Probabilistic Approximate Least-Squares

Simon Bartels **Philipp Hennig**
Max Planck Institute for Intelligent Systems
Tübingen Germany

Abstract

Least-squares and kernel-ridge / Gaussian process regression are among the foundational algorithms of statistics and machine learning. Famously, the worst-case cost of exact nonparametric regression grows cubically with the data-set size; but a growing number of approximations have been developed that estimate good solutions at lower cost. These algorithms typically return point estimators, without measures of uncertainty. Leveraging recent results casting elementary linear algebra operations as probabilistic inference, we propose a new approximate method for nonparametric least-squares that affords a probabilistic uncertainty estimate over the error between the approximate and exact least-squares solution (this is not the same as the posterior variance of the associated Gaussian process regressor). This allows estimating the error of the least-squares solution on a subset of the data relative to the full-data solution. The uncertainty can be used to control the computational effort invested in the approximation. Our algorithm has linear cost in the data-set size, and a simple formal form, so that it can be implemented with a few lines of code in programming languages with linear algebra functionality.

1 INTRODUCTION

The least-squares estimation of a regression function from a reproducing kernel Hilbert space (RKHS) is one of the foundational algorithms of statistics and machine learning. Partly as a result of this central role, it is known under a plurality of names, including kernel ridge regression (Hoerl and Kennard, 1970), spline regression (Wahba, 1990), and Kriging (Matheron, 1973). All of these terms refer to a unique real-valued function $f : \mathbb{X} \rightarrowtriangle \mathbb{R}$ over some domain $\mathbb{X}$: The element of the reproducing kernel Hilbert space of a kernel k over $\mathbb{X}$ that minimizes the regularized loss

$$\mathcal{L}(f) = \|f\|_k^2 + \sigma^{-2} \sum_{i=1}^{N} \|y_i - f(x_i)\|_2^2, \qquad (1)$$

where $(x_i, y_i) \in \mathbb{X} \times \mathbb{R}, i = 1, \ldots, N$ are input and output labels of some dataset, $\sigma \in \mathbb{R}$ is a parameter, and $\|\cdot\|_k$ is the associated norm of the RKHS of k. Equivalently, this function is also the posterior mean of the Gaussian process arising from the prior $p(f) = \mathcal{GP}(f; 0, k)$ with covariance function k and the likelihood $p(\boldsymbol{y} \,|\, \boldsymbol{f}(\boldsymbol{x})) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{f}(\boldsymbol{x}), \sigma^2 \boldsymbol{I})$ (Kimeldorf and Wahba, 1970; Wahba, 1990; Rasmussen and Williams, 2006). It is given by the function

$$\bar{f}(\boldsymbol{x}_*) = \boldsymbol{k}_*^\intercal (\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N)^{-1} \boldsymbol{y}, \qquad (2)$$

where $\boldsymbol{K}$ is the kernel Gram matrix, the matrix in $\mathbb{R}^{N \times N}$ whose elements are given by $K_{ij} = k(x_i, x_j)$, and $\boldsymbol{k}_*^\intercal \in \mathbb{R}^{1 \times N}$ is the projection vector with element $\boldsymbol{k}_{*,i} = k(x_*, x_i)$. Throughout, we will use the shorthand notation $\boldsymbol{B} := \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$. The marginal variance of the Gaussian process is given by

$$\mathbb{V}(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\intercal (\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N)^{-1} \boldsymbol{k}_*. \qquad (3)$$

For the purposes of this paper we will take this situation as given and ignore the issue of how the kernel k should be chosen and adapted to the dataset (see Schölkopf and Smola, 2002; Rasmussen and Williams, 2006, for discussions). The primary *computational* issue of least-squares estimation is the solution of the linear problem $\boldsymbol{B}^{-1}\boldsymbol{y}$. The standard algorithms for this purpose—Gaussian elimination (more precisely, LU decomposition (Turing, 1948)) and the Cholesky decomposition (Benoît, 1924)—have worst-case complexity $\mathcal{O}(N^3)$.

In the kernel and Gaussian process communities, many approximate inference methods have been proposed to

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

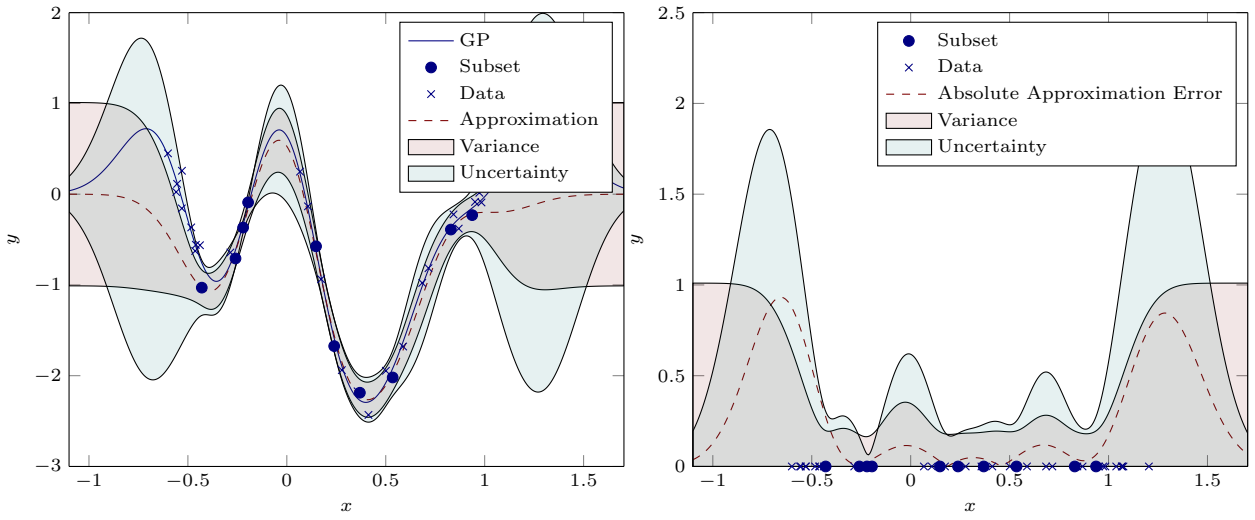

Figure 1: Conceptional sketch: Difference between posterior marginal variance (red shading) of an approximate Gaussian process model over f and the estimate of the numerical error on the approximated posterior mean/least-squares estimate (green shading) introduced in this paper. Crosses denote the full data set; circles denote the subset used in the approximation. **Left:** absolute function values. The solid line is the exact posterior mean/least squares estimate arising from the entire dataset. The dashed line is the Subset of Data approximation. **Right:** plot relative to the exact least-squares/posterior mean function. The dashed line is the absolute value of the approximation error. While the exact posterior mean is not always within one posterior variance of the approximated GP, the numerical error estimate is a hard upper bound on the difference between the posterior means. For clarity, this conceptional sketch uses the implicit exact choice $\boldsymbol{W} = \sqrt{2}\boldsymbol{H}$, which is only *estimated* in the experiments of Section 4.

address this issue by constructing an approximation $\hat{f}$ to $\bar{f}$ at cost linear or sub-linear in N (Zhu et al., 1998; Csató and Opper, 2002; Snelson and Ghahramani, 2007; Walder et al., 2008; Rahimi and Recht, 2009; Titsias, 2009; Lázaro-Gredilla et al., 2010; Yan and Qi, 2010; Wilson et al., 2013; Le et al., 2013; Solin and Särkkä, 2014). Reviews like that of Chalupka et al. (2013) or Quiñonero-Candela and Rasmussen (2005) suggest that several of these methods do give sizeable speedups and reasonably good approximations; there is currently no unique "best" such approximation. Virtually all these approximate inference methods provide a point-estimator for $\bar{f}$ without an uncertainty estimate. Our aim here is to provide such uncertainty measures, i.e. an estimate of the residual $r(x) := |\hat{f}(x) - \bar{f}(x)|$. In this new line of work we provide insights for the Subset of Data approach. Other approximation techniques are under investigation.

Building on recent results (Hennig, 2015) that cast elementary linear algebra algorithms as instances of Gaussian regression, Section 2 shows how the LU and Cholesky decompositions of symmetric positive definite (s.p.d.) matrices can be interpreted as two different formulations of the same posterior mean of a family of Gaussian distributions arising from conditioning a Gaussian prior over the elements of the matrix on 'observed' linear projections of said matrix along conjugate directions (this is related to an argument already made by Hestenes and Stiefel (1952)). We also show that within this family, there exists a possible choice of prior covariance that provides a calibrated uncertainty estimate if the algorithm is stopped after less than N steps (after the full N steps, the algorithm converges, and the posterior covariance vanishes). Interestingly, it is not trivial to make this posterior covariance explicit, as it is only used implicitly in the classic algorithm. Doing so requires the estimation of a large number of unobserved parameters, for which we provide a simple, regularized solution of low computational cost. The result is a statistical estimate of a strict upper bound of the residual $r(x)$. If the algorithm is run for M steps, it has cost $\mathcal{O}(MN + M^3)$. This is more expensive than the cheapest approximate inference methods for least-squares (which are sub-linear in N), but cheaper than many state of the art Gaussian process approximation methods that are $\mathcal{O}(M^2N)$.

While we do not study application examples, such an uncertainty estimate has obvious use cases. For example, this numerical uncertainty could be propagated forward as an additional term to control exploration in reinforcement learning with Gaussian processes (e.g. Boedecker et al., 2014). Or, it could be used to control

computational effort invested into maximum likelihood optimization of kernel hyper-parameters. One could identify irrelevant parameter configurations by training on a subset of the data and evaluating the predictive performance taking into account the residual.

A practical advantage of the fact that our algorithm is a re-interpretation of classic matrix decompositions of s.p.d. matrices is that it can be implemented with ease and great efficiency: Since the Cholesky and LU decompositions are an elementary operation, they are available (and optimized well) in all major linear algebra libraries. Leveraging the output of such a basic library, our algorithm can be formulated in a few lines of additional code. An instance of such code in matlab can be found at the end of this text, in Section 5.

2 GAUSSIAN INFERENCE FOR LEAST-SQUARES

Before we move to the main derivations, we establish some relevant background.

2.1 Notation

Our notation largely follows that of Rasmussen and Williams (2006). In the following, $\mathrm{diag}(\boldsymbol{B})$, for a square matrix $\boldsymbol{B}$, is a diagonal matrix of same shape as $\boldsymbol{B}$, containing the diagonal elements of $\boldsymbol{B}$. Throughout the discourse $\boldsymbol{B}$ will be a symmetric and positive definite matrix. We will also use the shorthand $\boldsymbol{H} := \boldsymbol{B}^{-1}$ for the inverse of $\boldsymbol{B}$.

2.2 The Subset of Data (SoD) Approximation

Arguably the most straightforward approximation $\hat{f}$ arises by simply considering only a subset of the provided data, and computing the full least-squares solution on this subset. Chalupka et al. (2013) showed empirically that if the dataset is sufficiently regular, even randomly selected subsets can provide good approximations. This result is intuitive: If the latent function is regular enough to be over-sampled by the dataset, then such randomly selected sub-sets provide good coverage of the function. The SoD approximation naturally has sub-linear complexity $\mathcal{O}(M^3)$, where M the number of samples is much smaller than N, the total number of data points.

2.3 Gaussian Elimination and the LU, Cholesky Decompositions

Gaussian elimination transforms the system $\boldsymbol{B}\boldsymbol{\alpha} = \boldsymbol{y}$ into triangular form, whence $\boldsymbol{\alpha}$ can be obtained by substitution. If $\boldsymbol{B}$ is invertible (in particular, if it is s.p.d.), there exists a lower unitriangular[1] matrix $\boldsymbol{L}$ and an upper triangular matrix $\boldsymbol{U}$, such that $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{U}$. The matrix $\boldsymbol{U}$ is the triangular result of Gaussian elimination, while $\boldsymbol{L}$ contains the applied transformations (Golub and Van Loan, 1996).

For the s.p.d. case, Hestenes and Stiefel (1952) showed that Gaussian elimination can be formulated as a *conjugate directions method*. Two vectors $\boldsymbol{s}_i, \boldsymbol{s}_j \in \mathbb{R}^{N\times 1}$ are called $\boldsymbol{B}$-conjugate if $\boldsymbol{s}_i^\intercal \boldsymbol{B} \boldsymbol{s}_j \propto \delta_{ij}$ (using Kronecker's symbol). Conjugate vectors can be constructed with a slight modification of the Gram-Schmidt orthogonalization procedure: given a linearly independent set $\boldsymbol{u}_1, ..., \boldsymbol{u}_M \in \mathbb{R}^{N\times 1}$, set

$$\boldsymbol{s}_i := \boldsymbol{u}_i - \sum_{j=1}^{i-1} \frac{\boldsymbol{s}_j^\intercal \boldsymbol{B} \boldsymbol{u}_i}{\boldsymbol{s}_j^\intercal \boldsymbol{B} \boldsymbol{s}_j} \boldsymbol{s}_j. \tag{4}$$

When using the standard unit vectors $\boldsymbol{e}_i$ for $\boldsymbol{u}_i$ then

$$\boldsymbol{S} := [\boldsymbol{s}_1, ..., \boldsymbol{s}_M], \tag{5}$$

and $\boldsymbol{Y} := \boldsymbol{B}\boldsymbol{S}$ have the property that $\boldsymbol{Y}^\intercal = \boldsymbol{U}$ and $\boldsymbol{Y}(\boldsymbol{S}^\intercal\boldsymbol{Y})^{-1} = \boldsymbol{L}$ where $\boldsymbol{S}^\intercal\boldsymbol{Y}$ is diagonal. If $\boldsymbol{B}$ is positive definite, then the diagonal elements of $\boldsymbol{S}^\intercal\boldsymbol{Y} = \boldsymbol{S}^\intercal\boldsymbol{B}\boldsymbol{S}$ are all positive. Let $(\boldsymbol{S}^\intercal\boldsymbol{Y})^{1/2}$ be the diagonal matrix containing the element-wise square-roots of the corresponding elements in $\boldsymbol{S}^\intercal\boldsymbol{Y}$. Then $\boldsymbol{R} = (\boldsymbol{S}^\intercal\boldsymbol{Y})^{1/2}\boldsymbol{Y}^\intercal$ is a lower-triangular matrix, and $\boldsymbol{B} = \boldsymbol{R}\boldsymbol{R}^\intercal$. Thus, $\boldsymbol{R}$ is the Cholesky decomposition of $\boldsymbol{B}$ (which is unique, (Golub and Van Loan, 1996)).

2.4 Gaussian Inference over Matrices

Hennig (2015) recently showed that the method of conjugate gradients (Hestenes and Stiefel, 1952) can be interpreted as least-squares/Gaussian inference on matrix elements. We will use a similar but simpler argument regarding the Cholesky and LU decompositions in the next section. As a preliminary, we briefly introduce the notion of parametric Gaussian inference on the elements of a matrix, using the following notation (van Loan, 2000): For $\boldsymbol{B} \in \mathbb{R}^{N\times N}$, we will denote with $\overrightarrow{\boldsymbol{B}} \in \mathbb{R}^{N^2\times 1}$ a vector created by stacking the rows of $\boldsymbol{B}$. The Kronecker product of two matrices $\boldsymbol{A} \in \mathbb{R}^{M_A\times N}$ and $\boldsymbol{C} \in \mathbb{R}^{M_C\times N}$ is the $M_A M_C \times N^2$ matrix with $[\boldsymbol{A}\otimes\boldsymbol{C}]_{(i,j),(k,l)} = \boldsymbol{A}_{ik}\boldsymbol{C}_{jl}$ where (i,j) is a double index. It has the property

$$(\boldsymbol{A}\otimes\boldsymbol{C})\overrightarrow{\boldsymbol{B}} = \overrightarrow{\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}^\intercal}. \tag{6}$$

Using this notation, one can define a multivariate Gaussian[2] prior over the matrix $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ (the inverse of

[1] i.e. it is triangular, with only ones on the diagonal.

[2] Such distributions are also known as "matrix-variate" Gaussians (Dawid, 1981), but this name does not generalize to the symmetric version used below.

$\boldsymbol{B}$), using a s.p.d. matrix $\boldsymbol{W} \in \mathbb{R}^{N\times N}$,

$$\mathcal{N}(\boldsymbol{H}; \overrightarrow{\boldsymbol{H}_0}, \boldsymbol{W} \otimes \boldsymbol{W}). \tag{7}$$

Assume some algorithm chooses a matrix of 'directions' $\boldsymbol{S} \in \mathbb{R}^{N\times M}$ of rank M, and then computes $\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{S}$. Clearly, $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{Y}$ is a linear transformation of $\boldsymbol{H}$, and $\overrightarrow{\boldsymbol{S}} = (\boldsymbol{I} \otimes \boldsymbol{Y}^\intercal)\overrightarrow{\boldsymbol{H}}$. Since Gaussians are closed under linear conditioning, the posterior arising from these linear observations is also Gaussian. It has the form (Hennig and Kiefel, 2012)

$$\begin{aligned} p(\boldsymbol{H}|\boldsymbol{S},\boldsymbol{Y}) = \mathcal{N}(\boldsymbol{H}; \overrightarrow{\boldsymbol{H}_0 + \boldsymbol{H}_M}, \boldsymbol{W} \otimes \boldsymbol{W}_M) \\ \text{where} \quad \boldsymbol{H}_M = \boldsymbol{W}\boldsymbol{Y}\boldsymbol{G}^{-1}\boldsymbol{\Delta}^\intercal, \text{ and} \\ \boldsymbol{W}_M = \left[\boldsymbol{W} - \boldsymbol{W}\boldsymbol{Y}\boldsymbol{G}^{-1}\boldsymbol{Y}^\intercal\boldsymbol{W}\right], \end{aligned} \tag{8}$$

using the shorthands $\boldsymbol{\Delta} := \boldsymbol{S} - \boldsymbol{H}_0\boldsymbol{Y}$ and $\boldsymbol{G} := \boldsymbol{Y}^\intercal\boldsymbol{W}\boldsymbol{Y}$. Note that the posterior mean can be computed in $\mathcal{O}(N^2M)$.

Encoding Symmetry It is possible to restrict probability mass to only symmetric matrices, using the symmetric Kronecker product for which we define the matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{N^2\times N^2}$ with $[\boldsymbol{\Gamma}]_{(i,j),(k,l)} = 0.5(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{kj})$. This operator symmetrizes matrices: $2\boldsymbol{\Gamma}\overrightarrow{\boldsymbol{B}} = \overrightarrow{\boldsymbol{B} + \boldsymbol{B}^\intercal}$. Then, for square matrices $\boldsymbol{A}, \boldsymbol{C} \in \mathbb{R}^{N\times N}$ the symmetric Kronecker product is $\boldsymbol{A}\otimes\boldsymbol{C} := \boldsymbol{\Gamma}(\boldsymbol{A} \otimes \boldsymbol{C})\boldsymbol{\Gamma}$ and satisfies

$$4(\boldsymbol{A}\otimes\boldsymbol{C})\overrightarrow{\boldsymbol{B}} = \overrightarrow{\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}^\intercal + \boldsymbol{A}\boldsymbol{B}^\intercal\boldsymbol{C}^\intercal + \boldsymbol{C}\boldsymbol{B}\boldsymbol{A}^\intercal + \boldsymbol{C}\boldsymbol{B}^\intercal\boldsymbol{A}^\intercal}. \tag{9}$$

A prior of the form

$$\mathcal{N}(\boldsymbol{H}; \overrightarrow{\boldsymbol{H}_0}, \boldsymbol{W}\otimes\boldsymbol{W}) \tag{10}$$

yields the posterior (Hennig, 2015)

$$\begin{aligned} \mathcal{N}(\overrightarrow{\boldsymbol{H}}; \overrightarrow{\boldsymbol{H}_0 + \boldsymbol{H}_M + \boldsymbol{H}_M^\intercal - \boldsymbol{W}\boldsymbol{Y}\boldsymbol{G}^{-1}\boldsymbol{Y}^\intercal\boldsymbol{H}_M}, \\ \boldsymbol{W}_M\otimes\boldsymbol{W}_M). \end{aligned} \tag{11}$$

Here, too, the mean can be computed in $\mathcal{O}(N^2M)$. Note the difference in the posterior covariance: If symmetry is not encoded, the posterior covariance is $\boldsymbol{W} \otimes \boldsymbol{W}_M$; whereas, encoding symmetry, the posterior covariance is $\boldsymbol{W}_M\otimes\boldsymbol{W}_M$. Loosely speaking, under a $\otimes$ prior the uncertainty 'decreases only column-wise'.

2.5 LU and Cholesky decompositions as a Posterior Mean

We note that, from the general forms of posteriors above, there is a *family* of Gaussian priors, indexed by a positive number $\gamma \in \mathbb{R}_+$, whose posterior means, given conjugate directions $\boldsymbol{S}$, equal the LU and Cholesky decompositions: If we set $\boldsymbol{B}_0 := \boldsymbol{0}$ and[3] $\boldsymbol{W} := \gamma\boldsymbol{B}$ in Equation 7 , then Equation 8 gives

$$\boldsymbol{B}_M = \boldsymbol{Y}(\boldsymbol{S}^\intercal\boldsymbol{Y})^{-1}\boldsymbol{Y}^\intercal = \boldsymbol{L}\boldsymbol{U} = \boldsymbol{R}\boldsymbol{R}^\intercal. \tag{12}$$

Here, the roles of $\boldsymbol{S}$ and $\boldsymbol{Y}$ are exchanged relative to Equation 7, since we now perform inference over $\boldsymbol{B}$. Then by choice of $\boldsymbol{W}$, $\boldsymbol{W}\boldsymbol{S} = \boldsymbol{Y}$ and also $\boldsymbol{\Delta} = (\boldsymbol{Y} - \boldsymbol{B}_0\boldsymbol{S}) = \boldsymbol{Y}$.

For inference over $\boldsymbol{H}$, recall that $\boldsymbol{S}$ can be seen as observations of linear transformations of $\boldsymbol{H}$ with $\boldsymbol{Y}$. Thus the same data can be used to obtain a decomposition for $\boldsymbol{B}$ and $\boldsymbol{H}$. Analogously to the inference over $\boldsymbol{B}$ we set $\boldsymbol{H}_0 := \boldsymbol{0}$ and $\boldsymbol{W} := \gamma\boldsymbol{H}$. It may seem counter-intuitive to use the very matrix that is to be inferred in the computations. In this argument, however, this is an implicit choice: for the posterior mean it is not actually required; instead, every term $\boldsymbol{W}\boldsymbol{Y}$ is simply replaced by $\boldsymbol{S}$. Below, we will first show (Section 3.1) that there is a choice of γ in the family that can be interpreted as a meaningful error estimate. Section 3.2 then shows how to *estimate* this $\boldsymbol{W}$ empirically. The posterior mean is

$$\boldsymbol{H}_M = \boldsymbol{S}(\boldsymbol{Y}^\intercal\boldsymbol{S})^{-1}\boldsymbol{S}^\intercal \tag{13}$$

A difference to the posterior over $\boldsymbol{B}$ is that the decomposition is $\boldsymbol{U}\boldsymbol{L}$, and that the intermediate estimates for $\boldsymbol{H}$ are only non-zero in the upper left $M\times M$ block (see Section 2.6 below). For better uncertainty estimates we also include the knowledge about the symmetry of the matrix. For the specific choice $\boldsymbol{H}_0 = \boldsymbol{0}$, this has no effect on the posterior mean as many terms cancel, but the posterior uncertainty is less conservative.

2.6 Relation to Subset of Data

The predictive mean $\boldsymbol{H}_M$ of Eq. (13) is that of the subset of data approximation. To see this, first note that, as pointed out above, only the upper $M\times M$ block is non-zero and we are going to show that this upper block is exactly $(\boldsymbol{K}_{\boldsymbol{U},\boldsymbol{U}} + \sigma^2\boldsymbol{I})^{-1}$, where $\boldsymbol{U}$ denotes the indices of the selected subset, i.e.

$$\boldsymbol{K} + \sigma^2\boldsymbol{I} = \begin{pmatrix} \boldsymbol{B}_{\boldsymbol{U},\boldsymbol{U}} & \boldsymbol{B}_{\boldsymbol{U},\boldsymbol{X}\setminus\boldsymbol{U}} \\ \boldsymbol{B}_{\boldsymbol{X}\setminus\boldsymbol{U},\boldsymbol{U}} & \boldsymbol{B}_{\boldsymbol{X}\setminus\boldsymbol{U},\boldsymbol{X}\setminus\boldsymbol{U}} \end{pmatrix}. \tag{14}$$

By construction, $\boldsymbol{S}$ is upper triangular and we denote this upper part with $\boldsymbol{S}_U$, i.e $\boldsymbol{S} = \begin{pmatrix}\boldsymbol{S}_U^\intercal & \boldsymbol{0}\end{pmatrix}^\intercal$. Multiply-

[3] This choice is only well-defined for s.p.d. $\boldsymbol{B}$ since, as a covariance, $\boldsymbol{W}$ must be s.p.d. itself.

ing $\boldsymbol{H}_M = \boldsymbol{S}(\boldsymbol{S}^\intercal \boldsymbol{Y})^{-1}\boldsymbol{S}^\intercal$ with $\boldsymbol{B}$ yields

$$\boldsymbol{S}\left[(\boldsymbol{S}_U^\intercal \quad \boldsymbol{0})\begin{pmatrix}\boldsymbol{B}_{U,U} & \boldsymbol{B}_{U,\boldsymbol{X}\setminus U}\\ \boldsymbol{B}_{\boldsymbol{X}\setminus U,U} & \boldsymbol{B}_{\boldsymbol{X}\setminus U,\boldsymbol{X}\setminus U}\end{pmatrix}\begin{pmatrix}\boldsymbol{S}_U\\ \boldsymbol{0}\end{pmatrix}\right]^{-1}\boldsymbol{S}^\intercal\boldsymbol{B} \tag{15}$$

$$= \begin{pmatrix}\boldsymbol{S}_U\\ \boldsymbol{0}\end{pmatrix}\left[\boldsymbol{S}_U^\intercal \boldsymbol{B}_{U,\boldsymbol{X}\setminus U}\boldsymbol{S}_U\right]^{-1}(\boldsymbol{S}_U^\intercal \quad \boldsymbol{0})\,\boldsymbol{B} \tag{16}$$

$$= \begin{pmatrix}\boldsymbol{S}_U\\ \boldsymbol{0}\end{pmatrix}\boldsymbol{S}_U^{-1}\boldsymbol{B}^{-1}_{U,\boldsymbol{X}\setminus U}(\boldsymbol{S}_U^{-1})^\intercal(\boldsymbol{S}_U^\intercal \quad \boldsymbol{0})\,\boldsymbol{B} \tag{17}$$

$$= \begin{pmatrix}\boldsymbol{B}^{-1}_{U,U} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}\boldsymbol{B} \tag{18}$$

$$= \begin{pmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix} \tag{19}$$

We note in passing that the approximation of Titsias (2009) can be formulated in an analogous fashion, by choosing $\boldsymbol{B}_0 = \sigma^2\boldsymbol{I}$, $\boldsymbol{W} = \boldsymbol{K}$ and $\boldsymbol{S} := \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{U}}$. But it is less clear how the corresponding posterior variance can be fashioned into a meaningful error estimate.

3 EMPIRICAL FITTING OF THE POSTERIOR VARIANCE

The preceding section showed that the "intermediate" LU and Cholesky decompositions (those of $\boldsymbol{H}_{\boldsymbol{UU}}$) can be interpreted as the posterior mean of Gaussian distributions over $\boldsymbol{H}$ arising from conditioning a family of Gaussian priors on conjugate projections $\boldsymbol{S}$ of $\boldsymbol{B}$. As such, this observation does not yet imply that the associated posterior variances of the Gaussian family can be interpreted as a measure of uncertainty. In this section, we show that the posterior variance of the belief over $\boldsymbol{H}$ provides a strict upper bound to the approximation error of the Subset of Data approximation, *if* one could set $\boldsymbol{W} = \sqrt{2}\boldsymbol{H}$. Of course, empirically, this is not possible. Thus, a subsequent section will show how to practically estimate a useful $\boldsymbol{W}$ that approximates this bound at low cost.

3.1 An Upper Bound on the Approximation Error

Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^N$ be arbitrary vectors. The Gaussian measure (11) on $\boldsymbol{H}$ implies that the scalar $\hat{\mu} := \boldsymbol{a}^\intercal\boldsymbol{H}\boldsymbol{b}$ is Gaussian distributed as well, with mean $\boldsymbol{a}^\intercal\boldsymbol{H}_M\boldsymbol{b}$ and a variance we denote with $\hat{\epsilon}^2 := \frac{1}{2}\left(\boldsymbol{a}^\intercal\boldsymbol{W}_M\boldsymbol{a}\boldsymbol{b}^\intercal\boldsymbol{W}_M\boldsymbol{b} + (\boldsymbol{a}^\intercal\boldsymbol{W}_M\boldsymbol{b})^2\right)$ (derived in the proof).

Theorem. *The absolute error $|\hat{\mu} - \boldsymbol{a}^\intercal\boldsymbol{H}\boldsymbol{b}|$ divided by the standard deviation $\hat{\epsilon}$ is always less than* 1*:*

$$\frac{|\boldsymbol{a}^\intercal\boldsymbol{H}_M\boldsymbol{b} - \boldsymbol{a}^\intercal\boldsymbol{H}\boldsymbol{b}|}{\hat{\epsilon}} < 1 \tag{20}$$

For $\boldsymbol{a} = \boldsymbol{b}$ the ratio is exactly $\frac{1}{\sqrt{2}}$ (or, for the choice $\boldsymbol{W} = \boldsymbol{H}$, it is exactly one).

Proof. See supplementary. □

Corollary. *The difference between the subset of data approximation and the exact least-squares solution, at a specific test location $\boldsymbol{x}_*$, divided by the standard deviation of the estimate under the Gaussian posterior assigned by Eq.* (11) *with $\boldsymbol{W} = \sqrt{2}\boldsymbol{H}$ is bound above by 1.*

$$\frac{|\boldsymbol{k}_*^\intercal\boldsymbol{H}\boldsymbol{y} - \boldsymbol{k}_*^\intercal\boldsymbol{H}_M\boldsymbol{y}|}{\sqrt{\hat{\epsilon}_*^2}} < 1 \tag{21}$$

Further, under the choice $\boldsymbol{W} = \boldsymbol{H}$, the difference between the exact and approximate correction term in the marginal variance of the Gaussian process on f, divided by the posterior standard deviation of the estimate, is exactly 1*.*

$$\frac{|\boldsymbol{k}_*^\intercal\boldsymbol{H}\boldsymbol{k}_* - \boldsymbol{k}_*^\intercal\boldsymbol{H}_M\boldsymbol{k}_*|}{\sqrt{\hat{\epsilon}_*^2}} = 1 \tag{22}$$

To avoid confusion, it is important to note that the proof of Theorem 3.1 does not assume anything about $\boldsymbol{a}$ or $\boldsymbol{b}$. In particular when setting $\boldsymbol{a} = \boldsymbol{k}_*$ and $\boldsymbol{b} = \boldsymbol{y}$ in the Corollary, there is no assumption that $\boldsymbol{y}$ is actually a draw from the Gaussian process with covariance k. The implication of the Theorem and Corollary is that *there is* a well-calibrated Gaussian belief around the SoD approximation that remains well-calibrated across all possible choices of M. However, the prior covariance necessary for this bound involves the unknown matrix $\boldsymbol{H}$ itself. The Theorem is the matrix-valued (and symmetric Kronecker-covariant) extension of the trivial scalar statement that there is a variance σ^2 such that, given x and μ the Gaussian distribution $\mathcal{N}(x; 0, \sigma^2)$ is well-scaled in the sense that $x^2/\sigma^2 < 1$ (namely $\sigma = |x|$). Below, we will now introduce a practical empirical estimator for $\boldsymbol{W}$.

3.2 Estimating the Posterior Variance

We adopt an empirical Bayesian approach, constructing an approximation to $\boldsymbol{W}$ based on the collected observations $(\boldsymbol{S}, \boldsymbol{Y})$. A good approximation $\hat{\boldsymbol{W}}$ exhibits the characteristics of $\boldsymbol{H}$. Besides symmetry and positive definiteness, $\boldsymbol{H}$ also satisfies $\boldsymbol{H}\boldsymbol{Y} = \boldsymbol{S}$ and $\boldsymbol{Y}^\intercal\boldsymbol{H} = \boldsymbol{S}^\intercal$. The space of all matrices satisfying this condition can be parametrized as

$$\hat{\boldsymbol{W}}(\boldsymbol{\Omega}) := \boldsymbol{H}_M + (\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{Y}^\intercal\boldsymbol{S})^{-1}\boldsymbol{Y}^\intercal)\boldsymbol{\Omega}(\boldsymbol{I} - \boldsymbol{Y}(\boldsymbol{S}^\intercal\boldsymbol{Y})^{-1}\boldsymbol{S}^\intercal) \tag{23}$$

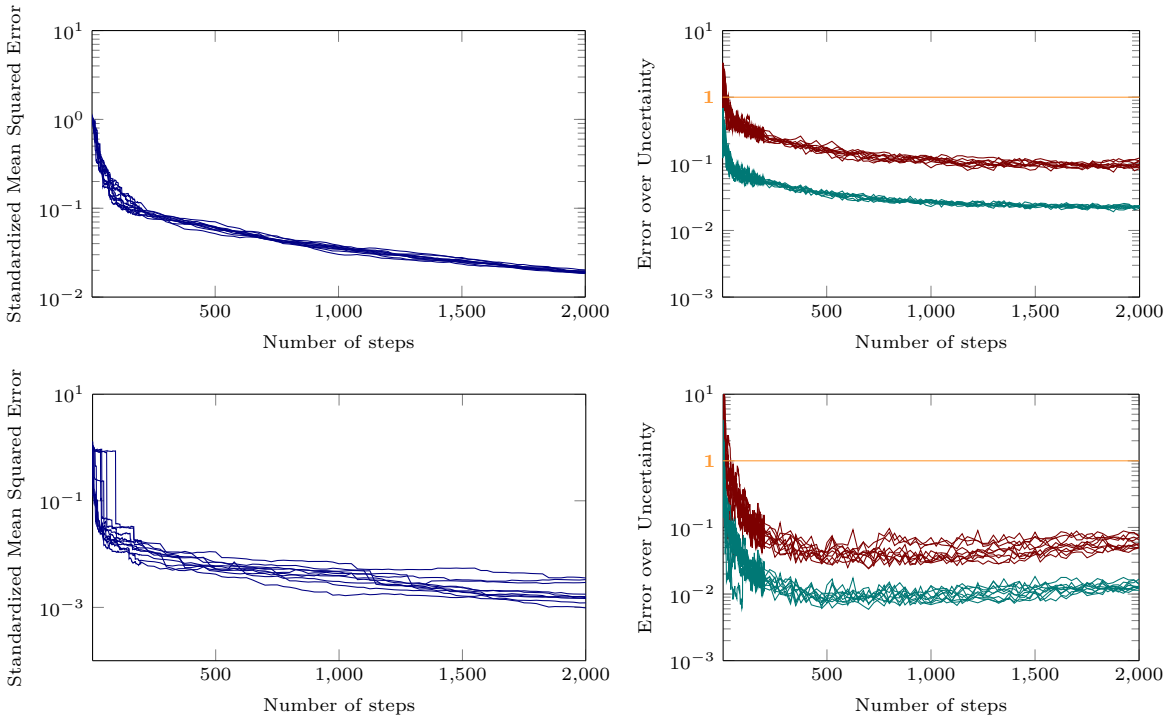

Figure 2: Ten random initializations of the probabilistic subset of data approximation on the PUMADYN (**top row**) and CPU (**bottom row**) data sets, using the ARD Squared Exponential kernel. **Left:** standardized mean squared error for Subset of Data. **Right:** ratio between absolute error and uncertainty. The upper lines are the maximum, the lower lines the average over all test inputs. The horizontal line shows the theoretical bound at 1 that would be guaranteed if $\boldsymbol{W} = \sqrt{2}\boldsymbol{H}$ where estimated exactly.

Furthermore, for all 'future' $\boldsymbol{Y}_{M+1}$, the matrix $\boldsymbol{Y}_{M+1}^\intercal \boldsymbol{H} \boldsymbol{Y}_{M+1}$ is diagonal, because $\boldsymbol{S}$ is $\boldsymbol{B}$-conjugate.

$$\begin{aligned} \boldsymbol{Y}_{M+1}^\intercal \boldsymbol{H} \boldsymbol{Y}_{M+1} &= \boldsymbol{S}_{M+1} \boldsymbol{B}^\intercal \boldsymbol{H} \boldsymbol{B} \boldsymbol{S}_{M+1} && (24) \\ &= \boldsymbol{S}_{M+1}^\intercal \boldsymbol{B} \boldsymbol{S}_{M+1} && (25) \\ &= \operatorname{diag}(\boldsymbol{S}_{M+1}^\intercal \boldsymbol{B} \boldsymbol{S}_{M+1}) && (26) \end{aligned}$$

For the same reason, $\hat{\boldsymbol{W}}(\boldsymbol{\Omega})$ has the property $\boldsymbol{Y}_{M+1}^\intercal \hat{\boldsymbol{W}}(\boldsymbol{\Omega}) \boldsymbol{Y}_{M+1} = \boldsymbol{Y}_{M+1}^\intercal \boldsymbol{\Omega} \boldsymbol{Y}_{M+1}$. This suggests choosing a scalar $\boldsymbol{\Omega} = \omega \boldsymbol{I}$. Assuming the subset is drawn i.i.d. from the full dataset, ω can be estimated by a simple average,

$$\omega := \operatorname{avg}(\operatorname{diag}(\boldsymbol{S}^\intercal \boldsymbol{B} \boldsymbol{S}) \cdot \operatorname{diag}(\boldsymbol{Y}^\intercal \boldsymbol{Y})^{-1}). \quad (27)$$

For clarity, we point out that this choice does not yield a scalar $\hat{\boldsymbol{W}}$. Rather, the projections to the left and right of $\boldsymbol{\Omega}$ in Eq. (24) capture aspects of the structure of $\boldsymbol{H}$ to construct a nontrivial estimate $\hat{\boldsymbol{W}}$. Computing this estimate ω requires $\mathcal{O}(M^2 N)$ operations. However, in practice the last 10 columns of $\boldsymbol{S}$ and $\boldsymbol{Y}$ are enough such that the effort is only $10 \cdot \mathcal{O}(MN)$.

4 EXPERIMENTS

The purpose of this section is to provide insights on the practicability of *the uncertainty estimate* on the Subset of Data approximation for the least-squares solution. For studies on the quality of the Subset of Data approximation in relation to the many other approximation methods see e.g. Titsias (2009) or Chalupka et al. (2013). These reviews find that the SoD approximation is competitive with other approaches if the regression function is sufficiently regular.

Figure 1 shows, on a toy example, the posterior uncertainty given the correct $\boldsymbol{W}$ and contrasts it with the posterior uncertainty of the approximated Gaussian process. To assess the performance of the estimated $\hat{\boldsymbol{W}}$, we chose two medium sized data sets where computing the full Gaussian process posterior is still feasible. For each data set we optimized the kernel parameters by maximum marginal likelihood, until a test error of less than 0.1 was achieved. As test metric, we use the Standardized Mean Squared Error (Rasmussen and

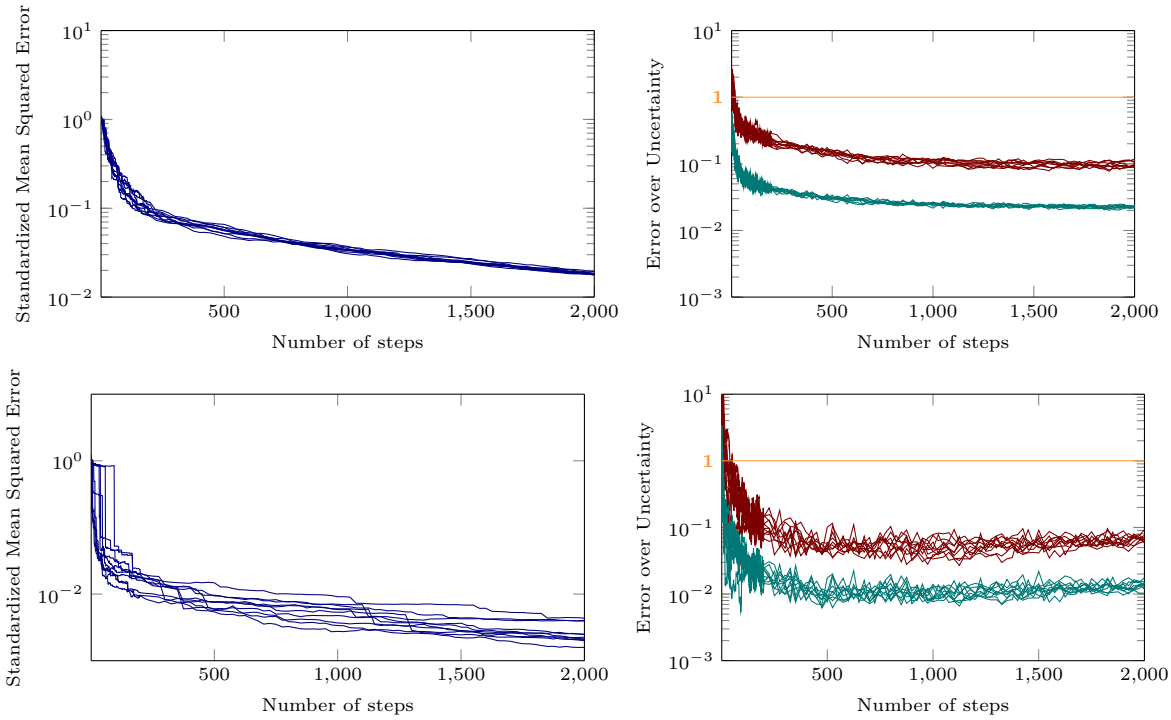

Figure 3: Same setup as Figure 2, but using the ARD Matérn 5/2 kernel.

Williams, 2006, p. 23),

$$\text{SMSE} = \frac{1}{n_*} \sum_{k=1}^{n_*} \frac{(y_{*k} - \mu_{*k})^2}{\text{Var}[\boldsymbol{y}_*]}, \tag{28}$$

where $\boldsymbol{y}_*$ are the test targets and $\boldsymbol{\mu}_*$ is the mean prediction in the test locations. For Subset of Data we replaced the test targets with the corresponding mean predictions of the Gaussian process. We performed these experiments using two different kernel functions, namely automatic relevance determination (ARD) Squared Exponential and ARD Matérn 5/2 (Rasmussen and Williams, 2006, p. 83f, p. 106).

$$k_{\text{SE}}(d(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda})) = f \exp\left(-\frac{1}{2} d^2\right) \tag{29}$$

$$k_{52}(d(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda})) = f \left(1 + \sqrt{5}d + \frac{5}{3}d^2\right) \exp\left(-\sqrt{5}d\right) \tag{30}$$

where $f \in \mathbb{R}^+, \boldsymbol{\lambda} \in \mathbb{R}^D$ are parameters and $d(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda}) := \boldsymbol{x}^\intercal \operatorname{diag}(\boldsymbol{\lambda})^{-1} \boldsymbol{z}$. The reason to do so is that different choices in kernel give linear problems of varying difficulty, and kernel Gram matrices of differing sparseness. We evaluated the estimator selecting the subsets randomly and – as suggested by Chalupka et al. (2013) – using *Farthest Point Clustering* (FPC) (Gonzalez, 1985). We recorded maximum and average of the ratio between approximation error and estimated error across the test inputs. The first dataset is `pumadyn32nm`, available from `http://www.cs.toronto.edu/~delve/data/datasets.html`. The other data set is called `CPU`, available from `http://archive.ics.uci.edu/ml/`. `CPU` contains 6554 data points in a 21 dimensional input space, and `pumadyn32nm` consists of 7168 data points with 32 input dimensions.

For the randomly selected subsets the results for the Squared Exponential kernel are reported in Figure 2, the results for the Matérn kernel in Figure 3. The results of the FPC experiments are part of the supplementary material. The figures show that it takes a subset of about 200 samples for the estimate $\hat{W}$ to converge to a meaningful value, so that the error bound holds. They also show that, while the upper bound is not tight, it is usually within one or at most two orders of magnitude of the actual maximal error, and thus should also be a meaningful metric for the control of computational effort.

5 CODE

Computing the uncertainty over the prediction can be done in a few lines of code given the already computed objects for a Subset of Data prediction in $\boldsymbol{x}_*$. We use `monospace font` to denote objects referenced in the Matlab function below. For simplicity we assume the selected subset $\boldsymbol{U}$ are the first `M` entries of the

$N \times D$ input data matrix $\boldsymbol{X}$. Further let `k` be the kernel function and $\mathtt{R} = \mathrm{chol}(k(\boldsymbol{X_U}, \boldsymbol{X_U}) + \sigma^2 \boldsymbol{I}_M)$. In Matlab this requires an additional transpose as `chol` computes an upper triangular matrix. We do not directly compute $\boldsymbol{S}$ via Gram-Schmidt but can infer it from `R`. Recall that $\boldsymbol{H}_M$ is zero except for the upper $M \times M$ block and that it is the inverse of the subset kernel matrix. Therefore $(\boldsymbol{R}\boldsymbol{R}^\intercal)^{-1} = \boldsymbol{S}_U(\boldsymbol{S}_U^\intercal \boldsymbol{Y}_U)^{-1}\boldsymbol{S}_U^\intercal$ and $\boldsymbol{S}_U = (\boldsymbol{R}^\intercal)^{-1}(\boldsymbol{S}_U^\intercal \boldsymbol{Y}_U)^{-\frac{1}{2}}$. Since $\boldsymbol{S}$ is constructed from Gram-Schmidt using the standard unit vectors, $\boldsymbol{S}$ is unitriangular and therefore $(\boldsymbol{S}_U^\intercal \boldsymbol{Y}_U)^{-\frac{1}{2}}$ must be $\mathrm{diag}(\boldsymbol{R})$.

In addition to `R` we require `Kus` $:= k(\boldsymbol{U}, \boldsymbol{x}_*)$ and `alpha` $:= (\boldsymbol{R}\boldsymbol{R}^\intercal)^{-1}\boldsymbol{y_U}$. These objects are already available after computing mean and variance for $\boldsymbol{x}_*$. Furthermore we require `Ksx` $:= k(\boldsymbol{x}_*, \boldsymbol{X})$ and `Kxu` $:= k(\boldsymbol{X}, \boldsymbol{U})$. Then $\hat{\epsilon}_*$ from section 3.1 is `get_uncertainty(M, R, alpha, Kus, Ksx, Kxu, y)`.

```
function std = get_uncertainty(M, R, ...
    alpha, Kus, Ksx, Kxu, y)
  % compute last 10 S_U
  S     = zeros(M, 10);
  dR    = diag(R(M-9:M, M-9:M));
  S(M-9:M, :) = diag(dR);
  S     = R' \ S;
  Y     = Kxu * S;
  omega = mean(dR.^2 ./ sum(Y'.*Y', 2));

  a = Kxu * alpha;
  b = Kxu * (R \ (R' \ Kus));
  aWb = Ksx*y - b'*y - Ksx*a + b'*a;
  aWa = y'*y - 2 * (a'*y) + a'*a;
  bWb = Ksx*Ksx' - 2 * (Ksx*b) + b'*b;
  std = aWb^2 + aWa * bWb;
  std = omega / 2 * sqrt(std);
end
```

6 CONCLUSIONS

Computational approximations, like statistical estimators constructed from physical data sources, should come with uncertainty estimates. We introduced a construction of such an uncertainty estimate for the linear optimization problem at the heart of least-squares estimation, one of the most fundamental computations of statistics. Our uncertainty estimate is not based on assumptions about the data labels $\boldsymbol{y}$, in particular it does not require the assumption that the data be sampled from a Gaussian process. The algorithm has linear complexity in the size of the data set. Because it builds a light-weight statistical estimate from the output of a classic numerical method (the Cholesky decomposition), it can be implemented efficiently by leveraging existing, highly optimized implementations of this decomposition.

This kind of error estimate should, in principle, be feasible for most approximate inference methods. Adding this kind of functionality to other approximation methods for least-squares, as well as finding better (i.e. cheaper and more precise) statistical estimators for the posterior variance, is left for future work.

Acknowledgements

The authors would like to thank Maren Mahsereci for helpful discussions that improved an early version of this work.

References

Benoît, E. (1924). Note sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindres carrés à un système d'équations linéaires en nombre inférieur à celui des inconnues. application de la méthode à la résolution d'un système défini d'équations linéaires. (procédé du commandant cholesky). *Bulletin géodésique*, 2(1):67–77.

Boedecker, J., Springenberg, J. T., Wülfing, J., and Riedmiller, M. (2014). Approximate real-time optimal control based on sparse Gaussian process models. In *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*.

Chalupka, K., Williams, C. K. I., and Murray, I. (2013). A framework for evaluating approximation methods for Gaussian process regression. *Journal of Machine Learning Research*, 14(1):333–350.

Csató, L. and Opper, M. (2002). Sparse on-line Gaussian processes. *Neural Comput*, 14(3):641–668.

Dawid, A. P. (1981). Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274.

Golub, G. and Van Loan, C. (1996). *Matrix computations*. Johns Hopkins Univ Pr.

Gonzalez, T. F. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(0):293–306.

Hennig, P. (2015). Probabilistic interpretation of linear solvers. *SIAM Journal on Optimization*, 25(1):234–260.

Hennig, P. and Kiefel, M. (2012). Quasi-Newton methods – a new direction. In *International Conference on Machine Learning (ICML)*.

Hestenes, M. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436.

Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Kimeldorf, G. and Wahba, G. (1970). A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, pages 495–502.

Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881.

Le, Q., Sarlos, T., and Smola, A. (2013). Fastfood - computing Hilbert space expansions in loglinear time. In *International Conference on Machine Learning (ICML)*, volume 28, pages 244–252.

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in applied probability*, pages 439–468.

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

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1313–1320.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, 2 edition.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press.

Snelson, E. and Ghahramani, Z. (2007). Local and global sparse Gaussian process approximations. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 2, pages 524–531.

Solin, A. and Särkkä, S. (2014). Hilbert space methods for reduced-rank Gaussian process regression. *ArXiv e-prints*.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5, pages 567–574.

Turing, A. (1948). Rounding-off errors in matrix processes. *Quarterly Journal of Mechanics and Applied Mathematics*, 1(1):287–308.

van Loan, C. (2000). The ubiquitous Kronecker product. *J of Computational and Applied Mathematics*, 123:85–100.

Wahba, G. (1990). *Spline models for observational data*, volume 59. SIAM.

Walder, C., Kim, K. I., and Schölkopf, B. (2008). Sparse multiscale Gaussian process regression. In *International Conference on Machine Learning (ICML)*, pages 1112–1119.

Wilson, A. G., Gilboa, E., Nehorai, A., and Cunningham, J. P. (2013). Gpatt: Fast multidimensional pattern extrapolation with Gaussian processes.

Yan, F. and Qi, Y. (2010). Sparse Gaussian process regression via l1 penalization. In *International Conference on Machine Learning (ICML)*, pages 1183–1190.

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