On Random Subsampling of Gaussian Process Regression: A Graphon-Based Analysis

Kohei Hayashi
Preferred Networks, Inc.

Masaaki Imaizumi
Institute of Statistical Mathematics

Yuichi Yoshida
National Institute of Informatics

Abstract

In this paper, we study random subsampling of Gaussian process regression, one of the simplest approximation baselines, from a theoretical perspective. Although subsampling discards a large part of training data, we show provable guarantees on the accuracy of the predictive mean/variance and its generalization ability. For analysis, we consider embedding kernel matrices into graphons, which encapsulate the difference of the sample size and enables us to evaluate the approximation and generalization errors in a unified manner. The experimental results show that the subsampling approximation achieves a better trade-off regarding accuracy and runtime than the Nyström and random Fourier expansion methods.

1 Introduction

Gaussian process regression (GPR) is a fundamental tool for supervised learning. After learning parameters, we can make predictions in a distributional form, which is useful for measuring the uncertainty of the predictions. Of course, to enjoy such flexibility, we need to pay the price — computationally. For the number of samples n, both training (parameter learning) and the computation of the predictive distributions require polynomial time in n. The dominant part is the computation of the inverse of the n-by-n kernel matrix, which requires $O(n^3)$ time.

To reduce the time complexity, a lot of sophisticated approximation methods have been developed. Most of them introduce some structure into the kernel matrix to approximate it. For example, the Nyström method (Williams and Seeger, 2001) approximates the kernel matrix with a low-rank matrix. Given a shift-invariant kernel function, the random Fourier expansion (RFE) (Rahimi and Recht, 2008) approximately constructs a feature function in a finite-dimensional space. Several methods exploit specific properties of kernel matrices (Pleiss et al., 2018; Wilson and Nickisch, 2015).

A more drastic approach is *subsampling*, i.e., training GPR with a subset of the data. If we pick subsamples completely randomly, the time complexity only depends on the subsample size s, which is independent of n. With its simplicity and the computational cheapness, random subsampling has been seen as a baseline rather than a competitive method in the GP community (Rasmussen, 2004; Snelson, 2007; Quiñonero-Candela and Rasmussen, 2005). One of the main reasons is that it completely discards a large part of the data, and it seems impossible to estimate the uncertainties (Quiñonero-Candela and Rasmussen, 2005). Also, its theoretical justification is non-trivial, because subsampling changes the size of the kernel matrix. This is contrastive to the case of the structure-based approximations, which retain the size of the kernel matrix as n-by-n and can directly evaluate its approximation accuracy as the prediction accuracy, whereas they require at least $\Omega(n)$ computational cost in training.

In this paper, we study the subsampling approximation of GPR from a theoretical perspective. Our main results show that subsampling can maintain global information with a sufficiently small number of subsamples. In some special cases, $\Theta(s)$ subsamples guarantee $O(\log^{-1/4} s)$ prediction error at any new data point.

For analysis, we exploit the machinery of *graphons*, a continuous limit of bounded symmetric matrices, which have effectively been used in graph theory (see (Lovász, 2012)). Embedding the kernel matrices into graphons abstracts their difference in terms of the (sub)sample size, which enables to evaluate the predictive mean/variance without using strong statistical assumptions (Theorem 3.2). Because graphons

Proceedings of the 23^{rd}International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

can handle infinitely large matrices, i.e., the kernel matrices with $n \to \infty$, the result is immediately applicable to evaluate the generalization error (Corollary 3.4). Moreover, we show that, with a constant number of subsamples, hyperparameter tuning based on cross-validation (CV) succeeds with a high probability (Theorem 5.1). We performed experiments that provided encouraging results of subsampling in terms of the speed-accuracy trade-off.

2 Preliminaries

For an integer $n \in \mathbb{N}$, we denote the set $\{1, 2, \ldots, n\}$ by $[n]$. For $a, b \in \mathbb{R}$ and $c \in \mathbb{R}_+$, we mean $b-c \leq a \leq b+c$ by $a = b \pm c$.

For vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denotes their inner product. For a vector $\boldsymbol{x} \in \mathbb{R}^d$ and a set $S \subseteq [n]$, $\boldsymbol{x}_S \in \mathbb{R}^{|S|}$ denotes the vector obtained by restricting $\boldsymbol{x}$ to S. Similarly, for a matrix $A \in \mathbb{R}^{n \times m}$ and sets $S \subseteq [n]$ and $T \subseteq [m]$, $A_{ST} \in \mathbb{R}^{|S| \times |T|}$ denotes the matrix obtained by restricting A to $S \times T$. For a matrix $A \in \mathbb{R}^{n \times m}$, we define $\|A\|_{\max}$ as $\max_{i \in [n], j \in [m]} |A_{ij}|$. $\mathscr{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2.

2.1 Gaussian Process Regression

Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^p \times \mathbb{R}$ be training samples. The goal of the GPR is to obtain a predictive distribution for $f^*(\boldsymbol{x}^*)$ when a new sample $\boldsymbol{x}^* \in \mathbb{R}^p$ arrives. In this work, we consider the zero-mean GP prior for f with the covariance kernel function $k \colon \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. When the variance of the observation noise is specified as $\nu^2 > 0$, the predictive distribution for $f^*(\boldsymbol{x}^*)$ is given as the following Gaussian distribution:

$$\mathscr{N}\Big(\boldsymbol{k}^T (K + \nu^2 I)^{-1} \boldsymbol{y}, k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}^T (K + \nu^2 I)^{-1} \boldsymbol{k}\Big) \tag{1}$$

where $K \in \mathbb{R}^{n \times n}$ is the kernel matrix with $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ $(i, j \in [n])$ and $\boldsymbol{k} = (k(\boldsymbol{x}^*, \boldsymbol{x}_i))_{i \in [n]} \in \mathbb{R}^n$ (see Section 2 of (Rasmussen, 2004) for more details).

Let $\mathscr{H}$ be the reproducing kernel Hilbert space (RKHS) associated with $k(\cdot, \cdot)$. For a vector $\boldsymbol{x} \in \mathbb{R}^p$, let $\phi_{\boldsymbol{x}} = k(\boldsymbol{x}, \cdot) \in \mathscr{H}$ be the element corresponding to $\boldsymbol{x}$. Note that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi_{\boldsymbol{x}_i}, \phi_{\boldsymbol{x}_j} \rangle_{\mathscr{H}}$. We define a linear operator $\Phi \colon \mathbb{R}^n \to \mathscr{H}$ as $\Phi(\boldsymbol{w}) = \sum_{i \in [n]} \phi_{\boldsymbol{x}_i} w_i$.

2.2 Graphons and Matrices

A (measurable) bounded symmetric function $f \colon [0,1]^2 \to \mathbb{R}$ is called a *graphon*[1]. We can regard a graphon as a matrix in which the index is specified by a real value in $[0, 1]$. For two functions $f, g \colon [0, 1] \to \mathbb{R}$, we define their *inner product* as $\langle f, g \rangle = \int_0^1 f(x) g(x) dx$. We also define their *outer product* $fg^\top \colon [0,1]^2 \to \mathbb{R}$ as $fg^\top(x, y) = f(x) g(y)$. For a graphon $\mathscr{A} \colon [0,1]^2 \to \mathbb{R}$ and a function $f \colon [0, 1] \to \mathbb{R}$, we define the function $\mathscr{A} f \colon [0, 1] \to R$ as $(\mathscr{A} f)(x) = \langle \mathscr{A}(x, \cdot), f \rangle$.

For an integer $n \in \mathbb{N}$, let $I_1^n = [0, \frac{1}{n}]$, and for every $1 < k \leq n$, let $I_k^n = (\frac{k-1}{n}, \frac{k}{n}]$. For $x \in [0, 1]$, we define $i_n(x)$ as the unique integer $k \in [n]$ such that $x \in I_k^n$.

Definition 2.1. *Given a vector $\boldsymbol{v} \in \mathbb{R}^n$, we construct the corresponding function $\mathscr{v} \colon [0, 1] \to \mathbb{R}$ as $\mathscr{v}(x) = v_{i_n(x)}$. In addition, given a set of indices $S \subseteq [n]$, when we write $\mathscr{v}_S$, we first extract the vector $\boldsymbol{v}_S \in \mathbb{R}^{|S|}$ and then consider its corresponding function. Similarly, given a matrix $A \in \mathbb{R}^{n \times n}$, we construct the corresponding graphon $\mathscr{A} \colon [0,1]^2 \to \mathbb{R}$ as $\mathscr{A}(x, y) = A_{i_n(x) i_n(y)}$. In addition, given two sets of indices $S \subseteq [n]$ and $T \subseteq [n]$, when we write $\mathscr{A}_{ST}$, we first extract the matrix $A_{ST} \in \mathbb{R}^{|S| \times |T|}$ and then consider its corresponding graphon.*

For a graphon $\mathscr{A} \colon [0,1]^2 \to \mathbb{R}$, its *cut norm* is defined as

$$\|\mathscr{A}\|_\square = \max_{S, T \subseteq [0,1]} \Big| \int_S \int_T \mathscr{A}(x, y) \mathrm{d}x \mathrm{d}y \Big|,$$

where S and T run over all the measurable sets.

The following lemma states that we can approximate a matrix with its small submatrix with respect to the cut norm of the difference of their corresponding graphons.

Lemma 2.2 ((Hayashi and Yoshida, 2016))**.** *Let $L > 0$ and let $A^1, \ldots, A^T \in [-L, L]^{n \times n}$ be matrices. Let $S \subseteq [n]$ be a set of s elements that are uniformly selected at random. Then, with a probability of at least $1 - \exp(-\Omega(sT / \log_2 s))$, there exists a measure-preserving bijection $\pi \colon [0,1] \to [0,1]$ such that, for every $t \in [T]$, we have*

$$\|\mathscr{A}^t - \pi(\mathscr{A}^t_{SS})\|_\square = O\Big(L \sqrt{T / \log_2 s}\Big).$$

Moreover, we can assume $i_n(\pi(x)) = i_n(\pi(y))$ whenever $i_n(x) = i_n(y)$, that is, π is a block-wise bijection.

The following lemma states that the quadratic form of a graphon with a small cut norm is small.

Lemma 2.3 ((Hayashi and Yoshida, 2016))**.** *Let $\epsilon > 0$ and $\mathscr{A} \colon [0,1]^2 \to \mathbb{R}$ be a graphon with $\|\mathscr{A}\|_\square \leq \epsilon$. Then, for any functions $f, g \colon [0, 1] \to [-L, L]$, we have $|\langle f, \mathscr{A} g \rangle| \leq \epsilon L^2$.*

[1] Precisely speaking, such a function is called a *kernel* and a (measurable) symmetric function $f \colon [0,1]^2 \to [0, 1]$ is called a *graphon* in the literature. However, to avoid confusion with the kernel function $k(\cdot, \cdot)$, we adopt the term graphon here.

3 GPR with Graphons

The main purpose of GPR is to predict a function value at a new data point. The standard statistical result shows that, in a point-wise sense, the predictive mean converges to the true function as the sample size increases under some regularity conditions. In other words, the true function can be rephrased as the limit of the predictive mean of the GPR with infinitely many samples. The prediction accuracy (i.e., the generalization error) is therefore measured by the distance between the finite- and infinite-sample GPRs. However, analyzing the infinite-sample GPR is not trivial because we cannot write down the solution using standard matrix operations such as matrix inverse because the kernel matrix is infinitely large.

Graphons are a generic tool to handle both finite- and infinite-size matrices. First, a kernel matrix with infinitely many samples is embedded into a graphon by taking a map from the sample indices $[n]$ to $[0, 1]$. We can then reformulate the predictive distribution as the minimization problem of the quadratic objective function (i.e., the Gaussian log-likelihood of (1)) associated with the graphon. Also, a kernel matrix with a finite sample size is embedded into a graphon using the partition $I_1^n, \ldots, I_n^n$ defined in Section 2.2, which can be seen as the low-resolution version of the infinite one. Now, we can bound the difference between the finite- and infinite-sample objective values by using the distance between the two graphons in terms of the cut norm (using Lemma 2.3). The predictive accuracy is also derived in the same manner. We remark that the above approach can be used to analyze the difference between GPRs with different (finite) sample sizes, from which we can derive the accuracy of subsampling.

Using graphons and RKHSs to kernel methods have similar spirits: The kernel trick based on RKHSs provides an explicit form of the regression function when using the infinite-dimensional feature space whereas graphons provide an explicit form of that when using infinitely many samples.

3.1 Subsampled Predictive Distribution

First, we rephrase the predictive mean and variance. For a parameter $\lambda > 0$, we define a normalized loss function.

$$\ell_{K,\boldsymbol{k},\lambda}(\boldsymbol{v}) = \frac{1}{n}\left\|\frac{K\boldsymbol{v}}{n} - \boldsymbol{k}\right\|_2^2 + \frac{\lambda}{n^2}\langle \boldsymbol{v}, K\boldsymbol{v}\rangle \tag{2}$$

By setting $\lambda = \nu^2/n$ and with the solution

$$\boldsymbol{v}^* = \operatorname*{argmin}_{\boldsymbol{v}\in\mathbb{R}^n} \ell_{K,\boldsymbol{k},\lambda}(\boldsymbol{v}) = n(K + n\lambda I)^{-1}\boldsymbol{k}, \tag{3}$$

the predictive mean and variance in (1) can be rewritten as

$$\mu_{\boldsymbol{x}^*} := \frac{\langle \boldsymbol{v}^*, \boldsymbol{y}\rangle}{n} \quad \text{and} \quad \sigma^2_{\boldsymbol{x}^*} := k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \frac{\langle \boldsymbol{v}^*, \boldsymbol{k}\rangle}{n}. \tag{4}$$

Algorithm 1 Approximate solver for the normalized loss

```
Require: $n, s \in \mathbb{N}$, $\lambda > 0$, and query accesses to $K \in \mathbb{R}^{n\times n}$ and $\boldsymbol{k} \in \mathbb{R}^n$.
  Sample a set $S \subseteq [n]$ of size $s$ chosen uniformly at random.
  $\widetilde{\boldsymbol{v}}^* \leftarrow \operatorname{argmin}_{\widetilde{\boldsymbol{v}}} \ell_{K_{SS},\boldsymbol{k}_S,\lambda}(\widetilde{\boldsymbol{v}})$.
  return $\widetilde{\boldsymbol{v}}^*$ and $S$.
```

Algorithm 2 Approximation algorithm for predictive mean and variance

```
Require: $n, s \in \mathbb{N}$, $\lambda > 0$, and query accesses to $K \in \mathbb{R}^{n\times n}$, $\boldsymbol{k} \in \mathbb{R}^n$, and $\boldsymbol{y} \in \mathbb{R}^n$.
  Run Algorithm 1 to obtain $\widetilde{\boldsymbol{v}} \in \mathbb{R}^s$ and a subset $S \subseteq [n]$ of size $s$.
  $\widetilde{\mu}_{\boldsymbol{x}^*} \leftarrow \langle \widetilde{\boldsymbol{v}}, \boldsymbol{y}_S\rangle / s$.
  $\widetilde{\sigma}^2_{\boldsymbol{x}^*} \leftarrow k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \langle \widetilde{\boldsymbol{v}}, \boldsymbol{k}_S\rangle / s$.
  return $(\widetilde{\mu}_{\boldsymbol{x}^*}, \widetilde{\sigma}^2_{\boldsymbol{x}^*})$.
```

In what follows, we leave λ as a parameter as we often do not know the value of ν.

Our algorithm consists of two parts. The first part of our algorithm (Algorithm 1) approximately minimizes (2). For a small integer $s \in \mathbb{N}$, it samples a set $S \subseteq [n]$ of size s uniformly at random and then minimizes the function obtained by restricting (2) to S, that is, $\ell_{K_{SS},\boldsymbol{k}_S,\lambda}$. Here, we assume that the matrix K and vector $\boldsymbol{k}$ are given through query accesses. That is, if we specify the indices $i, j \in [n]$, we can obtain K_{ij} in constant time, and similarly, if we specify an index $i \in [n]$, we can obtain k_i in constant time. The second part of our algorithm (Algorithm 2) computes approximations for $\mu_{\boldsymbol{x}^*}$ and $\sigma^2_{\boldsymbol{x}^*}$ using the vector obtained in the first part.

For the first part of our algorithm, we show the following guarantee, which states that the minima of $\ell_{K,\boldsymbol{k},\lambda}$ and $\ell_{K_{SS},\boldsymbol{k}_S,\lambda}$ are close. The proof for this is deferred to Supplementary material.

Theorem 3.1. *For any $\epsilon > 0$, Algorithm 1 with $s = 2^{\Theta(1/\epsilon^2)}$ outputs $\widetilde{\boldsymbol{v}}^* \in \mathbb{R}^s$ such that*

$$\ell_{K_{SS},\boldsymbol{k}_S,\lambda}(\widetilde{\boldsymbol{v}}^*) = \ell_{K,\boldsymbol{k},\lambda}(\boldsymbol{v}^*) \pm O\Big(\epsilon L^2 R^2\Big)$$

with a probability of at least 0.99, where $\boldsymbol{v}^* = \operatorname{argmin}_{\boldsymbol{v}} \ell_{K,\boldsymbol{k},\lambda}(\boldsymbol{v})$, $L = \max\{\|K\|_{\max}, \|\boldsymbol{k}\|_\infty\}$, and $R = \max\{\|\boldsymbol{v}^*\|_\infty, \|\widetilde{\boldsymbol{v}}^*\|_\infty\}$.

For the second part of our algorithm, we show the following guarantee, which states that the approximations computed using Algorithm 2 are accurate. The proof for this is presented in Appendix C.

Theorem 3.2. *Let $\|\boldsymbol{z}\|_\phi := \inf\{\|\boldsymbol{\alpha}\|_{\mathscr{H}} : \boldsymbol{\alpha} \in \mathscr{H}, \forall_i z_i = \langle \phi_{\boldsymbol{x}_i}, \boldsymbol{\alpha}\rangle_{\mathscr{H}}\}$ be the norm of $\boldsymbol{z} \in \mathbb{R}^n$ in the feature space spanned by $\{\phi_{\boldsymbol{x}_i}\}_{i\in[n]}$. For any $\epsilon > 0$, Algorithm 2 with $s = 2^{\Theta(1/\epsilon^2)}$ and $\lambda = \Theta(1)$ outputs $(\widetilde{\mu}_{\boldsymbol{x}^*}, \widetilde{\sigma}^2_{\boldsymbol{x}^*})$ such that*

$$|\mu_{\boldsymbol{x}^*} - \widetilde{\mu}_{\boldsymbol{x}^*}| = O\left(\sqrt{\epsilon} L^2 R\right) \text{ and } |\sigma^2_{\boldsymbol{x}^*} - \widetilde{\sigma}^2_{\boldsymbol{x}^*}| = O\left(\sqrt{\epsilon} L^2 R\right),$$

with probability of at least 0.99, where $L = \max\{\|K\|_{\max}, \|\boldsymbol{k}\|_\infty, k(\boldsymbol{x}^*, \boldsymbol{x}^*), \|\boldsymbol{y}\|_\phi\}$ and $R = \max\{\|\boldsymbol{v}^*\|_\infty, \|\widetilde{\boldsymbol{v}}^*\|_\infty\}$.

Remark 3.3. Theorem 3.2 suggests that the prediction error of subsampling is explained by L and R. Although L and R can increase with n, they are regarded as constants in some cases. For L, we see that $\|K\|_{\max}$, $\|\boldsymbol{k}\|_\infty$, and $k(\boldsymbol{x}^*)$ are $O(1)$ when the kernel function is bounded. For the polynomial or linear kernel, this can be achieved by normalizing the input $\boldsymbol{x} \in \mathbb{R}^p$ so that each element is within $[-1/p, 1/p]$. The condition $\|\boldsymbol{y}\|_\phi = O(1)$ is typically admissible in the noiseless case. We can upper bound R when the matrix K is strictly diagonally dominant. Let $M = (K + n\lambda I)$. Then by the Ahlberg-Nilson-Varah bound (see, e.g., (Morača, 2008)), we have $\|M^{-1}\|_\infty \le 1/\min_i(n\lambda - \Sigma_{j\neq i} M_{ij}) \approx O(1/n\lambda)$. Hence, we have $\|\boldsymbol{v}^*\|_\infty \le n\|M^{-1}\|_\infty \|k\|_\infty = O(L/\lambda)$.

Let us elaborate on the condition $\|y\|_\phi = O(1)$. Intuitively, $\|y\|_\phi$ measures the complexity of the mapping from the training inputs x to the outputs y using the RKHS norm $\|\cdot\|_{\mathscr{H}}$ (note that the RKHS consists of functions with $\|f^*\|_{\mathscr{H}} < \infty$; for details, see the book (Steinwart and Christmann, 2008, Chapter 4)). For example, if the true function f^* is a member of $\mathscr{H}$ and $y_i = f^*(x_i)$ for every i (i.e., y is noiseless), then we have $\|y\|_\phi = \|f^*\|_{\mathscr{H}}$, which is independent of the number of samples n as the norm solely depends on f^*. Hence we obtain $\|y\|_\phi = O(1)$. If y contains noise or the mapping from x to y is not a member of $\mathscr{H}$, then we cannot say $\|y\|_\phi = O(1)$ in general. [2]

3.2 Generalization Error

We provide generalization analysis for the subsampling method, namely, we investigate how our method estimates an unknown data generating process. To this end, let us assume that the samples are generated through a function $f^* \colon \mathbb{R}^p \to \mathbb{R}$ that relates y_i and $\boldsymbol{x}_i$ as

$$y_i = f^*(\boldsymbol{x}_i) + \xi_i, \ i \in [n], \tag{5}$$

where $\xi_i \sim \mathscr{N}(0, \nu^2)$ is the Gaussian noise.

We note that Theorem 3.2 holds for any sample size n, even at the limit $n \to \infty$. It is well known that universal kernel functions (e.g., the Gaussian kernel and the polynomial kernel) can approximate any continuous function (Micchelli et al., 2006; Steinwart and Christmann, 2008), and several kernel-based estimators converge to any truth functions f^* at $n \to \infty$ (Györfi et al., 2006; Rasmussen, 2004). The result also holds with the GP regression estimator (van der Vaart and van Zanten, 2008) with some assumptions. Along with these results, Theorem 3.2 can be used to bound the generalization error.

Corollary 3.4. *Consider the same setting as in Theorem 3.2 and assume that the observations follow the model (5). Suppose $\mu_{\boldsymbol{x}^*}$ is a consistent estimator for $f^*(\boldsymbol{x}^*)$, namely, $|\mu_{\boldsymbol{x}^*} - f^*(\boldsymbol{x}^*)| \to 0$ as $n \to \infty$. Then, with probability at least 0.98, the following holds:*

$$|\widetilde{\mu}_{\boldsymbol{x}^*} - f^*(\boldsymbol{x}^*)| = O\left(\frac{L'^2 R}{\log^{1/4} s}\right),$$

where $L' = \max\{\|K\|_{\max}, \|\boldsymbol{k}\|_\infty, k(\boldsymbol{x}^*, \boldsymbol{x}^*), \|f^*\|_{\mathscr{H}}\}$. Furthermore, if $k(\boldsymbol{x}, \boldsymbol{x})$ and $\mu_{\boldsymbol{x}}$ are bounded for all $\boldsymbol{x}$, then, with probability at least 0.98, the following holds:

$$\|\widetilde{\mu}_\cdot - f^*\|_{L^2} = O\left(\frac{L'^2 R}{\log^{1/4} s}\right),$$

where $\|\cdot\|_{L^2}$ denotes the L^2-norm for square integrable functions.

Although Corollary 3.4 only guarantees a relatively slow rate of $O(\log^{-1/4} s)$, besides the consistency assumption on $\mu_{\boldsymbol{x}^*}$, it does not require any other assumption such as the differentiability of f^*.

4 Related Work

4.1 Approximation Analysis

Subsampling-based approximations are known as the subset of the data (SD) methods, which has several variants in terms of how the subsamples are chosen (Quiñonero-Candela and Rasmussen, 2005). The simplest version chooses samples completely randomly, which is equivalent to our algorithms except that the simplest SD method fixes the noise variance ν^2, independently of the subsample size s, whereas ours scales ν^2 to derive a theoretical guarantee on its accuracy. Other SD methods select subsamples based on more sophisticated criteria such as the differential entropy

[2] It is possible to analyze the case when $f^* \notin \mathscr{H}$, and achieve a new error bound by using the techniques of interpolation spaces (Section 5.6 in Steinwart and Christmann (2008)) or embedding operators (Theorem 4.6 in Steinwart and Scovel (2012)). However, it is not our main focus.

Table 1: Comparison of approximation methods for GPR.

Method	Time Complexity	Predictive Mean Error	Predictive Variance Error	Assumptions
Nyström	$O(ns^2)$ or $O(ns^2+s^3)$	$\widetilde{O}(s^{-\gamma})$	N/A	Incoherence
RFE	$O(ns^2)$	$\widetilde{O}(s^{-1/2})$	N/A	Restriction on kernels
Lanczos	$O(n+s)$	N/A	N/A	None
Subsampling	$O(s^3)$	$O(\log^{-1/4} s)$	$O(\log^{-1/4} s)$	Remark 3.3

score (Herbrich et al., 2003), which requires, however, $O(n)$ time as it scans all the samples.

The inducing points methods (Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Titsias, 2009) are another class of approximation methods, which picks up a small number of auxiliary variables as pseudo-samples—inducing points—and approximate the predictive mean using the cross-covariance between the inducing points and the rest of the samples. The inducing points are usually chosen based on the marginal likelihood (Snelson and Ghahramani, 2006) or the variational principle (Titsias, 2009; Hensman et al., 2013). Although they perform well in practice (Matthews, 2017), their time complexity depends on n due to computing the cross-covariance. Also, to the best of our knowledge, their theoretical properties, especially the approximation accuracy, have not been studied.

The Nyström method and its variants such as the leverage score method are also intensively studied (Alaoui and Mahoney, 2015; Bauer et al., 2016; Gittens and Mahoney, 2016; Musco and Musco, 2017; Williams and Seeger, 2001). They employ $s < n$ points as regressors and their time complexity is typically $O(ns^2)$. Assuming that the selected regressors have a nice property such as incoherence, their approximation error for the predictive mean is $\widetilde{O}(s^{-\gamma})$, which follows from the approximation guarantee for the kernel matrix in the spectral norm (Musco and Musco, 2017). Here, $\gamma > 0$ is a parameter depending on the kernel function.

RFE approximates predictors by using s Fourier bases (Avron et al., 2017; Sriperumbudur and Szabó, 2015; Yang et al., 2012), which requires $O(ns^2)$ time and some restriction on kernel functions such as shift-invariance. The approximation error for the predictive mean is $\widetilde{O}(s^{-1/2})$, which follows from the error analysis for the kernel matrix (Yang et al., 2012). Also, some other works (Yang et al., 2012; Sriperumbudur and Szabó, 2015) analyzed its generalization capability.

Pleiss et al. (2018) developed Lanczos approximation. The time complexity is $O(n+s)$, where s is the number of inducing points. No theoretical guarantee is known.

Time complexity: Note that subsampling requires only $O(s^3)$ time, which is $O(1)$ when $s = O(1)$. In contrast, all the other methods depend on n, and hence they cannot be run in $O(1)$ time.

Error bound: As for the error bound, recalling the relation $s = 2^{\Theta(1/\epsilon^2)}$ in Theorem 3.2, subsampling has a convergence rate of $O(\log^{-1/4} s)$, which is slower than the polynomial rates achieved by the Nyström method and RFE. However, we stress here that, at the cost of the slow convergence rate, we eliminated several assumptions used in their analysis. More specifically, the Nyström method requires that the subsampled regressors are incoherent and the RFE require that the kernel function is shift-invariant. Also, we can provide an error bound for the predictive variance, which has not been addressed in the Nyström, RFE, or Lanczos methods.

Table 1 summarizes our theoretical results for the subsampling method against those for other approximation methods.

4.2 Generalization Analysis

Some existing studies have developed generalization theory of GPR. van der Vaart and van Zanten (2008, 2011) evaluated GPR by using the notion of *posterior contraction*, and showed that the generalization error measured by the L^2-norm is

$$O\left(n^{-2\beta/(2\beta+p)}\right), \tag{6}$$

where β is the number of differentiability of f^*. For different metrics such as the L^∞-norm, the same rates (up to logarithmic factors) were obtained (Giné et al., 2011; Yoo et al., 2016).

Our generalization analysis (Corollary 3.4) provides the rate of $O(\log^{-1/4} n)$, which is much slower than (6). Nevertheless, this rate only requires the consistency of $\mu_{\boldsymbol{x}^*}$ and does not impose any assumption on f^*, while the existing rate (6) assumes the differentiability of f^*.

5 Hyperparameter Selection

GPR has several hyperparameters such as λ in (2) and hyperparameters used in kernel functions, e.g., the bandwidth $h > 0$ in the Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-h^{-1}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2)$ and the parameters h, a, b in the polynomial kernel $k(\boldsymbol{x}, \boldsymbol{x}') = (h^{-1}\langle \boldsymbol{x}, \boldsymbol{x}'\rangle + b)^a$. Cross validation (CV) (Geisser, 1975; Stone, 1974) is a popular approach for selecting such hyperparameters, although it is computationally expensive. In this section, we show that we can circumvent this issue by using our method (Algorithm 2).

Let θ be the set of hyperparameters, e.g., $\theta = (\lambda, h)$ for the Gaussian kernel. We consider a predictor $\widehat{f}_{S,\theta}(\boldsymbol{x}^*)$, which is the predictive mean obtained when we run Algorithm 2 on $\boldsymbol{x}^* \in \mathbb{R}^p$ with hyperparameters θ and the index set $S \subseteq [n]$ of size $s \in \mathbb{N}$. Furthermore, let $f_\theta^0(\boldsymbol{x}) := \mu_{\boldsymbol{x}^*}$ be the predictive mean using all the n samples. For any θ, we assume that f^*, $\widehat{f}_{S,\theta}$, and f_θ^0 are bounded and have finite second moments, i.e., $B := \max\{\|f^*\|_\infty, \|f_\theta^0\|_\infty, \|\widehat{f}_{S,\theta}\|_\infty\}$ and $B_\sigma := \max\{\|f^*\|_2^2, \|f_\theta^0\|_2^2, \|\widehat{f}_{S,\theta}\|_2^2\}$ are finite. These assumptions are standard and easy to verify for bounded kernels (Section 4 of (Steinwart and Christmann, 2008) presents detailed discussions).

We want to compute the expected loss of the (original) predictive mean $\mathrm{EL}(\theta) := \mathbb{E}_{\boldsymbol{x}}[(f^*(\boldsymbol{x}) - f_\theta^0(\boldsymbol{x}))^2]$ for a given θ and then select the best θ. To this end, in the CV, we first sample an index set $Q \subseteq [n]\backslash S$ of size $q \leq n - s$ uniformly at random. We then define the CV loss as

$$\mathrm{CV}_Q(\widehat{f}_{S,\theta}) := \frac{1}{q}\sum_{i \in Q}(y_i - \widehat{f}_{S,\theta}(\boldsymbol{x}_i))^2. \tag{7}$$

Now, we evaluate the selection performance of the CV based on Algorithm 2. For simplicity, we assume that we have two candidates for the choice of hyperparameters, θ_1 and θ_2. Then, we have the following:

Theorem 5.1. *Suppose that* $\mathrm{EL}(\theta_1) + \Xi < \mathrm{EL}(\theta_2)$ *holds for some* $\Xi > 0$. *Let us define* $\omega(s)$ *as the upper bound on* $|\mu_{\boldsymbol{x}^*} - \widetilde{\mu}_{\boldsymbol{x}^*}|$ *given in Theorem 3.2, and a parameter* $\widetilde{\Xi}(s) := \Xi - 3\omega(s)^2 - 2\omega(s)B - \nu^2(4B + \omega(s))$. *Then for any* $s, q \geq 1$,

$$\mathrm{CV}_Q(\widehat{f}_{S,\theta_1}) \leq \mathrm{CV}_Q(\widehat{f}_{S,\theta_2}),$$

holds, with probability at least

$$1 - 4\exp\left(-\frac{q}{2}\left(\frac{\widetilde{\Xi}(s)}{B^2} - \frac{9B_\sigma^2}{B^4}\right)\right) - \frac{3\nu^2(4B + 2\omega(s))}{q\widetilde{\Xi}(s)}.$$

Note that $\widetilde{\Xi}$ is an increasing function in Ξ and s. Hence, Theorem 5.1 implies that the probability that the approximated CV succeeds increases as q, s, and Ξ increase.

Note that we can easily extend the argument to the case we have a finite number of candidates. Also note that selecting the value of a hyperparameter by CV from a finite set of candidates is commonly used in GP (see Rasmussen (2004, Chapter 5)).

6 Experiments

6.1 Approximation Accuracy

First, we evaluated the performance of subsampling with a constant number of samples that are covered by our theory, that is, the predictive mean/variance (4), the minimum of the normalized loss function (2), and the CV error (7). Here, we used five real datasets[3] whose sample sizes are in thousands such that we could run the exact GPR for comparison. Each data set was standardized so that y and each feature of $\boldsymbol{x}$ are ranged in $[-1, 1]$.

The upper part of Figure 1 shows the contour of the 10-fold CV error with the Gaussian kernel. In datasets `housing` and `mg`, subsampling successfully selected the hyperparameters that were sufficiently close to the ones selected by the full-sample CV. In `abalone` and `cpusmall`, the selected hyperparameters look far. However, this was because the landscape of the full-sample CV error was flat (the lower part of Figure 1) and it was difficult to choose the optimal hyperparameters even in the original CV. Indeed, this case corresponds to the case that Ξ in Theorem 5.1 is small, and these empirical results agree with the claim of Theorem 5.1: the hyperparameter selection may fail for small Ξ.

Figure 2 shows the errors of the predictive mean $|\mu_{\boldsymbol{x}^*} - \widetilde{\mu}_{\boldsymbol{x}^*}|$, predictive variance $|\sigma_{\boldsymbol{x}^*}^2 - \widetilde{\sigma}_{\boldsymbol{x}^*}^2|$, and the objective $|\ell_{K,\boldsymbol{k},\lambda}(\boldsymbol{v}^*) - \ell_{K_{SS},\boldsymbol{k}_S,\lambda}(\widetilde{\boldsymbol{v}}^*)|$ with the Gaussian kernel. We see that the errors, especially of the predictive mean and variance, decrease faster than we expect from the theoretical convergence rate of $O(\log^{-1/4} s)$ shown in the dashed lines.

We also investigated how the choice of kernel functions affects the approximation quality. Figure 3 shows a similar behavior as in Figure 2 no matter which kernel function is used. We observe that all kernel functions behave very similarly, meaning subsampling works independently of the choice of the kernel function as our theory suggested. Due to the page limitation, we only show the result with a single data set here; see Appendix E for the complete results.

[3] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`

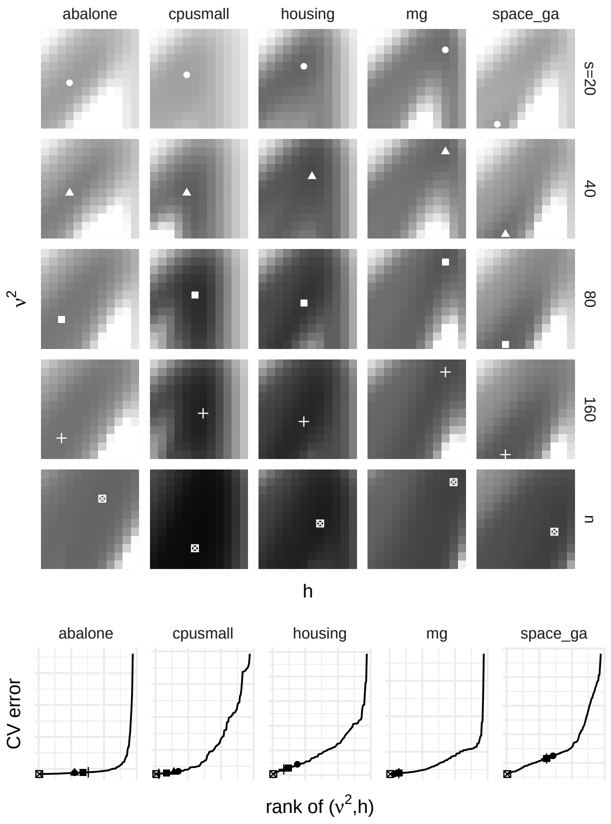

Figure 1: Top: Contours of the CV error for the noise variance ν^2 and the kernel bandwidth h in a logarithmic scale ($\nu^2, h^{-1} \in \{10^{-i/3} \mid i = 0, 1, \ldots, 11\}$). White dots indicate the selected hyperparameters by CV (i.e., the minima of the contours). Bottom: CV error for each pair (ν^2, h) in increasing order. The selected (ν^2, h) with other s are also shown as black dots with the same shape as above.

6.2 Prediction Accuracy and Runtime

Next, we compared the prediction performance. Specifically, we are interested in the trade-off between the prediction accuracy on the test data (i.e., the generalization power) and the runtime. To this end, we prepared several large-scale datasets: `cadata` ($n \simeq$ 20K), `YearPredictionMSD` ($\simeq$ 0.4M), `covtype` ($\simeq$ 0.6M), `SUSY` ($\simeq$ 5M), and `Airline` ($\simeq$ 7M).[4] After standardizing both input and output, we split each data set into a test set consisting of $1,000$ randomly selected samples and a training set consisting of the rest of the samples. As baselines, we prepared the Nyström method, RFE, and stochastic variational inference GP (SVI-GP) (Hensman et al., 2013)—an inducing points method based on the variational principle. All the methods were implemented in Python (we used GPytorch (Gardner et al., 2018) for SVI-GP). SVI-GP determined hyperparameters by variational inference; for the other methods we conducted 3-fold CV before training. The runtime was recorded with 32-core Intel Xeon 3.20GHz.

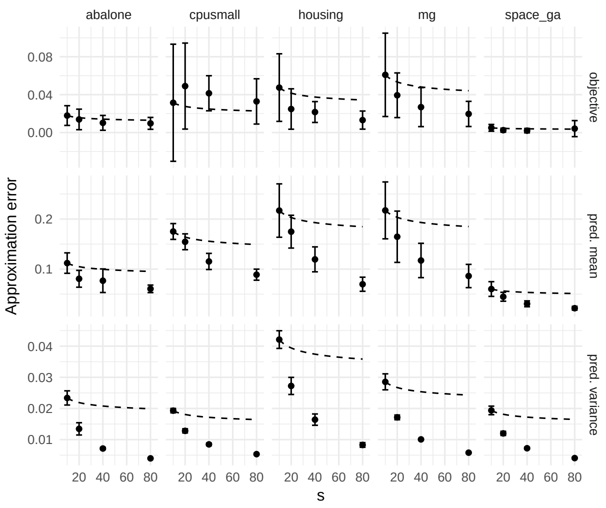

Figure 2: Errors of the predictive mean/variance and the normalized loss function with the Gaussian kernel. The hyperparameters were set as the noise variance $\nu^2 = 0.01$ and the bandwidth $h = 10$. The error bars indicate the standard deviation of the results over ten trials with different random seeds. The dashed lines indicate the theoretical bounds (Theorems 3.1 and 3.2) where we set unknown linear coefficients of the bounds as they fit the results.

Figure 4 shows the trade-off curves between the test error and the total runtime for training including hyperparameter selection. Overall, subsampling achieved better trade-off than the other methods. In `cadata`, `YearPredictionMSD`, and `SUSY`, subsampling mostly dominated the Pareto frontier, meaning that it outperformed the others in terms of both runtime and accuracy. In `airline`, on the other hand, RFE and SVI-GP achieved better accuracy. This might be because of the high-variance of the output $\boldsymbol{y}$. The task of `airline` is to predict the delays of flights in minutes, in which the distribution is skewed and heavy tailed (Bandyopadhyay and Guerrero, 2012). The high variance of $\boldsymbol{y}$ can increase the prediction error (see Remark 3.3).

7 Discussion

In this work, we explored the theoretical aspects of random subsampling of GPR. Using the tool of

[4] The labels of `covtype` and `SUSY` were binary but we regarded them as real values and solved as regression problems.

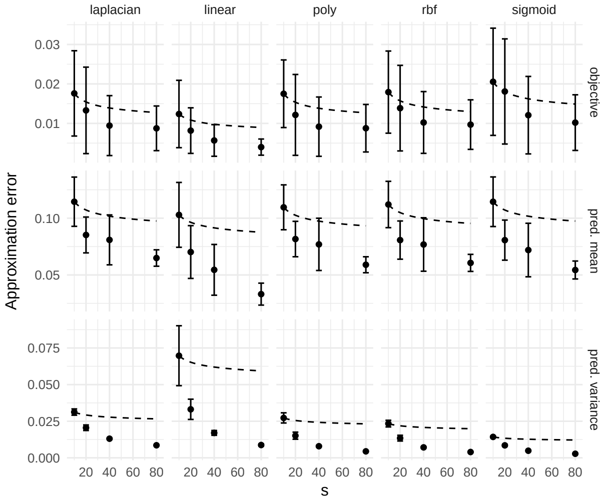

Figure 3: Approximation errors with the Laplacian, linear, polynomial, Gaussian (RBF), and sigmoid kernel functions on `abalone` data set. The hyperparameters were set as the noise variance $\nu^2 = 0.01$ and the bandwidth $h = 10$. Other kernel parameters were fixed as the default values of scikit-learn library (Pedregosa et al., 2011)

graphons, we built the error bounds for the predictive distribution and generalization. Although the derived rates are slower than other structure-based approximations, they only require minimum assumptions. The experimental results demonstrated that in many cases subsampling achieved a better runtime-accuracy trade-off than the Nyström, RFE, and SVI-GP methods. Combining the theoretical and empirical results, we conclude that subsampling is worth a try as well as more other complicated approximations.

The empirical results (Figures 2 and 3) repeatedly indicate that the actual performance of subsampling is far better than theoretically expected. This would be because the derived bounds (Theorem 3.2 and Corollary 3.4) are too conservative. Actually, they consider almost worst-case scenarios, such as the truth function is peaky everywhere or drawn subsamples are densely collected in a small input area. Adding some realistic assumptions such as smoothness may help to derive better error bounds.

We note that our methodology has less restrictions than the Nyström method and RFE. Specifically, the Nyström method requires that the Gram matrices are incoherent (Drineas and Mahoney, 2005). Also, RFE works only on shift-invariant kernels (Rahimi and Recht, 2008) and cannot be applied to polynomial and sigmoid kernels, which are commonly used. In contrast, our methodology is free from these restrictions.

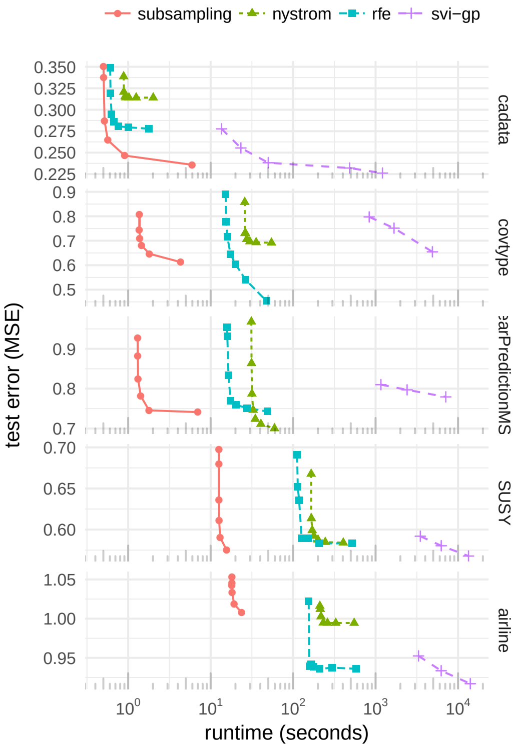

Figure 4: Comparison of runtime-accuracy trade-off. We changed the subsample size over $\{10 \cdot 2^i \mid i \in \{4, \ldots, 9\}\}$ for subsampling, and changed the number of basis over $\{10 \cdot 2^i \mid i \in \{1, \ldots, 7\}\}$ for the other methods. We stopped the training when the runtime exceeded 12 hours and those results are not shown.

We have shown that the CV strategy well admits subsampling (Section 5), but we may want to use subsampling to approximate other criteria. The marginal likelihood would be the most popular criterion in the GP community for hyperparameter selection (Rasmussen, 2004). Unfortunately, our analysis is not immediately applicable to approximating it. Let us explain why. The marginal likelihood has the explicit form of $\log\det(K + n\lambda I) + \langle \boldsymbol{y}, (K + n\lambda I)^{-1}\boldsymbol{y}\rangle + n/2 \log 2\pi$. The second term has the quadratic form as we have already seen (e.g., Eq. 3) and indeed subsampling can approximate it. The difficulty is in the first term, which we have to deal with the determinant of the kernel matrix. Remember that we treat the kernel matrix as the graphon in our analysis. However, the determinant of the graphon is not well-defined, meaning that we cannot compare kernel matrices with different sample size, and therefore, the approximation accuracy remains unknown. Further investigation on the marginal likelihood approximation is one of our future works.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 18H05291.

References

A. Alaoui and M. W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.

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

R. J. Bandyopadhyay and R. Guerrero. Predicting airline delays, 2012.

M. Bauer, M. van der Wilk, and C. E. Rasmussen. Understanding probabilistic sparse gaussian process approximations. In *Advances in neural information processing systems*, pages 1533–1541, 2016.

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

J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

E. Giné, R. Nickl, et al. Rates of contraction for posterior distributions in lr-metrics, $1 \leq r \leq \infty$. *The Annals of Statistics*, 39(6):2883–2911, 2011.

A. Gittens and M. W. Mahoney. Revisiting the nyström method for improved large-scale machine learning. *The Journal of Machine Learning Research*, 17(1):3977–4041, 2016.

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression.* Springer Science & Business Media, 2006.

K. Hayashi and Y. Yoshida. Minimizing quadratic functions in constant time. In *NIPS*, pages 2217–2225, 2016.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

R. Herbrich, N. D. Lawrence, and M. Seeger. Fast sparse gaussian process methods: The informative vector machine. In *Advances in neural information processing systems*, pages 625–632, 2003.

L. Lovász. *Large Networks and Graph Limits.* American Mathematical Society, 2012.

A. G. d. G. Matthews. *Scalable Gaussian process inference using variational methods.* PhD thesis, University of Cambridge, 2017.

C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec): 2651–2667, 2006.

N. Morača. Bounds for norms of the matrix inverse and the smallest singular value. *Linear Algebra and its Applications*, 429(10):2589–2601, 2008.

C. Musco and C. Musco. Recursive sampling for the nystrom method. In *Advances in Neural Information Processing Systems*, pages 3833–3845, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

G. Pleiss, J. R. Gardner, K. Q. Weinberger, and A. G. Wilson. Constant-time predictive distributions for gaussian processes. *CoRR*, abs/1803.06058, 2018.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2008.

C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

E. L. Snelson. *Flexible and efficient Gaussian process models for machine learning.* PhD thesis, UCL (University College London), 2007.

B. Sriperumbudur and Z. Szabó. Optimal rates for random fourier features. In *Advances in Neural Information Processing Systems*, pages 1144–1152, 2015.

I. Steinwart and A. Christmann. *Support vector machines.* Springer Science & Business Media, 2008.

I. Steinwart and C. Scovel. Mercers theorem on general domains: on the interaction between measures, kernels, and rkhss. *Constructive Approximation*, 35 (3):363–417, 2012.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B*, pages 111–147, 1974.

M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

A. van der Vaart and H. van Zanten. Rates of contraction of posterior distributions based on gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463, 2008.

A. van der Vaart and H. van Zanten. Information rates of nonparametric gaussian process methods. *Journal of Machine Learning Research*, 12(Jun):2095–2119, 2011.

C. K. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *NIPS*, pages 682–688, 2001.

A. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *ICML*, pages 1775–1784, 2015.

T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012.

W. W. Yoo, S. Ghosal, et al. Supremum norm posterior contraction and credible sets for nonparametric multivariate regression. *The Annals of Statistics*, 44 (3):1069–1102, 2016.