. We define a class of space called reproducing kernel Hilbert space (RKHS) that transfers the infinite-dimensional space to finite-dimensional space.

Definition 1.1 Let $\mathcal{X}$ be a set. A reproducing kernel Hilbert space over $\mathcal{X}$ is a Hilbert space $\mathcal{H}$ consisting of functions on $\mathcal{X}$ such that for each $\mathbf{x} \in \mathcal{X}$, there is a function $k_{\mathbf{x}} \in \mathcal{H}$ with the property

$$\langle f, k_{\mathbf{x}} \rangle_{\mathcal{H}} = f(\mathbf{x}) \ (\forall f \in \mathcal{H}),$$

where $k(\cdot, \mathbf{x}) := k_{\mathbf{x}}(\cdot)$ is called a reproducing kernel of $\mathcal{H}$. The reproducing kernel $k(\mathbf{x}, \mathbf{y})$ is symmetric and positive definite: $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$ and for $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{X}$ and $a_1, \cdots, a_n \in \mathbb{R}$

$$\sum_{i,j=1,\cdots,n} a_i a_j k\big(\mathbf{x}_i, \mathbf{x}_j\big) \geq 0.$$

Suppose we are given a set of training data $\big\{\big(\mathbf{x}_i, y_i\big)\big\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$ is an input vector and $y_i \in \mathcal{Y} \subseteq \mathbb{R}$ is the continuous output for a regression problem or $y_i = \pm 1$ is the binary output for a classification problem. Consider the standard non-parametric problem and estimate $f(\mathbf{x})$ by the following penalized loss function

$$\widehat{f} = \arg\min_{f \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^n L\big(f\big(\mathbf{x}_i\big), y_i\big) + \lambda \|f\|_{\mathrm{K}}^2 \right], \tag{1}$$

where $L(f(\mathbf{x}), y)$ is a loss function, $\|\cdot\|_{\mathrm{K}}$ is the RKHS norm, see Hastie et al. (2009).

Lemma 1.1 (*Nonparametric Representer Theorem*) (*Scholkopf et al.* 2001). *Let $\mathcal{X}$ be a non-empty set, k is a positive definite real-valued kernel on $\mathcal{X} \times \mathcal{X}$, g is a strictly monotonically increasing real-valued function on $[0, \infty]$, L is an arbitrary cost function and $\mathcal{F}$ is a class of functions that satisfy*

$$\mathcal{F} = \left\{ f \in \mathbf{R}^{\mathcal{X}} \mid f(\cdot) = \sum_{i=1}^{\infty} \beta_i k\big(\cdot, \mathbf{z}_i\big), \beta_i \in \mathbb{R}, \mathbf{z}_i \in \mathcal{X}, \|f\|_{\mathrm{K}} < \infty \right\}.$$

Then, any $f \in \mathcal{F}$ minimizing the penalized loss function

$$C\big(\big(\mathbf{x}_1, y_1, f\big(\mathbf{x}_1\big)\big), \ldots, \big(\mathbf{x}_n, y_n, f\big(\mathbf{x}_n\big)\big)\big) + g\big(\|f\|_{\mathrm{K}}\big)$$

admits a representation of the form

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

where

$$C\big(\big(\mathbf{x}_1, y_1, f\big(\mathbf{x}_1\big)\big), \ldots, \big(\mathbf{x}_n, y_n, f\big(\mathbf{x}_n\big)\big)\big) = \frac{1}{n} \sum_{i=1}^n L\big(f\big(\mathbf{x}_i\big), y_i\big).$$

By the representer theorem, the solution for (1) can be written as

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

where $\big\{\alpha_i\big\}_{i=1}^n$ are the corresponding kernel coefficients.

Notice that $\|f\|_{\mathrm{K}}^2 = \sum_{i,j=1}^n k\big(\mathbf{x}_i, \mathbf{x}_j\big) \alpha_i \alpha_j$, substituting it into (1) we obtain

$$\widehat{f} = \arg\min_{f \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n} L\big(f\big(\mathbf{x}_i\big), y_i\big) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \right],$$

where $\boldsymbol{\alpha} = \big(\alpha_1, \cdots, \alpha_n\big)^T$ is an $n \times 1$ regression vector and $\mathbf{K} = \big(\mathbf{k}_1, \cdots, \mathbf{k}_n\big)$ is the $n \times n$ kernel matrix with $\mathbf{k}_i = \big(k\big(\mathbf{x}_i, \mathbf{x}_1\big), \cdots, k\big(\mathbf{x}_i, \mathbf{x}_n\big)\big)^T$, see Hastie et al. (2009).

1.2 Review of random features

The kernel trick can be used to generate features for algorithms easily. It is based on the inner product between pairs of input points (Rahimi and Recht 2007). However, the kernel tricks consume substantial computational and storage resources when dealing with large training sets. Instead of using the normal kernel function, we introduce a randomized feature map $\mathbf{z} : \mathbb{R}^p \to \mathbb{R}^d$ that maps the data into a low-dimensional inner product space. It utilizes the inner product between a pair of transformed points to approximate the kernel function:

$$k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \approx \mathbf{z}(\mathbf{x})^T \mathbf{z}(\mathbf{y}),$$

where $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{z}(\mathbf{x}) \in \mathbb{R}^d$. when $d > p$, the approximation also holds in the case of a low-dimensional d.

With the kernel trick, evaluating the machine at a test point $\mathbf{x}$ requires computing $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k\big(\mathbf{x}, \mathbf{x}_i\big)$, which has a time complexity of $O(np)$. For large datasets, the scaling of this kernel method is at least quadratic in the number of examples (Bazavan et al. 2012). Therefore, this method is impractical if the dataset is beyond 10^4 elements.

After introducing the randomized feature map and learning a hyperplane $\mathbf{w}$, a linear machine can be evaluated by simply computing $f(\mathbf{x}) = \mathbf{w}^T \mathbf{z}(\mathbf{x})$. With the randomized feature maps presented, the computation requires only $O(p + d)$ operations and storage (Rahimi and Recht 2007). We can transform the input $\mathbf{x}$ with the low-dimensional $\mathbf{z}$ and apply linear methods to approximate the nonlinear kernel machine at high speed.

Lemma 1.2 (Mercer-Hilbert-Schmidt Theorem) (*Wahba 1990*). *Let* $\{\phi_j\}$ *be an orthogonal sequence of continuous eigenfunctions on* $L_2(\mathcal{X})$ *and eigenvalues* $l_1 \geq l_2 \geq \cdots \geq 0$. *Let* k *be a continuous kernel on compact metric space* $\mathcal{X}$, *then* $\forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X}$

$$k(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{r} l_j \phi_j(\mathbf{x}) \phi_j(\mathbf{y}).$$

We define the feature functions $\boldsymbol{\psi}(\mathbf{x}) = \left\{\sqrt{l_j}\phi_j(\mathbf{x})\right\}_{j=1}^{r}$, *i.e.* $\psi_j(\mathbf{x}) = \sqrt{l_j}\phi_j(\mathbf{x})$. *Consequently, the estimated function f can be expressed as follows*

$$f(\mathbf{x}) = \sum_{j=1}^{r} b_j \psi_j(\mathbf{x}) = \boldsymbol{\psi}(\mathbf{x})^T \mathbf{b},$$

where $\mathbf{b} = \left(b_1, \ldots, b_r\right)^T$, *r represents the dimension of the feature space*, *see Zhang et al.* (2011).

Let $\mathbf{b} = \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\alpha}$. From $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k\left(\mathbf{x}, \mathbf{x}_i\right)$, we get $k = \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\psi}(\mathbf{x})$. For the shift-invariant kernel function: $k\left(\mathbf{x}_i, \mathbf{x}_j\right) = k\left(\mathbf{x}_i - \mathbf{x}_j\right)$, we have

$$k\left(\mathbf{x}_i - \mathbf{x}_j\right) = \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\psi}(\mathbf{x}) \approx \mathbf{z}\left(\mathbf{x}_i\right)^T \mathbf{z}\left(\mathbf{x}_j\right) = \tilde{k}\left(\mathbf{x}_i - \mathbf{x}_j\right),$$

where $\tilde{k}$ is the approximate kernel. To be more explicit, Similar to Crawford et al. (2018), we represent $\mathbf{z}$ as $\tilde{\boldsymbol{\psi}}$ and specify a matrix $\widetilde{\boldsymbol{\Psi}} = \left[\tilde{\boldsymbol{\psi}}\left(\mathbf{x}_1\right), \ldots, \tilde{\boldsymbol{\psi}}\left(\mathbf{x}_n\right)\right]$ with a corresponding approximate kernel matrix

$$\widetilde{\mathbf{K}} = \widetilde{\boldsymbol{\Psi}}^T \widetilde{\boldsymbol{\Psi}}.$$

2 Bayesian approximate kernel methods

2.1 Random wavelet bases

The motivation of wavelet analysis is to approximate a signal or a function by using a mother wavelet function ψ

$$\psi_{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right),$$

where $x, a, b \in \mathbb{R}$, x is a variable, $a \neq 0$, a is a dilation factor and b is a translation factor. ψ, for all practical purposes, should satisfy the requirement $\int \psi(x)dx = 0$. If $|a| < 1, \psi_{a,b}(x)$ has smaller time-width than $\psi(x)$ and is in a higher frequency; if $|a| > 1, \psi_{a,b}(x)$ has larger time-width than $\psi(x)$ and is in a lower frequency. Thus wavelet has time-widths adapted to their frequencies (Sifuzzaman et al. 2009). If a function $f(x) \in L_2(\mathbb{R})$, the wavelet transform of $f(x)$ is written as

$$\sum_{j \in Z} \sum_{k \in Z} < f, \psi_{j,k} > \psi_{j,k}(t).$$

The wavelet coefficients are

$$\left\langle f, \psi_{j,k} \right\rangle = d_{j,k} = \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt.$$

If $\psi(\cdot)$ is a mother wavelet and $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$, $i, j = 1, \ldots, n$, then the dot-product wavelet kernel is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \prod_{h=1}^{p} \left(\psi\left(\frac{x_{ih} - b}{a} \right) \cdot \psi\left(\frac{x_{jh} - b}{a} \right) \right),$$

where $\mathbf{x}_i = (x_{ih}, h = 1, 2, \ldots, p)$.

In Zhang and Ding (2017), the translation-invariant wavelet kernel is given

$$k(\mathbf{x}_i, \mathbf{x}_j) = \prod_{h=1}^{p} \left(\psi\left(\frac{x_{ih} - x_{jh}}{a} \right) \right).$$

Definition 2.1 (Mercer's condition) (Vapnik 2013). A real-valued function $k(\mathbf{x}, \mathbf{y})$ is said to fulfill Mercer's condition if for all square-integrable functions $g(\mathbf{x})$, we have

$$\iint g(\mathbf{x}) k(\mathbf{x}, \mathbf{y}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0.$$

If the Mercer's condition holds, we can write $k(\mathbf{x}, \mathbf{y})$ as a dot product $k(\mathbf{x}, \mathbf{y}) = \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle$, see Zhang et al. (2004).

Lemma 2.2 *The dot-product wavelet kernel satisfies Mercer's condition, i.e. it can be written as a dot product.*

Proof For $\forall g(\cdot) \in L_2(\mathbb{R}^p)$, we have

$$\begin{aligned}
&\iint_{\mathbb{R}^p \otimes \mathbb{R}^p} k(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_i) g(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j \\
&= \int_{\mathbb{R}^p} \prod_{h=1}^{p} \psi\left(\frac{x_{ih} - b}{a} \right) g(\mathbf{x}_i) d\mathbf{x}_i \int_{\mathbb{R}^p} \prod_{h=1}^{p} \psi\left(\frac{x_{jh} - b}{a} \right) g(\mathbf{x}_j) d\mathbf{x}_j \\
&= \left(\int_{\mathbb{R}^p} \prod_{h=1}^{p} \psi\left(\frac{x_{ih} - b}{a} \right) g(\mathbf{x}_i) d\mathbf{x}_i \right)^2 \geq 0.
\end{aligned}$$

Thus the dot-product wavelet kernel can be represented as

$$\begin{aligned}
k(\mathbf{x}_i, \mathbf{x}_j) &= \prod_{h=1}^{p} \left(\psi\left(\frac{x_{ih} - b}{a} \right) \cdot \psi\left(\frac{x_{jh} - b}{a} \right) \right) \\
&= \prod_{h=1}^{p} \psi\left(\frac{x_{ih} - b}{a} \right) \cdot \prod_{h=1}^{p} \psi\left(\frac{x_{jh} - b}{a} \right) \\
&= \Psi(\mathbf{x}_i) \cdot \Psi(\mathbf{x}_j).
\end{aligned}$$

We then can use the dot product of $\Psi(\mathbf{x}_i)$ and $\Psi(\mathbf{x}_j)$ to approximate the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. Selecting a mother wavelet ψ is essential for the random wavelet features method. In this article, we construct a dot-product wavelet kernel function in terms of the mother wavelet of Morlet wavelet function (Shyu and Sun 2002). The mother wavelet of Morlet wavelet function is defined as

$$\psi(x) = \cos(1.75 * x) \exp\left(-x^2/2\right).$$

The dot-product wavelet kernel function based on the mother wavelet of Morlet wavelet function can be expressed as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \prod_{h=1}^{p} \cos\left(1.75 * \left(\frac{1}{a}x_{ih} - \frac{b}{a}\right)\right) \exp\left(-\left(\frac{1}{a}x_{ih} - \frac{b}{a}\right)^2/2\right) \cdot \prod_{h=1}^{p} \cos\left(1.75 * \left(\frac{1}{a}x_{jh} - \frac{b}{a}\right)\right) \exp\left(-\left(\frac{1}{a}x_{jh} - \frac{b}{a}\right)^2/2\right).$$

The approximation of the wavelet kernel function using random wavelet bases is formulated as follows:

$$\begin{aligned}
&m_\ell \overset{\text{iid}}{\sim} N(0,1), \quad n_\ell \overset{\text{iid}}{\sim} N(0,1), \quad \ell = 1, \ldots, d, \\
&\mathbf{m} = \left[m_1, \ldots, m_d\right] \in \mathbb{R}^d, \quad \mathbf{n} = \left[n_1, \ldots, n_d\right] \in \mathbb{R}^d, \\
&\widetilde{\boldsymbol{\psi}}\left(\mathbf{x}_j\right)^\top = \left(\prod_{i=1}^{p} \psi\left(m_1 x_{ji} - n_1\right), \cdots, \prod_{i=1}^{p} \psi\left(m_d x_{ji} - n_d\right)\right), \quad j = 1, \cdots, n.
\end{aligned}$$

Let $\widetilde{\boldsymbol{\Psi}} = \left[\widetilde{\boldsymbol{\psi}}\left(\mathbf{x}_1\right), \ldots, \widetilde{\boldsymbol{\psi}}\left(\mathbf{x}_n\right)\right]$, we then use $\widetilde{\mathbf{K}} = \widetilde{\boldsymbol{\Psi}}^T \widetilde{\boldsymbol{\Psi}}$ to approximate the kernel function, where $\psi(\cdot)$ is the mother wavelet of Morlet wavelet function.

2.2 Generalized kernel models

We can treat the loss function $L\left(f\left(\mathbf{x}_i\right), y_i\right)$ in (1) as a negative conditional loglikelihood using the logarithmic scoring rule (Bernardo and Smith 2009). The generalized linear model (GLM) is defined as

$$\mathbf{y} \sim p(\mathbf{y} \mid \boldsymbol{\mu}) \quad \text{with} \quad \boldsymbol{\mu} = g(\mathbf{X}\boldsymbol{\beta}),$$

where $p(\mathbf{y} \mid \boldsymbol{\mu})$ is a distribution function, $\boldsymbol{\mu}$ is the expected value of response $\mathbf{y}$ conditional on the input $\mathbf{X}$, $\mathbf{X}\boldsymbol{\beta}$ is the linear operator, a linear combination of unknown parameters $\boldsymbol{\beta}$. $g(\cdot)$ is the link function.

The generalized kernel model (GKM) (Zhang et al. 2011) is derived from the GLM and can be written as

$$\mathbf{y} \sim p(\mathbf{y} \mid \boldsymbol{\mu}) \quad \text{with} \quad \boldsymbol{\mu} = g(\widetilde{\mathbf{K}}\boldsymbol{\alpha}). \tag{2}$$

This model can be obtained from the model

$$\mathbf{y} \sim p(\mathbf{y} \mid \boldsymbol{\mu}) \quad \text{with} \quad \boldsymbol{\mu} = g(\widetilde{\boldsymbol{\Psi}}\mathbf{b}),$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1} \widetilde{\boldsymbol{\Psi}}^T \mathbf{b}$.

The generalized models have been widely used in classification and regression problems which are based on kernel methods (Chakraborty 2009). According to the

application interests, we can specify a proper likelihood and link function. To be specific, the likelihood is set as the likelihood of the normal distribution, and the link function is set as the identity in the regression problems (Crawford et al. 2018). This article applies these generalized kernel models to regression and classification problems.

Since the approximate kernel matrix $\widetilde{\mathbf{K}}$ is symmetric and positive definite, the spectral decomposition of $\widetilde{\mathbf{K}}$ is as follows:

$$\widetilde{\mathbf{K}} = \widetilde{\mathbf{Q}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{Q}}^T,$$

where $\widetilde{\mathbf{Q}}$ is an $n \times n$ orthogonal matrix whose ith column is the eigenvector q_i of $\widetilde{\mathbf{K}}$ and $\widetilde{\mathbf{\Lambda}} = \mathrm{Diag}\left(\lambda_1, \cdots, \lambda_n\right)$ is a diagonal matrix, where eigenvalues $\lambda_1 \geq \lambda_2, \cdots, \geq \lambda_n$.

We rewrite the Eq. (2) as

$$\mathbf{y} \sim p(\mathbf{y} \mid \boldsymbol{\mu}) \quad \text{with} \quad \boldsymbol{\mu} = g(\widetilde{\mathbf{Q}}\boldsymbol{\theta}), \tag{3}$$

where $\boldsymbol{\theta} = \widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{Q}}^T\boldsymbol{\alpha}$. Eigenvectors corresponding to small eigenvalues can be truncated to reduce the computational complexity. Thus, we can keep the top s eigenvalues and consider $\widetilde{\mathbf{Q}}$ as an $n \times s$ matrix and $\widetilde{\lambda}$ as an $s \times s$ diagonal matrix. We can further reduce the dimension from n to s parameters. This new representation can substantially speed up estimating the model parameters, especially when n is large.

Considering a nonlinear function $E[\mathbf{y}] = f = \left[f\left(\mathbf{x}_1\right), \cdots, f\left(\mathbf{x}_n\right)\right]^T$, we have

$$\widetilde{\boldsymbol{\beta}} = \mathbf{X}^{\dagger} f,$$

where $\mathbf{X}^{\dagger} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$ is the Moore-Penrose generalized inverse.

Recall that $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{\Psi}}^T\mathbf{b}$ and $\boldsymbol{\theta} = \widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{Q}}^T\boldsymbol{\alpha}$, we have the following representation

$$\mathbf{b} = \left(\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{Q}}^T\widetilde{\mathbf{K}}^{-1}\mathbf{\Psi}^T\right)^{-1}\theta.$$

Thus, $\widetilde{\boldsymbol{\beta}}$ can be written as

$$\widetilde{\boldsymbol{\beta}} = \mathbf{X}^{\dagger}\mathbf{\Psi}^T\mathbf{b}.$$

2.3 Bayesian hierarchical model and Gibbs sampler

By Bayes' theorem for probability distributions, the posterior distribution is proportional to

$$p\left(\boldsymbol{\vartheta} \mid \left\{\mathrm{y}_i, \mathbf{x}_i\right\}_{i=1}^n\right) \propto \exp\left\{-\sum_{i=1}^{n} L\left(f\left(\mathbf{x}_i\right), \mathrm{y}_i\right)\right\}\pi(\boldsymbol{\vartheta}),$$

where $\pi(\vartheta)$ is the prior distribution and $\exp\left\{-\sum_{i=1}^{n} L\left(f\left(\mathbf{x}_i\right), \mathrm{y}_i\right)\right\}$ is the likelihood function.

Let $\mathbf{x}_j$ be an observation and ϑ_j is a parameter governing the data generating process for $\mathbf{x}_j$. Assume that the parameters $\vartheta_1, \vartheta_2, \ldots, \vartheta_j$ are generated from the distribution governed by a hyperparameter φ. For the Bayesian hierarchical model, Bernardo et al. (1985) gives the following stages.

Stage 1. $\mathbf{x}_j \mid \vartheta_j, \varphi \sim p\left(\mathbf{x}_j \mid \vartheta_j, \varphi\right)$.

Stage 2. $\vartheta_j \mid \varphi \sim p(\vartheta_j \mid \varphi)$.

Stage 3. $\varphi \sim p(\varphi)$.

Thus, the posterior distribution is proportional to

$$\begin{aligned} p\left(\varphi, \vartheta_j \mid \mathbf{x}_j\right) &\propto p\left(\mathbf{x}_j \mid \vartheta_j, \varphi\right) p\left(\vartheta_j, \varphi\right) \\ &\propto p\left(\mathbf{x}_j \mid \vartheta_j\right) p\left(\vartheta_j \mid \varphi\right) p(\varphi). \end{aligned}$$

Markov chain Monte Carlo (MCMC) methods include a class of algorithms for sampling from probability distributions. The development of MCMC methods allows us to compute the large Bayesian hierarchical model with thousands of unknown parameters (Banerjee et al. 2003). In the application of the Bayesian hierarchical model, the Gibbs sampler is the most basic MCMC method (Lynch 2007). A general Gibbs sampler follows the following iterative process,

0. Assign a vector of starting values $\mathbf{S}$, $\boldsymbol{\theta}^{j=0} = \mathbf{S}$, where j is the iteration count.
1. Let $j = j + 1$.
2. Sample $\left(\theta_1^j \mid \theta_2^{j-1}, \theta_3^{j-1}, \ldots, \theta_k^{j-1}\right)$.
3. Sample $\left(\theta_2^j \mid \theta_1^j, \theta_3^{j-1}, \ldots, \theta_k^{j-1}\right)$.

k. Sample $\left(\theta_k^j \mid \theta_1^j, \theta_2^j, \ldots, \theta_{k-1}^j\right)$.

k+1. Return to step 1.

2.4 Bayesian approximate kernel method for regression

We restate Eq. (2) for the regression problem as,

$$\mathbf{y} = \widetilde{\mathbf{K}}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N\left(\mathbf{0}, \tau^2\mathbf{I}\right), \tag{4}$$

where $\boldsymbol{\varepsilon}$ is the random error vector, $N\left(\mathbf{0}, \tau^2\mathbf{I}\right)$ is the multivariate normal distribution with the mean zero vector and the covariance matrix $\tau^2\mathbf{I}$, and $\mathbf{I}$ is the identity matrix.

Combining the hierarchical model with the factor representation in Eq. (3), we can formulate the specific hierarchical model for the nonlinear regression model as follows

$$\begin{aligned} \mathbf{y} &= \widetilde{\mathbf{Q}}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N\left(\mathbf{0}, \tau^2\mathbf{I}\right), \\ \boldsymbol{\theta} &\sim N\left(\mathbf{0}, \sigma^2\widetilde{\boldsymbol{\Lambda}}\right), \\ \sigma^2, \tau^2 &\sim \text{Scale-inv-}\ \chi^2(\nu, \phi). \end{aligned} \tag{5}$$

The idea of using $\boldsymbol{\theta}$ in (5) instead of using $\boldsymbol{\alpha}$ in (4) is from the Silverman g-prior (Zhang et al. 2011). The variance of random error τ^2 and the shrinkage parameter σ^2 both come from the scaled inverse chi-squared distribution with the degrees of freedom ν and the scale parameter ϕ. The probability density function of the scaled inverse chi-squared distribution over the domain $x > 0$ is

$$f(x;\nu,\phi) = \frac{(\phi\nu/2)^{\nu/2}}{\Gamma(\nu/2)} \frac{\exp\left[\frac{-\nu\phi}{2x}\right]}{x^{1+\nu/2}}.$$

Given the Bayesian hierarchical model in (5), we propose the conditional densities $p\left(\boldsymbol{\theta} \mid \sigma^2, \tau^2, \mathbf{y}\right)$ using the Bayes' theorem. To be specific,

$$\begin{aligned} p\left(\boldsymbol{\theta} \mid \sigma^2, \tau^2, \mathbf{y}\right) &\propto p\left(\mathbf{y} \mid \boldsymbol{\theta}, \tau^2\right) p\left(\boldsymbol{\theta} \mid \sigma^2\right) \\ &\propto N(m^*, n^*), \end{aligned}$$

where $n^* = \tau^2\sigma^2\left(\tau^2\widetilde{\boldsymbol{\Lambda}}^{-1} + \sigma^2\mathbf{I}_q\right)^{-1}$ and $m^* = \tau^2 n^* \widetilde{\mathbf{Q}}^T\mathbf{y}$.

Similarly, we propose the conditional densities for σ^2 and τ^2. Inspired by Crawford et al. (2018), we then use a Gibbs sampler to generate the joint posterior $p\left(\boldsymbol{\theta}, \sigma^2, \tau^2 \mid \mathbf{y}\right)$, and the procedures are as follows

1. $\boldsymbol{\theta} \mid \sigma^2, \tau^2, \mathbf{y} \sim N(m^*, n^*)$, with $n^* = \tau^2\sigma^2\left(\tau^2\widetilde{\boldsymbol{\Lambda}}^{-1} + \sigma^2\mathbf{I}_q\right)^{-1}$ and $m^* = \tau^{-2} n^* \widetilde{\mathbf{Q}}^T\mathbf{y}$.
2. $\widetilde{\boldsymbol{\beta}} = \mathbf{X}^{\dagger}\widetilde{\boldsymbol{\Psi}}^T\left(\widetilde{\boldsymbol{\Lambda}}\widetilde{\mathbf{Q}}^T\widetilde{\mathbf{K}}^{-1}\widetilde{\boldsymbol{\Psi}}^T\right)^{-1}\boldsymbol{\theta}$.
3. $\sigma^2 \mid \boldsymbol{\theta}, \tau^2, \mathbf{y} \sim$ Scale-inv $-\chi^2\left(v_\sigma^*, \phi_\sigma^*\right)$, where $v_\sigma^* = v + q$ and $\phi_\sigma^* = v_\sigma^{*-1}\left(v\phi + \boldsymbol{\theta}^T\widetilde{\boldsymbol{\Lambda}}^{-1}\boldsymbol{\theta}\right)$.
4. $\tau^2 \mid \boldsymbol{\theta}, \sigma^2, \mathbf{y} \sim$ Scale-inv $-\chi^2\left(v_\tau^*, \phi_\tau^*\right)$, where $v_\tau^* = v + n$ and $\phi_\tau^* = v_\tau^{*-1}\left(v\phi + \mathbf{e}^T\mathbf{e}\right)$, with $\mathbf{e} = \mathbf{y} - \widetilde{\mathbf{Q}}\boldsymbol{\theta}$.

We achieve the following set of posterior samples by repeating the above procedure for T times

$$\left\{\boldsymbol{\theta}^{(t)}, \sigma^{2(t)}, \tau^{2(t)}, \widetilde{\boldsymbol{\beta}}^{(t)}\right\}_{t=1}^{T}.$$

For the sample test $\mathbf{X}$ observed, the prediction is stated as

$$\hat{\mathbf{y}} = \mathbf{X}\widetilde{\boldsymbol{\beta}}.$$

2.5 Bayesian approximate kernel method for classification

We extend the Bayesian approximate kernel method to binary classification. We also use the generalized kernel model for the classification problem,

$$\mathbf{y} \sim p(\mathbf{y} \mid \boldsymbol{\mu}) \quad \text{with} \quad \boldsymbol{\mu} = g(\widetilde{\mathbf{K}}\boldsymbol{\alpha}).$$

We can specify the hierarchical model for classification using the factor representation, where $\widetilde{\mathbf{K}} = \widetilde{\mathbf{Q}}\widetilde{\boldsymbol{\Lambda}}\widetilde{\mathbf{Q}}^T$,

$$\begin{aligned}
\mathrm{y}_i &= \begin{cases} 1 & \text{if } \mathrm{s}_i > 0 \\ 0 & \text{if } \mathrm{s}_i \leq 0 \end{cases}, \\
\mathbf{s} &= \widetilde{\mathbf{Q}}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{I}), \\
\boldsymbol{\theta} &\sim N\left(\mathbf{0}, \sigma^2\widetilde{\boldsymbol{\Lambda}}\right), \\
\sigma^2 &\sim \text{ Scale-inv- } \chi^2(\nu, \phi).
\end{aligned}$$

The vector of latent responses is defined as $\mathbf{s} = \left[s_1, \ldots, s_n\right]^T$. The MCMC procedure here is similar to the posterior sampling of probit regression (Albert and Chib 1993). Posterior samples are generated by iterating the following procedures:

(1) For $i = 1, \ldots, n$,

$$\mathrm{s}_i^{(t+1)} \mid \boldsymbol{\theta}, \sigma^2, \mathbf{s}^{(t)}, \mathbf{y} \sim \begin{cases} N(\widetilde{q}_i^T\boldsymbol{\theta}, 1)\mathbb{I}\left(\mathrm{s}_i^{(t)} \leq 0\right) & \text{if } \mathrm{s}_i^{(t)} \leq 0 \\ N(\widetilde{q}_i^T\boldsymbol{\theta}, 1)\mathbb{I}\left(\mathrm{s}_i^{(t)} > 0\right) & \text{if } \mathrm{s}_i^{(t)} > 0 \end{cases}.$$

(2) $\boldsymbol{\theta} \mid \mathbf{s}, \sigma^2, \mathbf{y} \sim N(m^*, n^*)$ where $n^* = \sigma^2\left(\widetilde{\boldsymbol{\Lambda}}^{-1} + \sigma^2\mathbf{I}\right)^{-1}$ and $m^* = n^*\widetilde{\mathbf{Q}^T}\mathbf{s}$.

(3) $\widetilde{\boldsymbol{\beta}} = \mathbf{X}^{\dagger}\widetilde{\boldsymbol{\Psi}}^T\left(\widetilde{\boldsymbol{\Lambda}}\widetilde{\mathbf{Q}}^T\widetilde{\mathbf{K}}^{-1}\widetilde{\boldsymbol{\Psi}}^T\right)^{-1}\boldsymbol{\theta}$.

(4) $\sigma^2 \mid \mathbf{s}, \boldsymbol{\theta}, \mathbf{y} \sim$ Scale-inv- $\chi^2(\nu^*, \phi^*)$, where $\nu^* = \nu + q$ and $\phi^* = \nu^{*-1}\left(\nu\phi + \boldsymbol{\theta}^T\widetilde{\boldsymbol{\Lambda}}^{-1}\boldsymbol{\theta}\right)$.

We obtain the following set of posterior samples by repeating the above procedure T times.

$$\left\{\boldsymbol{\theta}^{(t)}, \sigma^{2(t)}, \widetilde{\boldsymbol{\beta}}^{(t)}\right\}_{t=1}^{T}.$$

For the observations $\mathbf{X}$, the prediction holds $\hat{\mathbf{y}} = \mathbf{X}\widetilde{\boldsymbol{\beta}}$. For each response y_i,

$$\mathrm{y}_i = \begin{cases} 1 & \text{if } \mathrm{y}_i > 0 \\ 0 & \text{if } \mathrm{y}_i \leq 0 \end{cases}.$$

3 Simulations

In this section, we conduct simulations to evaluate the performances of the proposed method in both regression and classification problems. We also compare the random wavelet kernel method (WKM) with other classical methods, such as random Fourier features kernel (FKM) (Crawford et al. 2018), polynomial kernel (PKM) (Chakraborty et al. 2012), support vector machine (SVM) with Gaussian kernel (Noble 2006).

3.1 Simulations for regression problems

To evaluate the performance of proposed method in regression, we create continuous outcomes using the following generating polynomial model: $\mathbf{y} = \mathbf{X}^3\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim MVN(\mathbf{0}, \mathbf{I})$ and all elements of the coefficient vector $\boldsymbol{\beta}$ are independently generated from uniform (0, 1). $\mathbf{X}^3 = \mathbf{X} \circ \mathbf{X} \circ \mathbf{X}$ is the element-wise third power of $\mathbf{X}$. The row vectors of of $\mathbf{X}$ are independently generated from $0.05 + \tau$ uniform (0, 1), where τ is a constant. τ is used to control the fluctuation of elements in $\mathbf{X}$. Here, the larger the value of τ, the more severe the fluctuation of elements in $\mathbf{X}$. The number of iteration in the simulation is 30. For the regression problems, we use prediction mean squared error (PMSE) to compare out-of-sample predictive accuracy. The PMSE is defined as

$$\text{PMSE} = \|\mathbf{y}_t - \mathbf{X}_t^3\hat{\boldsymbol{\beta}}\|_2^2/n_t,$$

where $(\mathbf{y}_t, \mathbf{X}_t)$ represents the test dataset, n_t denotes the size of test dataset, we let $n_t = 0.2n$ in this simulations. $\hat{\boldsymbol{\beta}}$ represents the estimator of $\boldsymbol{\beta}$ corresponding to each method.

From Tables 1 and 2 and Fig. 1, we can see that, in general, the performance of the proposed method (WKM) is the best in terms of prediction accuracy and robustness among all considered methods. WKM has the smallest PMSE and standard deviation (SD) compared with the other two methods. More specifically, the proposed method can still maintain the best performance when the elements

Table 1 Comparisons of prediction mean squared error (PMSE) for random wavelet kernel method (WKM), random Fourier features kernel method (FKM) and polynomial kernel (PKM) based on n=500

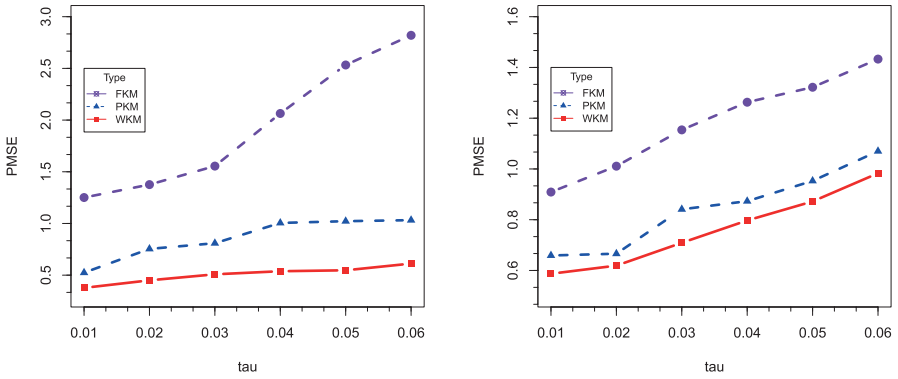
PMSE	$p = 500$			$p = 2000$			$p = 4000$		
	$\tau = 0.01$	$\tau = 0.05$	$\tau = 0.1$	$\tau = 0.01$	$\tau = 0.05$	$\tau = 0.1$	$\tau = 0.01$	$\tau = 0.05$	$\tau = 0.1$
FKM	1.251	2.533	2.884	1.480	2.122	2.854	2.436	3.512	4.724
	(0.190)	(0.373)	(0.494)	(0.286)	(0.221)	(0.290)	(0.275)	(0.341)	(0.562)
PKM	0.524	1.022	1.156	1.401	1.932	2.697	2.301	3.278	4.778
	(0.084)	(0.186)	(0.185)	(0.219)	(0.236)	(0.366)	(0.319)	(0.465)	(0.541)
WKM	0.377	0.547	0.625	1.205	1.761	2.526	2.115	3.079	4.495
	(0.062)	(0.093)	(0.093)	(0.210)	(0.201)	(0.258)	(0.215)	(0.271)	(0.532)

Standard deviations for the replicates of each model are given in the parentheses

Table 2 Comparisons of prediction mean squared error (PMSE) for random wavelet kernel method (WKM), random Fourier features kernel method (FKM) and polynomial kernel (PKM) based on p=1000

PMSE	$n = 500$			$n = 1000$			$n = 2000$		
	$\tau = 0.01$	$\tau = 0.05$	$\tau = 0.1$	$\tau = 0.01$	$\tau = 0.05$	$\tau = 0.1$	$\tau = 0.01$	$\tau = 0.05$	$\tau = 0.1$
FKM	0.909	1.225	1.450	1.033	1.729	2.380	0.285	0.438	0.632
	(0.131)	(0.160)	(0.174)	(0.094)	(0.181)	(0.281)	(0.022)	(0.033)	(0.042)
PKM	0.659	0.953	1.182	0.593	0.888	1.192	0.201	0.324	0.455
	(0.103)	(0.146)	(0.162)	(0.078)	(0.115)	(0.114)	(0.016)	(0.024)	(0.034)
WKM	0.587	0.772	1.016	0.302	0.440	0.576	0.149	0.228	0.311
	(0.082)	(0.102)	(0.154)	(0.036)	(0.045)	(0.074)	(0.013)	(0.017)	(0.024)

Standard deviations for the replicates of each model are given in the parentheses



(a) n=500, p=500 (b) n=500, p=1000

Fig. 1 Comparisons of PMSE based on different τ values

in $\mathbf{X}$ fluctuate and the dimension of $\mathbf{X}$ increases. On the other hand, random Fourier features kernel method (FKM) is more sensitive to fluctuation of data, and it performs poorly compared to the other two methods. The polynomial kernel method (PKM) is seen to be better than the FKM based on prediction accuracy and robustness, but its PMSE and SD are greater than those of WKM.

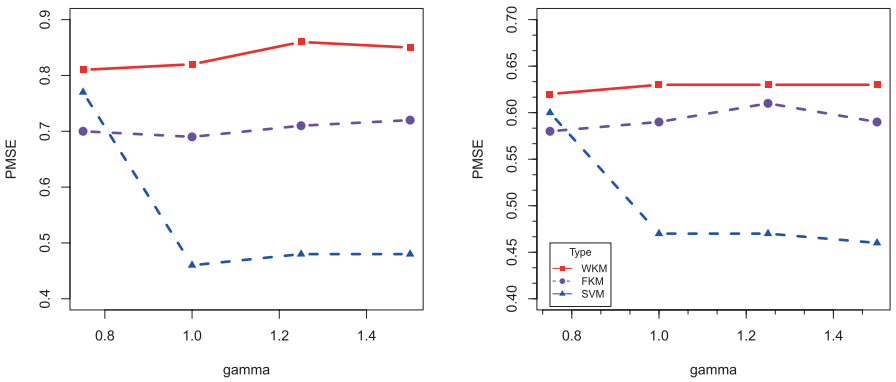
3.2 Simulations for classification problems

In order to evaluate the performance of the proposed method in classification problems, we still use the polynomial model: $f(\mathbf{Z}) = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim MVN(\mathbf{0}, \mathbf{I})$ and $\mathbf{Z} = \mathbf{X} \circ \mathbf{X} \circ \mathbf{X}$ is the element-wise third power of $\mathbf{X}$. But the generating model is different from the previous model in regression analysis. All elements of the data matrix $\mathbf{X}$ are independently generated from normal distribution $N(0, \gamma^2)$. The coefficient vector $\boldsymbol{\beta}$ is generated from multivariate normal distribution with mean vector $\mathbf{0}$ and covariance

Table 3 Comparisons of the accuracy and standard deviation for the random wavelet kernel (WKM), random Fourier features kernel (FKM) and support vector machine (SVM) methods based on $n = 500$

Accuracy	$p = 100$			$p = 500$			$p = 2000$		
	$\gamma = 0.75$	$\gamma = 1.00$	$\gamma = 1.50$	$\gamma = 0.75$	$\gamma = 1.00$	$\gamma = 1.50$	$\gamma = 0.75$	$\gamma = 1.00$	$\gamma = 1.50$
SVM	0.77	0.46	0.48	0.60	0.47	0.46	0.55	0.52	0.61
	(0.07)	(0.05)	(0.06)	(0.04)	(0.03)	(0.04)	(0.04)	(0.02)	(0.02)
FKM	0.70	0.69	0.72	0.58	0.59	0.59	0.63	0.61	0.60
	(0.04)	(0.07)	(0.06)	(0.05)	(0.05)	(0.05)	(0.04)	(0.05)	(0.05)
WKM	0.81	0.82	0.85	0.62	0.63	0.63	0.65	0.63	0.62
	(0.03)	(0.04)	(0.02)	(0.03)	(0.03)	(0.04)	(0.03)	(0.02)	(0.02)

Standard deviations for the replicates of each model are given in the parentheses



(a) n=500, p=100 (b) n=500, p=500

Fig. 2 Comparisons of PMSE based on different γ values

matrix $\mathbf{I}_p$. Here, the binary response variable y with values 1 or 0 is generated from the logistic model:

$$p(y_i = 1) = \frac{\exp\{f(\mathbf{Z}_i)\}}{1 + \exp\{f(\mathbf{Z}_i)\}},$$

where $\mathbf{Z}_i$ denotes the ith row of $\mathbf{Z}$, $i = 1, \ldots, n$.

We use Accuracy to evaluate the models of classification. The Accuracy is defined as

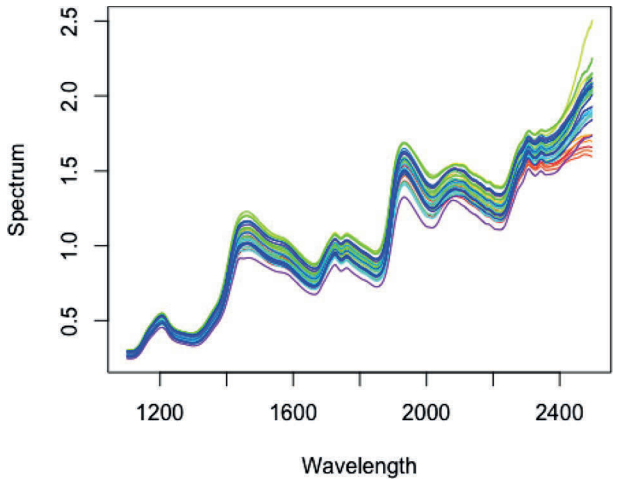


Fig. 3 The NIR spectrum of the observations in the biscuit dough piece dataset

Table 4 Comparisons of the prediction mean square error (PMSE) for the random wavelet kernel method (WKM) and random Fourier features kernel method (FKM)

Compositions	Methods	PMSE (SD)
Fat	FKM	0.459 (0.201)
	WKM	0.401 (0.237)
Sucrose	FKM	0.614 (0.271)
	WKM	0.347 (0.144)
Flour	FKM	0.526 (0.271)
	WKM	0.348 (0.139)
Water	FKM	0.387 (0.216)
	WKM	0.378 (0.177)

Standard deviations (SD) are given in the parentheses

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}.$$

From Table 3, we conclude that the the random wavelet kernel (WKM) has the highest accuracy and the lowest standard deviation among the three methods. Especially, when the ratio p/n is small, the WKM method performs better. Figure 2 shows that the results of the WKM method are more accurate and stable than those of FKM and SVM methods when the data fluctuate.

In a word, the proposed method has a good performance in both regression and classification simulations.

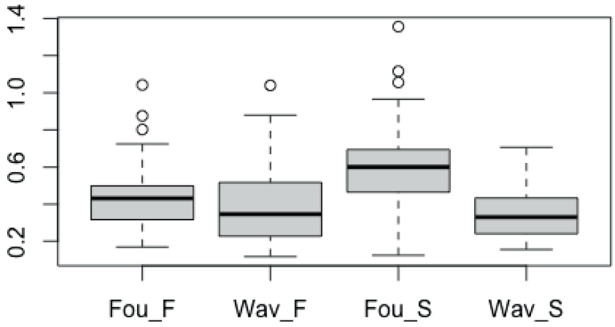


Fig. 4 Boxplots of the random wavelet kernel method and random Fourier features kernel method results for the prediction of the fat and sucrose in the biscuit. **Fou_F** represents using Fourier method for the prediction of the fat; **Wav_F** represents using wavelet method for the prediction of the fat; **Fou_S** represents using Fourier method for the prediction of sucrose; **Wav_S** represents using wavelet method for the prediction of sucrose

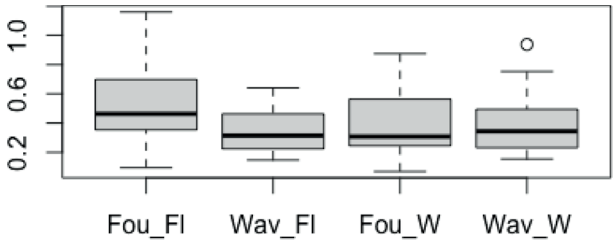


Fig. 5 Boxplots of the random wavelet kernel method and random Fourier features kernel method results for the prediction of the flour and water in the biscuit. **Fou_Fl** represents using Fourier method for the prediction of the flour; **Wav_Fl** represents using wavelet method for the prediction of the flour; **Fou_W** represents using Fourier method for the prediction of water; **Wav_W** represents using wavelet method for the prediction of water

4 Real data study

4.1 Real data for the regression problems

We further evaluate and compare the random wavelet kernel method with random Fourier features kernel method by analyzing the biscuit dough piece dataset from the R package functional datasets (fds) (brown et al. 2001). This example uses the near-infrared reflectance (NIR) spectra to measure the composition of biscuit dough pieces. The NIR spectrum of the observations is continuous curves, as shown in Fig. 3. The information from these curves can be used to predict the composition of the biscuit. The compositions of the biscuit we estimated include fat, sucrose, flour and water, and they all record in percent. We treat them as response values. The dataset contains 32 observations with 700 features. The results are shown in Table 4, Figs. 4 and 5.

Table 5 Comparisons of the accuracy and standard deviation (SD) of three methods for the Duke breast cancer dataset

Methods	Accuracy	SD
WKM	0.957	0.063
FKM	0.934	0.072
SVM	0.550	0.137

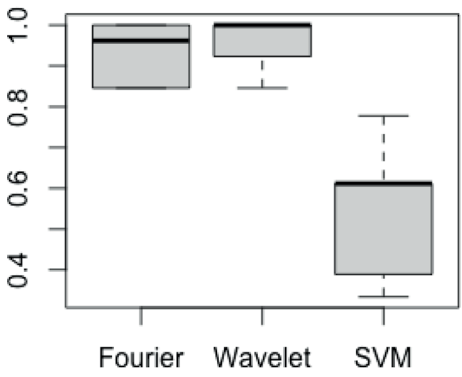


Fig. 6 Boxplots of three classification methods for the Duke breast cancer dataset

We conclude that the random wavelet kernel method performs better than random Fourier features kernel method in our real data study. To be specific, the WKM method has smaller prediction mean square errors for all four compositions of the biscuit, which are 0.401, 0.347, 0.348 and 0.378. The standard deviations of the WKM method are lower than those of the FKM method. These results are consistent with our simulation studies.

4.2 Real data for the classification problems

In this real data study for the classification problems, we use the Duke Breast Cancer database that consists of 86 tumour samples and 7129 genes. The data is numerical and has no missing values. The aim is to classify these tumour samples into estrogen receptor-positive (ER+) and estrogen receptor-negative (ER-) (west et al. 2001). We can access the dataset from the following website: https://www.kaggle.com/andreicosma/duke-breast-cancer-dataset.

We compare the results of three methods applied to the Duke Breast Cancer dataset: (1) random wavelet kernel method (WKM); (2) random Fourier features kernel method (FKM); (3) support vector machine (SVM) with Gaussian kernel. The results are shown in Table 5 and Fig. 6. We conclude that the method WKM performs the best among the three methods with an accuracy of 0.957. It is stable and accurate to use the random wavelet kernel function. SVM, the traditional method for nonlinear classification, has the lowest accuracy of 0.55 processing the large p small n dataset.

5 Conclusions

This article proposes the Bayesian approximate kernel method approximated by wavelet transform based on the framework of Bayesian approximate kernel regression (crawford et al. 2018). In particular, we combine wavelet analysis with random bases and use random wavelet bases to approximate the kernel function. The proposed method can lower the dimension. It is an efficient approach to deal with the large p small n problem. The performance of the kernel approximated by wavelet transform is better than that of the kernel approximated by Fourier transform when the data have significant fluctuations. We apply the proposed method to both regression and classification problems and compare the performance with other classical methods.

Numerical studies demonstrate that the Bayesian approximate kernel method approximated by wavelet transform outperforms the Bayesian approximate kernel method approximated by Fourier transform. We have smaller mean square errors solving regression problems and higher accuracy solving classification problems when using random wavelet bases to approximate the kernel function. It shows that the random wavelet bases method is more stable since its standard deviation of duplicates is small.

In conclusion, the Bayesian approximate kernel method approximated by wavelet transform has a good performance in regression and classification problems.

Acknowledgements Bei Jiang and Linglong Kong were partially supported by grants from the Canada CIFAR AI Chairs program, the Alberta Machine Intelligence Institute (AMII), and Natural Sciences and Engineering Council of Canada (NSERC), and Linglong Kong was also partially supported by grants from the Canada Research Chair program from NSERC. Yaozhong Hu was supported by the NSERC discovery fund and a centennial fund of the University of Alberta. The authors would like to thank the Editor, the Associate Editor and the two anonymous referees for the critical comments and constructive suggestions which have led to the improvement of this article.

References

Albert JH, Chib S (1993) Bayesian analysis of binary and polychotomous response data. J Am Statist Assoc 88:669–679

Aronszajn N (1950) Theory of reproducing kernels. Trans Am Math Soc 58:337–404

Banerjee S, Carlin BP, Gelfand AE (2003) Hierarchical modeling and analysis for spatial data. Chapman and Hall/CRC

Băzăvan EG, Li F, Sminchisescu C (2012) Fourier kernel learning. In European Conference on Computer Vision, Springer, pp. 459-473

Bernardo JM, Degroot MH, Lindley DV (1985) Bayesian Statistics 2: Proceedings of the Second Valencia International Meeting. North-Holland, 2: 371-372

Bernardo JM, Smith AF (2009) Bayesian theory. John Wiley & Sons, vol. 405

Brown PJ, Fearn T, Vannucci M (2001) Bayesian wavelet regression on curves with application to a spectroscopic calibration problem. J Am Statist Assoc 96:398–408

Chakraborty S (2009) Bayesian binary kernel probit model for microarray based cancer classification and gene selection. Comput Stat Data Anal 53:4198–4209

Chakraborty S, Ghosh M, Mallick BK (2012) Bayesian nonlinear regression for large p small n problems. J Multiv Anal 108:28–40

Crawford L, Wood KC, Zhou X, Mukherjee S (2018) Bayesian approximate kernel regression with variable selection. J Am Stat Assoc 113:1710–1721

DeCoste D, Mazzoni D (2003) Fast query-optimized kernel machine classification via incremental approximate nearest support vectors. In IEEE International Conference on Machine Learning (ICML), pp. 115-122

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning. Springer, Second edition

Lynch SM (2007) Introduction to applied Bayesian statistics and estimation for social scientists. Springer Science & Business Media

Mallick BK, Ghosh D, Ghosh M (2005) Bayesian classification of tumours by using gene expression data. J Royal Stat Soc Series B (Statistical Methodology) 67:219–234

Noble WS (2006) What is a support vector machine? Nat Biotechnol 24:1565–1567

Rahimi A, Recht B (2007) Random features for large-scale kernel machines. In Advances in Neural Information Processing Systems (NIPS), pp. 1177-1184

Schölkopf B, Herbrich R, Smola AJ (2001) A generalized representer theorem. In International Conference on Computational Learning Theory, Springer, pp. 416-426

Shyu H, Sun Y (2002) Construction of a morlet wavelet power spectrum. Multidimens Syst Signal Process 13:101–111

Sifuzzaman M, Islam MR, Ali M (2009) Application of wavelet transform and its advantages compared to fourier transform. J Phys Sci 13:121–134

Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10:988–999

Vapnik VN (2013) The nature of statistical learning theory. Springer Science & Business Media

Wahba G (1990) Spline models for observational data. SIAM, Philadelphia

Wang F, Du T (2000) Using principal component analysis in process performance for multivariate data. Omega 28:185–194

West M, Blanchette C, Dressman H, Huang E, Ishida S, Spang R, Zuzan H, Olson JA, Marks JR, Nevins JR (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. Proc Nat Acad Sci 98:11462–11467

Zhang L, Zhou W, Jiao L (2004) Wavelet support vector machine. IEEE Trans Syst Man Cybern Part B (Cybernetics) 34:34–39

Zhang N, Ding S (2017) Unsupervised and semi-supervised extreme learning machine with wavelet kernel for high dimensional data. Memetic Comput 9:129–139

Zhang Z, Dai G, Jordan MI (2011) Bayesian generalized kernel mixed models. J Mach Learn Res 12:111–139