Received: 24 January 2023 | Revised: 14 June 2023 | Accepted: 5 July 2023

DOI: 10.1002/sam.11635

WILEY

RESEARCH ARTICLE

Traditional kriging versus modern Gaussian processes for large-scale mining data

Ryan B. Christianson[1] | **Ryan M. Pollyea**[2] | **Robert B. Gramacy**[3]

[1]Department of Statistics & Data Science, NORC at the University of Chicago, Chicago, Illinois, USA

[2]Department of Geosciences, Virginia Tech, Blacksburg, Virginia, USA

[3]Department of Statistics, Virginia Tech, Blacksburg, Virginia, USA

Correspondence
Ryan B. Christianson, Department of Statistics & Data Science, NORC at the University of Chicago, 55 E Monroe St, 30th Floor, Chicago, IL 60603, USA.
Email: christianson-ryan@norc.org

Funding information
National Science Foundation, Grant/Award Numbers: 1822108, 1822146; US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics; Scientific Discovery through Advanced Computing (SciDAC) Program, Grant/Award Number: 0000231018

Abstract

The canonical technique for nonlinear modeling of spatial/point-referenced data is known as kriging in geostatistics, and as Gaussian Process (GP) regression for surrogate modeling and statistical learning. This article reviews many similarities shared between kriging and GPs, but also highlights some important differences. One is that GPs impose a process that can be used to automate kernel/variogram inference, thus removing the human from the loop. The GP framework also suggests a probabilistically valid means of scaling to handle a large corpus of training data, that is, an alternative to ordinary kriging. Finally, recent GP implementations are tailored to make the most of modern computing architectures, such as multi-core workstations and multi-node supercomputers. We argue that such distinctions are important even in classically geostatistical settings. To back that up, we present out-of-sample validation exercises using two, real, large-scale borehole data sets acquired in the mining of gold and other minerals. We compare classic kriging with several variations of modern GPs and conclude that the latter is more economical (fewer human and compute resources), more accurate and offers better uncertainty quantification. We go on to show how the fully generative modeling apparatus provided by GPs can gracefully accommodate left-censoring of small measurements, as commonly occurs in mining data and other borehole assays.

KEYWORDS

Gaussian process regression, multiple imputation, ordinary kriging, surrogate modeling, variogram, Vecchia approximation

1 | INTRODUCTION

The modern literature on spatial nonparametric regression (e.g., "kriging") traces its origins to the mining analytics of Danie Krige and Henri de Wijs and the subsequent work of Matheron [1]. Similar ideas were developed independently around the same time to aid the early analysis of computer simulation experiments, like those conducted in the study of nuclear weapons and energy, however (unclassified) publications did not appear until later (e.g., Sacks et al. [2]). The spatial statistics community was responsible for much of the subsequent advances in methodology (e.g., Cressie [3]), and software for kriging in use commercially (e.g., `LeapFrog`, `Vulcan`, `Surfer`, etc.) and academically (e.g., `GSLIB`) in mining today.

More recently, researchers in geospatial statistics, surrogate modeling of computer experiments, and more broadly in statistical and machine learning communities,

have pushed the boundaries of fidelity and computational tractability as modeling ambition and scale of data collection continue to grow, for example, Gramacy [4]. These disparate literatures have converged around the nomenclature of Gaussian process (GP) regression as a generative framework for the kinds of data and procedures involved in kriging, but with a more cohesive and flexible approach to inference, approximation and automation based upon the likelihood, which is the foundation to modern statistical learning.

Common geoscience applications for spatial smoothing and interpolation include ore-grade estimation and reservoir characterization/simulation; however, software tools utilized for such applications lag the state-of-the-art (as outlined above) by a decade or more. For example, obtaining fits requires expert human interaction with the software library and intuition to entertain alternatives of spanning anisotropies, neighborhood sizes for ordinary kriging in the face of large training data sets, and appropriate semivariogram forms modeling the decay of spatial correlation. Recent advances from the statistics and data analytics communities automate many of these time-consuming tasks, while offering substantial improvements in computational efficiency including the use of contemporary computing architectures such as multi-core workstations and clusters.

The main goal of this paper is to advocate for the more modern, GP perspective via open-source libraries such as for `R: GPvecchia` [5] and `laGP` [6]. These are just two of many examples offering a modern take on ordinary kriging, embodying advances in engineering and statistical learning: likelihood-based criteria offloaded to robust optimization libraries; human out-of-the-loop inference. Not only are they easy to use, but they are also hard to misuse. To emphasize the modern GP perspective, we also provide a review which focuses on the similarties and differences between GPs and kriging.

This narrative is supported by empirical comparison. We consider two real borehole-based mining examples with data records on gold and other minerals, over spatial and depth coordinates, sized in the hundreds of thousands. Later, figure 4 in Section 4 shows a 2d projection of a subset of these data, where they are described in more detail. These data exhibit many typical yet challenging features such as abrupt changes in dynamics, left-censoring of small values due to the sensitivity of the measurement instrument, and large measurement gaps in space. Using these data, we devise a cross-validation-based out-of-sample exercise which is careful to respect the borehole nature of data collection. The outcome of that exercise is evidence that modern GP-based methods are both more accurate, more hands-off, more economical (in terms of computing resources), and offer better uncertainty quantification than their kriging-based analogues. They also enable extensions which would be difficult to entertain without a fully probabilistic generative framework. As a showcase, we entertain a multiple imputation scheme to handle left censoring that involves only a few lines of code around library-based GP fitting and prediction subroutines.

The rest of the paper is outlined as follows. Section 2 contains a review of GP regression and kriging. Building on those, Section 3 contains the main, large-scale GP regression and kriging methods we compare using two real borehole ore data sets. Section 4 has cross validation results comparing those methods on both time and accuracy, including extensions to faciliate (without discarding) a large degree of left-censoring in one of the two data sets. Finally, Section 5 concludes with a discussion and ideas for future work.

2 | GAUSSIAN PROCESS VERSUS KRIGING

We begin by introducing Gaussian process (GP) regression with an eye toward connecting to kriging. At some level, they are the same thing. The biggest differences lie in vocabulary and inference for unknown quantities, which is coupled with the degree of automation/human intervention.

2.1 | Gaussian process regression

Suppose we wish to model a function $f : \mathbb{R}^d \to \mathbb{R}$ with a limited number of noisy evaluations $y_i = f(x_i) + \varepsilon_i$, for $i = 1, \ldots, N$. Let X_N be an $N \times d$ matrix formed with d-dimensional x_i^{T} in each of its rows. Similarly combine scalar outputs y_i into an N-vector Y_N. Throughout this paper, we privilege an input–output (x, y) notational scheme in keeping with the vast majority of the statistical learning literature on regression, nonparametric and nonlinear or otherwise. In many geospatial contexts (e.g., Banerjee [7]), where f might be an environmental or geological process, it is common to use s_i for (spatial) input sites and $z_i = Z(s_i)$, among many alternatives, for the response. We think this unproductively biases thinking toward $d = 2$-dimensional point-referenced (latitude and longitude) data, whereas these methods can be applied much more widely than that. Machine learning (e.g., Rasmussen & Williams [8]) and computer surrogate modeling for example, Gramacy [4] applications are typically in higher input dimension, and one of our goals in this paper is to introduce this way of thinking into the mining literature.

A common nonparametric model for such data is a Gaussian process (GP), which assumes that outputs Y_N follow a multivariate normal (MVN) distribution. Inputs X_N are primarily involved in the specification of the MVN covariance $\Sigma_N \equiv \Sigma(X_N, X_N)$ with a form for $\Sigma(\cdot,\cdot)$ that inverts Euclidean distances between its arguments. For example,

$$Y_N \sim \mathcal{N}_N(0, \Sigma_N), \quad \text{where} \quad \Sigma_N^{ij} \quad \text{follows} \quad S\left(\frac{1}{\text{Dist}(x_i, x_j)}\right) \quad \text{for some decreasing } S. \tag{1}$$

In Equation (1) we are being deliberately imprecise about the form of Σ_N, a topic we shall detail shortly in Section 2.2. For now, simply suppose correlation in outputs decays as a function of distance in inputs: $\mathbb{C}\text{orr}(y_i, y_j) < \mathbb{C}\text{orr}(y_i, y_k)$ if x_i is "closer" to x_k than it is to x_j. We are also using a zero mean specification, so that all of the modeling "action" is in the covariance. Extensions abound.

Although a Bayesian interpretation is not essential in characterizing GP regression, Equation (1) can be said to specify a prior over (noisy evaluations of) functions like f, abstracting as $Y_N \sim \text{GP}$. Choices for the mean (0) and variance (Σ) determine the modeling properties of f like its smoothness and wiggliness. We shall largely leave those properties to our references, except as relevant to particular choices for $\Sigma(\cdot,\cdot)$, again in Section 2.2. Then, if N' new locations $\mathcal{X}$ come along where we do not yet have observations, $Y(\mathcal{X})$, we can summarize our understanding for those in light of the (training) data we do have—a predictive distribution—through the lens of posterior conditioning: $Y(\mathcal{X}) \mid Y_N$. First, extend the GP prior to cover $Y(\mathcal{X})$ jointly with Y_N:

$$\begin{bmatrix} Y_N \\ Y(\mathcal{X}) \end{bmatrix} \sim \mathcal{N}_{N+N'}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_N & \Sigma(X_N, \mathcal{X}) \\ \Sigma(\mathcal{X}, X_N) & \Sigma(\mathcal{X}, \mathcal{X}) \end{bmatrix}\right).$$

In so doing, we may leverage that $Y(\mathcal{X})$ values are more highly correlated with Y_N values whose X_N entries are close to $\mathcal{X}$ by applying standard MVN conditioning rules, such as found in Kalpić & Hlupić [9], $Y(\mathcal{X}) \mid Y_N \sim \mathcal{N}_{N'}(\mu_N(\mathcal{X}), \Sigma_N(\mathcal{X}))$ where

$$\begin{aligned} \mu_N(\mathcal{X}) &= \Sigma(\mathcal{X}, X_N)\Sigma_N^{-1} Y_N \\ \Sigma_N(\mathcal{X}) &= \Sigma(\mathcal{X}, \mathcal{X}) - \Sigma(\mathcal{X}, X_N)\Sigma_N^{-1}\Sigma(X_N, \mathcal{X}). \end{aligned} \tag{2}$$

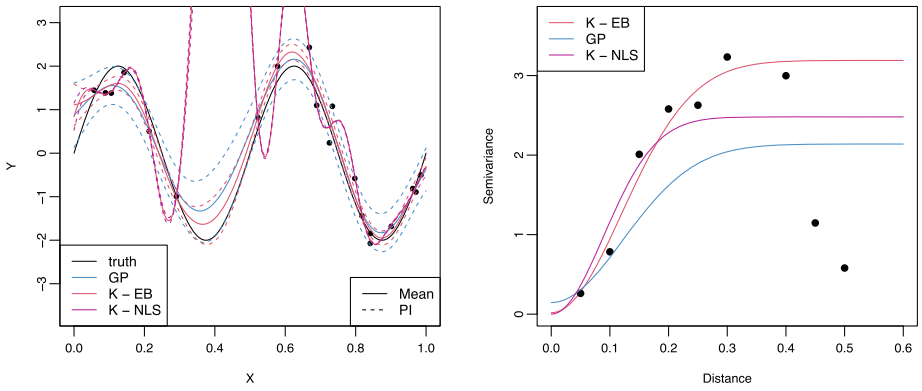
Note that $\Sigma_N(\mathcal{X})$ and $\Sigma(\mathcal{X}, \mathcal{X})$ are $N' \times N'$ matrices; the N subscript serves as a reminder of conditioning on Y_N. Observe that $\mu_N(\mathcal{X})$ is a (high dimensional) linear projection of those Y_N values, where the "weights" involved are inversely proportional to the distance between their X_N values and those of $\mathcal{X}$. Such conditioning identities apply for any MVN, based on a GP prior or otherwise. The special thing about the regression context is the (inverse) distance-based dynamics manifest as $\mathcal{O}(N)$ weights in each row of $\Sigma(\mathcal{X}, X_N)$, and $\mathcal{O}\left(N^2\right)$ in Σ_N, involved in the projection, rather than the usual $\mathcal{O}(d)$ or $\mathcal{O}\left(d^2\right)$ weights in, say, an ordinary linear regression. That higher-dimensional linear projection, $\mu_N(\mathcal{X})$, has properties that transcend the Bayesian interpretation. For example, under certain conditions it is a best linear unbiased predictor (BLUP). Although much of modern statistical learning now understands Equation (2) in a wider, primarily Bayesian GP context, they are identical to the so-called *kriging equations* [10], which have been instrumental in geospatial/mining analysis for more than half a century.

To illustrate Equation (2) consider $f(x) = 2 + 2\sin(4\pi x)$, observed at $N = 20$ x_i-values uniform in $[0, 1]$ as $y_i = f(x_i) + \varepsilon_i$, where $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1)$. While GP regression is usually applied in higher dimension, such as 2d and beyond, the 1d setting is convenient for visualization. These 20 (x_i, y_i) pairs comprise of our "training data" (Y_N, X_N). Suppose we had a dense testing grid of $N' = 1000$ testing locations $\mathcal{X}$ covering $[0, 1]$. Applying Equation (2) provides us with an N' vector of predictive means $\mu_N(\mathcal{X})$ and an $N' \times N'$ matrix of predictive covariances $\Sigma_N(\mathcal{X})$ summarizing our regression of y onto x. Variances $\sigma_N^2(\mathcal{X})$ along the diagonal of $\Sigma_N(\mathcal{X})$ could be used to build error-bars describing a predictive interval (PI) as roughly $\mu_N \pm 2\sigma_N$ for 95% coverage.

These quantities are shown for one example of such data in the left panel of Figure 1. The middle panel and table on the right will be discussed in Section 2. Observe how noisy data evaluations (solid dots) dance around the true unknown function f (black line); our prediction(s) μ_N and PIs in (blue/red lines, dashed respectively) in two variations (labeled "GP" and "Kriging") accurately distill the essence of the input–output relationship. All of this is modulo a fortuitous choice for $\Sigma(\cdot,\cdot)$ which we have yet to detail. Its specification, and method of inference or unknown quantities, comprises the wedge between modern GPs (red in the figure) and traditional kriging (blue).

2.2 | Modeling

The discussion above hinges on a choice of $\Sigma(\cdot,\cdot)$, or S in Equation (1). S was merely used as a notational device to delay discussion until this moment; we shall not use S going forward. Yet, that Equation (1) formulation is attractive because it abstracts all modeling details down to this "one" choice. "One" is in quotes because $\mathcal{O}\left(N^2\right)$ quantities, one for each pair of N data elements (x_i, y_i),



	GP	K-EB	K-NLS
τ^2 or σ^2	1.99	3.17	2.48
$\tau^2 g$ or τ_k^2	0.145	0.018	0.00010
θ or R	0.17^2	0.17	0.13
RMSE	0.14	0.084	23.11
score_t	869	-4161	-125514

FIGURE 1 Left: A 1d example function in black with dots being the observed locations. The GP prediction is in blue with dashed lines being the 90% predictive interval (PI). Kriging predictions are shown in red and magenta with EB and NLS referring to different variography techniques. Middle: A Gaussian kernel/semivariogram fit to the function. "EB" denotes a semivariogram fit by hand, whereas "NLS" uses non-linear least squares; "GP" derives the semivariogram from an MLE hyperparameterization. Right: Compares hyperparameter estimates and contains out-of-sample RMSE and score.

are actually constrained by the covariance structure. This vast number of potentially tunable quantities, more even than N, is why one refers to GPs as nonparametric. But of course, it's neither practical nor valid to allow oneself such unbridled freedoms. For example, we must choose $\Sigma(\cdot,\cdot)$ so that Σ_N is finite and positive definite for use as an MVN covariance.

An inverse-distance-based covariance is conventional as an intuitive spatial modeling device. However, this is not a requirement and may not be ideal in all situations, for example, when modeling periodic effects. We may wish to allow flexibility in how distances are measured, in what coordinates and with what decay in inversion, and to control how such choices relate signal to noise. Such considerations lead to frameworks for choices of $\Sigma(\cdot,\cdot)$ whose tunable quantities, sometimes called *hyperparameters* to acknowledge a nonparametric modeling apparatus, can be learned from data.

For example, if the range of the responses Y_N is unknown a priori we might wish to design $\Sigma(\cdot,\cdot)$ to include a scale hyperparameter, say τ^2. If Y_N is noisy and/or contains measurement error, we may wish to encode it as part signal, and part noise. Sometimes this is governed by a so-called *nugget* hyperparameter, which we shall denote as g. We caution that the role of GP nugget is inspired by, but is subtly different from, a parameter of the same name in the geostatistics/kriging literature. More in Section 2.4. We may wish to control the smoothness and rate of decay of correlation of the signal in terms of (inverse) distance, and thereby the smoothness and other properties of the underlying response surface. This may be accomplished through selection of a so-called *kernel* function $k_\theta(x_i, x_j) : \mathbb{R}^d \to [0,1]$ whose hyperparameter θ can be used to describe the rate of radial decay from $k_\theta(x_i, x_j) = 1$ for $x_i = x_j$ down to zero as x_j moves away from x_i in an isotropic modeling context. Kernels k may also re-scale and/or rotate the space for anisotropic effects. One way to put these elements together is

$$\Sigma(x_i, x_j) = \tau^2\left(k_\theta(x_i, x_j) + g\delta_{ij}\right) \quad \text{so that} \quad \Sigma_N = \tau^2(K_N + g\mathbb{I}_N). \tag{3}$$

Above, δ_{ij} is the Kronecker delta function returning 1 when the index $i = j$, that is, when the same training data element appears in both arguments, and zero otherwise, and $K_N \equiv k_\theta(X_N, X_N)$ applying k elementwise as $K_N^{ij} = k_\theta(x_i, x_j)$. Observe that the diagonal of Σ_N is $\tau^2(1+g)$ and all off-diagonal entries are less than or equal to τ^2, and strictly less than for all $x_i \neq x_j$. This discontinuity between diagonal and off-diagonal, as long as $g > 0$, leads to smoothing of the predictive surface when following Equation (2). Otherwise, when $g = 0$ the surface interpolates. Again, this is a little different than the typical geostatistics formulation as explained later in Section 2.4.

Choices for distance-based kernels k preserving positive definiteness and targeting certain other properties abound. See, for example, Abrahamsen [11] or Wendland [12]. The two that are most widely used are the power exponential and the Matérn. These are provided below in an isotropic setting, that is, with radial decay as a function of distance.

$$k_\theta^P(x_i, x_j) = \exp\left\{-\frac{\|x_i - x_j\|^p}{\theta}\right\}$$
$$k_\theta^M(x_i, x_j) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\|x_i - x_j\|\sqrt{\frac{2\nu}{\theta}}\right)^\nu \mathcal{K}_\nu\left(\|x_i - x_j\|\sqrt{\frac{2\nu}{\theta}}\right) \tag{4}$$

In both cases above the hyperparameter θ appears in the denominator, scaling (squared or square root) Euclidean distances between x_i and x_j, and is thus sometimes called the *characteristic lengthscale*. Both have additional hyperparmeters, p and ν, respectively, which must be positive and which our notation does not include in θ because they are usually specified, rather than inferred. These control the of the resulting response surface. When $p = 2$ the power exponential produces infinitely smooth (mean-square differentiable) realizations, and sometimes this special case is called the Gaussian kernel,[1] which we denote as k_θ^G. When $p \neq 2$, the surface is nowhere differentiable. While sometimes such pathological non-smoothness is a reasonable assumption—and in spite of this the predictive surfaces (2) often look smooth—there are better mechanisms for relaxing unreasonable (infinite) smoothness.

Many of the choices in our references above offer higher fidelity control over smoothness (beyond none and infinite). Of those, the Matérn has percolated into the canonical position thanks largely to persuasive technical arguments from Stein [13]. The parameter ν controls this aspect, with higher values leading to greater smoothness, yielding surfaces which are $\lceil \nu \rceil - 1$ mean-square differentiable. Ultimately when $\nu \to \infty$ the Gaussian kernel is recovered as a special case. However, the modified Bessel function $\mathcal{K}_\nu$ can be difficult to work with computationally. Specific settings with $\nu \in \left\{ \frac{3}{2}, \frac{5}{3} \right\}$ have algebraic closed forms (no Bessel functions) which yield degree one and two differentiability, with the latter being most often applied in practice because most interesting dynamics result from at least twice (but not infinitely) differentiable processes.

Both the (isotropic) power exponential/Gaussian and Matérn are *stationary* kernels because they are defined only in terms of displacement $\left\| x_i - x_j \right\|$, so the resulting response surface would have identical dynamics throughout the entire input space. Nonstationary modeling is also possible, but is a lot more difficult in general, for example, see Sauer et al. [14], and further discussion in Section 3.1. However, one can still capture nontrivial dynamics with stationary kernels, for example by deploying several of them simultaneously: sums, products, convolutions (and more) of valid kernels for GP regression (i.e., are positive definite) are also valid, for example, see Rasmussen & Williams [8] or Gramacy [4]. One of the most common applications of this result is to extend to axis-aligned anisotropy by taking a product of kernels applied univariately in each coordinate direction: $k_\theta(x_i, x_j) = \prod_{k=1}^d k_{\theta_k}(x_{ik}, x_{jk})$, abusing the notation somewhat. This can be done with any kernel. Notice here we are introducing a d-dimensional lengthscale parameter $\theta = (\theta_1, \ldots, \theta_d)$, controlling the rate of spatial correlation differentially in each coordinate direction. For the Gaussian kernel, the result is identical to

$$k_\theta^G(x_i, x_j) = \exp\left\{ -\sum_{k=1}^d \frac{(x_{ik} - x_{jk})^2}{\theta_k} \right\}, \tag{5}$$

which is sometimes called the *separable* Gaussian kernel or the ARD Gaussian kernel. ARD stands for *automatic relevance determination* [15], borrowing terminology from early neural networks literature. The idea is that the data can inform on longer lengthscales (less relevant) or shorter ones (more relevant) for each input variable separately. ARD/separable Matérn kernels are also common, but their expression(s) are less tidy so we do not include it here.

There is a one-to-one relationship between vectorized lengthscale in the ARD kernel formulation and with re-scaling inputs X_N, say as a pre-processing step. Rather than scaling each input differently, one can extend to rotations and projections to accommodate less rigid anisotropy either as preprocessing [16] or as a hyperparameterized kernel [17]. We find that such high-powered approaches are overkill for most applications, including the mining ones discussed later.

2.3 | Inference

The models above have tunable quantities, or hyperparameters, that could be set by hand but would ideally be learned from data. We restrict our focus to those which we introduced for $\Sigma(\cdot, \cdot)$, particularly $\phi \equiv (\tau^2, g, \theta)$ with the latter usually vectorized in an ARD setting, but there could potentially be additional quantities which must be estimated from data. There are many criteria and algorithms across several literatures devoted to such "fitting" enterprises, for example, Diggle & Ribeiro [18]. Yet there is a remarkable confluence in modern statistical learning practice when it comes to the near universality of likelihood-based methods when distributional assumptions are being made (like the MVN in Equation (1)). The reason is that no additional criteria need be introduced to commence with learning. One may choose to impose additional assumptions, like priors on aspects of ϕ for a Bayesian approach [7], which is still likelihood-based, or not—simply maximize the likelihood. This is the approach we present here because it is tidy and fast.

[1] *Gaussian* here refers to the expression resembling the density of a Gaussian distribution; it has nothing to do with making a Gaussian assumption, or its use in GP. Such kernels are used in a variety of other contexts.

Equation (1) depicts how observations/outputs (like Y_N), are distributed in relation to inputs (like X_N) and parameters or other structure (like $\Sigma(\cdot,\cdot)$ via hyperparameters ϕ and kernels k). The *likelihood* simply re-frames the density of that distribution, which assigns positive real values to Y_N as a function of parameters (or hyperparameters ϕ, say), the other way around: providing positive reals for ϕ given Y_N. Once in that context, it makes sense to seek out the parameterization that makes the observed Y_N most likely, that is, that maximizes the likelihood. There are two benefits to this approach. One is that it reduces a statistical inference question to an optimization one without introducing auxiliary criteria. The other is that the solution to this optimization, the so-called maximum likelihood estimator (MLE), has special properties that can be used to quantify uncertainty. For a review of likelihood-based inference, see Casella & Berger [19]. Details for GPs are provided in section 1 of our Appendix S1; all Figure 1 quantities labeled as "GP" utilize MLE settings of hyperparameters.

2.4 | Kriging and variography

The main difference between classical kriging and the GP presentation above regards inference for unknown quantities and, in the case of the latter, a more up-front and highly-leveraged distributional assumption (Gaussian) for the response. Both use Equation (2) to form predictions and quantify uncertainty. In geostatistics, these are known as the "kriging equations," even when other aspects historically associated with kriging are not faithfully replicated. Classical kriging focuses on lower input dimension—particularly $d \in \{2, 3\}$ in spatial contexts—and as such prefers isotropic modeling after a suitable transformation of spatial inputs. *Variography* is used to select the kernel and its hyperparameters, rather than the likelihood. This has advantages and disadvantages. Many of the advantages are related to the historically larger training data sets encountered in spatial problems, although that gap is narrowing in wider statistical learning and computer experiments contexts. More on this in Section 3, wherein further distinctions arise. The main disadvantage is that input pre-processing and variogram inspection are inherently hands-on, human driven enterprises, albeit ones enhanced by computational tools. Other differences are more superficial, like naming, symbol choice, and applications of hyperparameters within variography.

The *semivariogram*, or half the *variogram*, denoted as $\gamma(h)$, is the variance of two output y-values that are distance h apart in the input x-space: $\gamma(h) = \frac{1}{2}\mathbb{V}\mathrm{ar}(Y(x+h) - Y(x))$. Implicit in this definition is an assumption of intrinsic stationarity, implying that $\mathbb{E}[Y(x+h) - Y(x)] = 0$, or that the covariance between two y values depends not on position but on relative distance notated by the displacement h between them. If h is calculated using Euclidean distance, intrinsic stationarity implies isotropy. When this is a limitation to effective spatial modeling, one may prescale or rotate the coordinate system. This is often based on expert-judgment of the prevailing variabilities within the input domain, like the direction of an ore body within the geologic topology. As mentioned earlier, a modern GP approach would deploy separable lengthscales (5), or more flexibly parameterized rotations and scales that are learned jointly with other unknowns.

The semivariogram is a theoretical construct that would be hard to specify a priori even with expert knowledge, but simple to observe empirically given data. One estimate of an *empirical semivariogram* could be obtained by binning the data by distance and calculating sample covariances within those bins. Let $N(h_k) = \left\{ (x_i, x_j) : \left\| x_i - x_j \right\| \in I_k \right\}$ where $I_1 = [0, h_1], I_2 = (h_1, h_2], \ldots, I_k = (h_{k-1}, h_k]$ denote a neighborhood structure striated by bands of distance $0, h_1, \ldots, h_k$. Then estimate

$$\hat{\gamma}(h) = \frac{1}{2\,|N(h)|} \sum_{(x_i, x_j) \in N(h)} (y_i - y_j)^2. \tag{6}$$

As defined continuously for any h, $\hat{\gamma}(h)$ is a step function. However it is customarily visualized discretely as a scatter plot with $(h_i + h_{i+1})/2$ as the x-axis coordinate. The middle panel of Figure 1 shows these as dots for the 1d problem introduced in the left panel of Figure 1 using a bin size $(h_{i+1} - h_i)$ of 0.05.

One can then match these empirical observations of spatial covariance with a parameterized form for the population semivariogram. Here, similar constructs are used to model spatial dependence as the kernels introduced earlier (4). Let $\gamma(0) = 0$ and for $h > 0$, power exponential and Matérn model semivariograms are often, respectively, written as

$$\begin{aligned} \gamma_\theta^P(h) &= \tau_k^2 + \sigma^2\left(1 - \exp\left\{-\left(\frac{h}{R}\right)^p\right\}\right) \\ \gamma_\theta^M(h) &= \tau_k^2 + \sigma^2\left(1 - \frac{2^{1-\nu}}{\Gamma(\nu)}\left(h\sqrt{\frac{2\nu}{R}}\right)^\nu \mathcal{K}_\nu\left(h\sqrt{\frac{2\nu}{R}}\right)\right). \end{aligned} \tag{7}$$

Semivariogram parameters are known as *nugget* (τ_k^2),[2] *partial sill* (σ^2), and *range* (R).

[2] A subscript k is not standard; we added it to distinguish with the GP scale τ^2.

Taking $\gamma(0) = 0$ is a contentious choice outside of the geospatial modeling literature. It implies that there is no intrinsic variance in measurements. In part this is because such measurements are inherently unrepeatable in certain contexts; you cannot dig a borehole in the same place twice. But if you could, it stands to reason that you would get different measurements for such *replicates* for all sorts of reasons, for example, even without operator error the drill bit might interact with the surface and ore body differently the second time. Primordial process producing the ore body are subject to uncertainties that are best characterized as random variables even if the process is not inherently stochastic. Thus, it is acknowledged that there will be small-scale variability between *nearby* observations that are best described by noise. This noise, at all distances $h = \varepsilon > 0$, is what is parameterized by the nugget τ_k^2. The distinction with the GP nugget g, which characterizes the noise as $\tau^2 g$ at $h = 0$, is thus subtle. Choosing $\gamma(0) = 0$ has an impact on the kriging equations (2), leading to discontinuities at the training data locations, where an otherwise smooth predictive surface would be pocked with spikes of "interpolation." We do not show these in the red curve by deliberately omitting X_N values from our predictive grid $\mathcal{X}$ for Figure 1 (left panel) for aesthetic reasons.

Kriging-versus-GP distinctions between the other two kernel parameters are more superficial. The partial sill σ^2 controls the maximum covariance as $h \to \infty$. This has a 1:1 correspondence with τ^2 from earlier. Sometimes a *sill* parameter, $\tau_k^2 + \sigma^2$, is preferred by geostatisticians instead. The range R controls the distance between maximum and minimum covariance, and plays an identical role as the square root lengthscale: $R = \sqrt{\theta}$. It is not uncommon to instead specify a decay parameter $\phi = 1/R$, and such inversions are common in the GP literature as well.

Each setting of these parameters could be used to overlay a curve onto the middle panel of Figure 1. For example, using the MLE hyparameters from our earlier GP analysis yields the curve in blue. Alternatively, one could automate a search for the "best fitting" variogram parameterization with a generalized/nonlinear (possibly weighted) least-squares (NLS) criterion [20]. This corresponds to the magenta curve. Observe that neither of these result in a terrific fit to the semivariogram "data." An outlying pair of dots near $h = 0.5$ drags these variograms down, sacrificing fit for smaller pairwise distances. One reason these are outlying may be that we have many fewer long-distance pairs in the data than short distance ones. A common remedy would be to downweight or altogether ignore these when fitting the variogram parameters, focusing only on the short distance readings. We refer to one such fit as the "eyeball" (or EB) variogram in the figure, although in practice NLS may similarly be deployed. This can lead to more accurate predictions out-of-sample, as we demonstrate momentarily. However it has the downside of introducing non-statistical (e.g., NLS) and non-metric (determination of outlying semivariogram estimates) criteria diminishing reproducibility and automation, and incurring the expense of human intervention. This enterprise is also sensitive to other choices such as bin size $h_{i+1} - h_i$ and a choice of maximum distance to calculate the empirical variogram. We chose $h_{\max} = 0.5$ for the middle panel of Figure 1, but could have gone out to $h_{\max} = 1$, producing a much "noisier" empirical semivariogram.

The table in the right panel of Figure 1 details hyperparameter estimates for each of the three techniques. Lengthscale and range settings exhibit high agreement. For scale/-partial sill, NLS and GP MLE values are "drawn down" by the noisier higher distance bins relative to our EB alternative which ignored those values. The nugget is where things start to substantially diverge: $\tau^2 g \gg \tau_k^2$ means our GP-MLE estimates more noise/less signal than the kriging alternatives (EB and NLS). Notice that EB and NLS nuggets are on different orders of magnitude. The tiny NLS τ_k^2 may be attributed to a lack of small distance pairs, and consequently the optimizer converged at the boundary of our search space for that parameter: 10^{-4}, meaning very high signal/low noise. Although this seems innocuous when it comes to the corresponding semivariograms on the left in the figure, the implications out-of-sample are severe.

In the left panel of Figure 1 the EB kriging fit (solid-red) is visually similar to the GP-MLE fit (solid-blue) except perhaps near $x = 0.4$. This is noteworthy in light of the disparate parameterization and semivariograms in the middle panel of Figure 1, and in particular the human intervention required to ignore outlying values in favor of short distances. Qualitatively, the red curve may be more accurate compared to the truth (black), but closer inspection reveals a more pernicious concern despite higher accuracy: poor uncertainty quantification. The red 90% PI (error-bars) cover only about half of the training data locations, suggesting that nominal coverage has not been achieved. In contrast, the blue (GP-MLE) error-bars cover many more of the data points. In the table residing in the right panel of the figure, we use root mean squared error (RMSE, lower is better) and score (higher is better) to compare methods out-of-sample. Formulas and commentary are provided in section 2 of our Appendix S1.

3 | LARGE-SCALE MODELING VIA LOCALIZATION

In the modern age, datasets can easily push into the multimillions and the methods described in Section 2 quickly become infeasible. The reason is that likelihood-based GP

inference for hyperparameters and prediction (2) requires matrix decomposition for the inverse and determinant of the covariance structure. Most kernels, for example, Gaussian and Matérn, produce dense K_N (and thus Σ_N), which require $\mathcal{O}(N^3)$ decomposition, say via Cholesky (which furnishes both inverse and determinant). This is prohibitive for N larger than a few thousand. For example, decomposing a single $40{,}000 \times 40{,}000$ matrix on a workstation using 8 cores and specialized linear algebra libraries (Intel MKL) takes about 10 min. Numerical optimization of hyperparameters might require hundreds of such decompositions in search of the MLE via BFGS. With cubic scaling for larger N, computation time quickly explodes to hours or days. $\mathcal{O}(N^2)$ storage of the $N \times N$ matrix can also become problematic even on the most powerful workstations. Kriging-based inference for hyperparameters via variography bypasses the need to work with an $N \times N$ covariance matrix by binning the data. However, Equation (2) still requires a dense $N \times N$ inverse to furnish predictions, which is still cubic in computational order.

Consequently, there are increasingly many approaches seeking a thrifty approximation to GP/kriging models. Heaton et al. [21] give a thorough comparison of about a dozen recently introduced spatial methods equipped to handle large data. Here we focus on three representative approaches as a means of spanning myriad alternatives in a mining context: Ordinary kriging (OK) [1] is the standard method in mining/geostats which makes local (approximate) prediction after full-data variogram-based hyperparameter estimation; Local Approximate Gaussian Processes (LAGP) [23] can be seen as a likelihood-based contemporary analog of OK developed in the surrogate modeling community, making it a natural comparator to OK; finally, the scaled Vecchia approximation (SVecchia) [5] uses a global approximation to estimate the full covariance structure based on similar locality principles as LAGP/OK. Details for each of these follow in subsections below.

Last as a baseline, we consider subset GPs trained via a randomly selected, computationally feasible, $m \ll N$-sized subset of the data points and use them to form an approximation to the full model. Specifically, we build $(X_m, Y_m) \subset (X_N, Y_N)$, using the likelihood for hyperparameter inference via using (X_m, Y_m) and prediction similarly following (2). We consider m ranging from 1000 to 8000; we show later in Figure 5 that $m > 8{,}000$ is very slow and not competitive with the other methods in terms of accuracy out-of-sample.

3.1 | Transductive modeling

Perhaps the most common solution to big-data matrix issues when predicting via kriging is to deploy what is known as *ordinary kriging* (OK) [1]. OK involves using full-data variography to learn kernel hyperparameters, and then a *local* application of that learned kernel through predictive equations (2) conditioned only on a small subset of the data nearby the predictive location(s) of interest. Let $x \in \mathcal{X}$ denote the coordinates of one such location, and $X_m(x) \subseteq X_N$ denote the m "closest" (e.g., via Euclidean distance) members of X_N to x, and let $Y_m(x)$ be the m-associated output values. These are sometimes called the *m-nearest neighbors* (NN) to x in X_N. Then simply apply Equation (2) with $(X_m(x), Y_m(x))$ rather than (X_N, Y_N). When N is so large that the requisite $N \times N$ matrix decompositions are intractable, choosing $m \ll N$ like $m = 50$ can represent a thrifty-yet-accurate alternative acknowledging that the discarded points $X_N \backslash X_m(x)$ have vanishingly small impact on the predictive equations especially when kernels involve exponential decay.

If a multitude of $x \in \mathcal{X}$ are of interest, these may be processed in serial or, as is increasingly common with modern computing architectures, in parallel on multiple cores of a workstation and/or nodes of a supercomputer. Vast predictive grids $\mathcal{X}$ can be processed efficiently in this manner. There are several variations on this theme, many involving how the "neighborhood" $X_m(x)$ and its size m are defined. For example, one may work with a radius r instead, implicitly defining m depending on the local nature of design locations X_N nearby x. Suitable r from a modeling perspective may be selected by the estimated range R of the semivariogram. However, this does not guarantee a suitably-sized m for all x. One may end up with too small of a neighborhood to make computationally stable calculations/reliable low-variance predictions, or too large of one to be carried out efficiently from a computational perspective. Consequently, there are many hybrids that are often deployed in this space [24].

The idea of tailoring a statistical calculation to a predictive task, using different data and possibly different calculations depending on the predictive location x of interest, is now known as *transductive learning* [25]. The transductive moniker is meant to contrast with the more typical *inductive learning* setup where one trains first and predicts second. Under transductive learning, the training happens bespoke to each $x \in \mathcal{X}$, and usually on-demand/in real time. Examples span the gamut of statistical modeling enterprises, often offering both speed and accuracy gains over the inductive analog. Reviewing these would be a distraction here. Instead, we note that OK is an example of transductive learning ahead of its time, albeit a somewhat limited one. Hyperparameter learning with OK is inductive, whereas posterior predictive conditioning is transductive. It is this latter stage where the nonparametric flexibility really comes from, although one might wonder

whether things could improve by enhancing the degree transductively, as it were.

A prime example of transductive GP modeling from the wider statistical learning literature is the *local approximate Gaussian process* (LAGP) [23]. LAGP is similar to OK, using $X_m(x) \subset X_N$ and $Y_m(x)$ analogously for prediction, but it is different in that it extends the notion of locality to hyperparameter inference via the (local) likelihood. That is, the entire process conditions only on $(X_m(x), Y_m(x))$ for inference and prediction (2). All inference is off-loaded to numerical optimization. When the response surface is nonstationary, for example, benefiting by longer lengthscales for some x and shorter for others, LAGP offers enhanced reactivity compared to single, global setting of hyperparameters. Any variation tailored to the full-GP is easy to port to the local setting because LAGP is just many small GPs. Hybrids are possible too. One such example alluded to earlier involves pre-scaling or rotating/projecting [16, 26] to handle non-axis-aligned anisotropy. However, the biggest difference between LAGP and OK does not lie in potential for extension; it is how the neighborhood is defined.

Given fixed m, usually chosen via computational considerations (a common default is $m = 50$), it has been known for sometime that the m-NNs in X_N to x, whether via Euclidean distance or otherwise, do not comprise of an optimal conditioning set $(X_m(x), Y_m(x))$ under any reasonable criteria [27]. Example criteria include (Fisher) information about unknown hyperparameters or, as is usually more relevant when predictive accuracy is concerned, predictive uncertainty (a.k.a., mean-squared prediction error). We note that this, in turn, means that the OK predictor is also sub-optimal as a transductive learner. However searching for the best conditioning set $(X_m(x), Y_m(x))$, again under almost any criteria, represents a computationally daunting task because there are $\binom{N}{m}$ alternatives to explore.

Here, another modern statistical learning idea comes in handy: *active learning* (AL). AL is a branch of reinforcement learning/optimal control, or may be viewed as a modern take on sequential design of experiments. In the AL literature, one can often show that a one-step-at-a-time, *greedy* selection of training data is nearly as good as an exhaustive optimization of some criteria if it obeys a *submodularity* property [28]. For example, it can be shown that repeatedly acquiring training data (x_{n+1}, y_{n+1}) such that x_i maximizes the predictive variance $x_{n+1} = \text{argmax}_x \ \sigma_n^2(x)$ of a GP (2) or neural network model training only on $\{(x_i, y_i)\}_{i=1}^n$ obtained previously (e.g., via similar greedy optimization), well-approximates a so-called maximum entropy design, that is, maximizing Shannon information about unknown hyperparameters (GP lengthscales) for the entire selection $i = 1, \ldots, N$, say. This idea is due to [29] for neural networks, and dubbed ALM by [30] in extension to GPs.

Intuitively, selecting points which have maximum variance will result in a space-filling design because variance is higher away from the training data. Also intuitively, spreading points out will increase accuracy and reduce uncertainty throughout the input space. But this is coincidental. Guaranteeing reduced predictive variance everywhere, or at a particular location (for choosing an LAGP neighborhood), requires a criteria that squarely targets reduced variance in the region of interest. A common choice is integrated mean-squared predictive error (IMSPE) [2]:

$$\begin{aligned} \text{IMSPE}(x_{n+1}) &= \int_{\mathcal{X}} \sigma_{n+1}^2(x) \ dx \approx \sum_{x \in X_{\text{ref}}} \sigma_{n+1}^2(x) \\ &= \text{ALC}(x_{n+1}, X_{\text{ref}}). \end{aligned} \tag{8}$$

The integral is usually taken over the entire input space, but $\mathcal{X}$ could be any set. This could be interpreted as a criteria for the entire design $X_{n+1} = [X_n; x_{n+1}]^{\text{T}}$, or simply to select the next input x_{n+1} in an active learning context: $x_{n+1} = \text{argmin}_{x \in \mathcal{X}} \ \text{IMSPE}(x)$. Due to submodulatirty, both (approximately) optimize the IMSPE criteria over all X_{n+1}. Cohn [31] developed this for neural networks, and Seo et al. [30] extended it to GPs. Notice that ALC approximates the integral as a quadrature over a discrete reference set X_{ref}. This is not necessary for GPs, because the integral is analytic when following Equation (2), but it is for neural networks.

For LAGP, the goal is to get as accurate of a prediction at x as possible, which can be interpreted as a singleton $\{x\} = \mathcal{X} = X_{\text{ref}}$, effectively discarding the sum or integral. We can select a new $x_{n+1} = \text{argmin}_{x_{n+1}} \ \text{ALC}(x_{n+1}, x)$, and repeated applications will approximate a "local" optimal design for predicting at x. This would usually be applied for selecting new training data in an AL context, but for LAGP we already have a fixed training data set (X_N, Y_N) and so we desire a subsample instead: $x_{n+1} = \text{argmin}_{x_{n+1} \in X_N \setminus X_n} \ \text{ALC}(x_{n+1}, x)$, which is even easier than a continuous search everywhere in the input space.

In practice, early ALC acquisitions (small n) result in neighborhoods that are indistinguishable from NN. However, later acquisitions (larger n) tend to concentrate on "satellite" points farther away. Information farther afield becomes more valuable as NNs accumulate near x: you want x_{n+1} to be both close to x but far from X_n. At the start the former dominates, but eventually the latter has higher weight in the criteria. The end result for $n = 50$, seen in

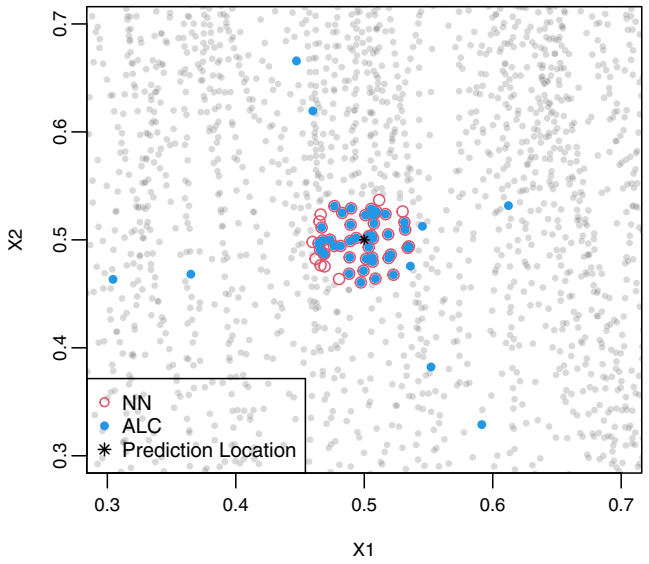


FIGURE 2 ALC neighborhood for $n = 50$ on simulated borehole data.

Figure 2, involves 10 or so satellite points, departing from an OK or NN subdesign set of the same size.

To a certain extent, LAGP has a "chicken or the egg" problem when dealing with anisotropy. Neighborhoods selected for x are based on Euclidean distance to determine hyperparamters, like $\widehat{\theta}_k(x)$, but those values control notions of distance differently in each input coordinate through the kernel. OK experiences this problem too, albeit to a lesser extent when anisotropy is handled by the practitioner as a pre-processing step. Several remedies have been proposed. The original LAGP paper [23] suggested initializing with a default, isotropic θ_0 for all testing locations x, upon which neighborhoods are built (e.g., via ALC) and local, anisotropic $\widehat{\theta}(x)$ are learned through local MLEs separately for each x. This can then be repeated, with neighborhoods based on those $\widehat{\theta}(x)$, until things stabilize. Subsequently, a simpler/better approach was promoted by Sun et al. [26] that is more akin to OK pre-processing, but still completely automated. Liu & Hung [32] showed that unbiased (global) lengthscales $\widehat{\theta} = \left(\widehat{\theta}_1, \ldots, \widehat{\theta}_d\right)$ can be estimated via MLEs from carefully constructed data subsets of large X_N without expense cubic in N. Once these have been learned, they can be used to pre-scale inputs X as $X_k/\sqrt{\theta_k}$ so that the implied MLE global lengthscale is $\widehat{\theta}_0 = 1$ under squared-distance kernels like in Equation (4). In this transformed space, Euclidean distance can be used to determine neighborhoods. This pre-scaled LAGP has become the default setup, and has since been extended to other input "warpings" [16].

Taken as a predictive field over a densely gridded testing set of x-values, both OK and LAGP (NN or ALC) are discontinuous. Prediction at each point $x \in \mathcal{X}$ is processed independently, both in a statistical and computation sense. So two testing locations right next to each other may have substantively different predictions (in mean and/or variance) because different neighborhoods are used for conditioning. This can be an asset when the response surface exhibits regime/abrupt changes. In tamer, more stationary settings, a non-smooth prediction could be detrimental to statistically efficient and aesthetically pleasing analysis.

3.2 | The scaled Vecchia approximation

To address that potential downside, the Vecchia GP approximation [33] borrows the neighborhood idea while providing a global model for smooth predictions. It relies on a familiar identity for joint distributions:

$$\begin{aligned} p(y) &= p(y_1)p(y_2|y_1)\ \ldots\ p(y_n|y_1, y_2, y_3, \ \ldots\ y_{n-1}) \\ &= \prod_{i=1}^{N} p\left(y_i|y_{k(i)}\right) \qquad \text{where} \quad k(i) = \{j\ :\ j < i\} \\ &\approx \prod_{i=1}^{N} p\left(y_i|y_{c(i)}\right) \qquad \text{where} \quad c(i) \subset k(i) \end{aligned} \tag{9}$$

The first line above (equality) is true for any re-indexing of the variables $y = y_1, \ldots, y_n$, and for any y—not specifically for GPs. The approximation (second line) arises from dropping some of those conditioning variables. Let m denote the maximum size of those sets, that is, so that $|c(i)| = \min(i-1, m)$, controlling the fidelity of the approximation—more severely for $m \ll i$. The quality of this approximation is determined by the indexing (i.e., the ordering of the conditionals), size m, and which of the conditioning variables $k(i)$ are dropped in $c(i)$ when $m < i$.

Specifically for GPs, one may view the likelihood, in this context:

$$\begin{aligned} L(\phi; Y_N) \approx \prod_{i=1}^{N} L\left(\phi; y_i|y_{c(i)}\right) &= (2\pi)^{-\frac{N}{2}} \left(\prod_{i=1}^{N} \sigma_i^2\right)^{-\frac{1}{2}} \\ &\exp\left\{-\sum_{i=1}^{N} \frac{1}{2\sigma_i^2}\left(y_i - \Sigma\left(x_i X_{c(i)}\right)\Sigma\left(X_{c(i)}, X_{c(i)}\right)^{-1} y_{c(i)}\right)^2\right\} \end{aligned} \tag{10}$$

where $\Sigma(\cdot,\cdot)$ is defined as in Section 2.1 and $\sigma_i^2 = \Sigma(x_i, x_i) - \Sigma\left(x_i, X_{c(i)}\right)\Sigma\left(X_{c(i)}, X_{c(i)}\right)^{-1}\Sigma\left(X_{c(i)}, x_i\right)$ is the predictive variance at location i given the conditioning set, $c(i)$. Since distance in the input space, via $\Sigma(\cdot,\cdot)$ is fundamental to GP inference and prediction, one can think of $c(i)$ as defining a "neighborhood." In that context it makes sense (as it did

for OK and LAGP) to include in the neighborhood those indices whose input values $X_{c(i)}$ are closer to x_i.

Equation (10) is similar to the likelihood except instead of performing one $N \times N$ matrix decomposition (for inverse and determinant) in $\mathcal{O}\left(N^3\right)$ time, the Vecchia approximation involves N smaller inversions of $m \times m$ matrices, requiring $\mathcal{O}\left(Nm^3\right)$ flops. If m is small, typically between 10 and 25 [34, 5], $\mathcal{O}\left(Nm^3\right)$ is quasilinear in N [35]. Further computational gains can be realized through sparse-matrix libraries and parallelization by re-writing the likelihood through a Cholesky decomposition of the precision matrix of Y_N, denoted as U:

$$\begin{aligned} L(\phi; Y_N) &\approx (2\pi)^{-\frac{N}{2}} \left(\prod_{i=1}^{N} \sigma_i^2\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{N}\left(Y_N^{\mathrm{T}} U_i U_i^{\mathrm{T}} Y_N\right)\right\} \\ &= (2\pi)^{-\frac{N}{2}} \left|UU^{\mathrm{T}}\right|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(Y_N^{\mathrm{T}} UU^{\mathrm{T}} Y_N\right)\right\}, \end{aligned} \tag{11}$$

where U_i is a $1 \times N$ vector whose j^{th} entry is:

$$U_i^{(j)} = \begin{cases} \frac{1}{\sigma_i} & 0 \\ -\frac{1}{\sigma_i}\left(\Sigma\left(x_i, x_j\right)\left(\Sigma\left(X_{c(i)}, X_{c(i)}\right)^{-1}\right)^{(j,j)}\right) & j \in c(i) \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Observe that there are no $N \times N$ matrix decompositions, and that any inverses are implicit in the sparse Cholesky factor UU^{T}.

One may maximize the likelihood (11) to estimate hyperparameters [36]. Prediction follows the classical setup (2), forming $Y(\mathcal{X}) \mid Y_N$ by stacking training and testing responses:

$$\begin{bmatrix} Y_N \\ Y(\mathcal{X}) \end{bmatrix} \sim \mathcal{N}_{N+N'}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} UU^{\mathrm{T}} = \sum_{i=1}^{N} U_i U_i^{\mathrm{T}} & \sum_{i=1}^{N} U_i \sum_{i=1}^{N'} U_i'^{\mathrm{T}} \\ \sum_{i=1}^{N'} U_i' \sum_{i=1}^{N} U_i^{\mathrm{T}} & U'U'^{\mathrm{T}} = \sum_{i=1}^{N'} U_i' U_i'^{\mathrm{T}} \end{bmatrix}^{-1}\right).$$

Here we have introduced a new $N' \times N'$ matrix, U' following Equation (12), via $\mathcal{X}$ rather than X_N. Then, the analog of Equation (2) yields

$$\begin{aligned} \mu_N(\mathcal{X}) &= -\left(U'U'^{\mathrm{T}}\right)^{-1} \sum_{i=1}^{N'} U_i' \sum_{i=1}^{N} U_i^{\mathrm{T}} Y_N \quad \text{and} \\ \Sigma_N(\mathcal{X}) &= U'U'^{\mathrm{T}}. \end{aligned} \tag{13}$$

So the Vecchia GP approximation provides a full joint distribution. Moreover, a single prediction ($\mathcal{X} \equiv \{x\}$), after training, requires just $\mathcal{O}\left(m^3\right)$ additional time [5], assuming cached values of U. A total of $\mathcal{O}\left(\left(N' + N\right)m^3\right)$ flops are required for inference and prediction, as opposed to $\mathcal{O}\left(N'^3 + N^3\right)$ for an ordinary GP.

All that remains is to determine the ordering of indices i in y_i and the composition of the neighborhood sets $c(i)$, since not all choices (when $m \ll n$) lead to equally good approximations (9–10). One option is to follow the LAGP playbook and attempt to optimize over these variables. However this has proved elusive in the literature because an exhaustive search over alternatives would be combinatorially cumbersome, and there is no obvious greedy approach that enjoys submodularity for active learning. Nevertheless there are rules of thumb that make sense intuitively. Many orderings work well [27, 36, 37], but there is a consensus in the literature [34, 38, 39] for random indexing. Likewise, those authors prefer NN conditioning sets $c(i)$ comprised of indices $j < i$ whose x_j-values are closest to x_i. This choice has been dubbed NNGP by Datta et al. [34], although it is important to note that NN are not being used in the same way as LAGP or OK.

Since distances are involved in NN calculations, the Vecchia approximation faces the same "chicken or the egg" problem as LAGP in the face of anisotropy. To help, Katzfuss et al. [5] describe a scheme similar to pre-scaling for LAGP which updates lengthscales $\widehat{\theta}_k$ via Fisher scoring [40], then re-scales inputs so that NNs can be recalculated, and repeats. Katzfuss et al. [5] call this "scaled Vecchia" (SVecchia), and argue that it works best with a maximin [41] indexing. We adopt SVecchia as our preferred variation on this theme, in part because it is neatly packaged in software (Section 4).

Figure 3 shows an illustration using simulated borehole data, providing conditioning sets of size $m = 10$ for two points, labeled with indices $i = 4$ (triangles) and $i = 115$ (circles) respectively. The left plot shows what the conditioning sets look like in the raw, unscaled space while the right is after scaling x_1 by $\frac{2}{3}$; both using the same maximin ordering for easy comparison. The scaling has no effect on the small indices, since point 4, for example, can only condition on $k(4) = c(4) = \{1,2,3\}$. See that the lower indices in $c(4)$ (purple triangles) are spread out through space because of maximin ordering. However, point 115 has $|k(115)| = 114$ points to choose from for it's neighborhood $c(115)$. Consequently, the conditioning sets are different between the unscaled and scaled versions. Observe that point 97 is closer than point 34 in the scaled plot, but farther in the unscaled plot.

4 | EMPIRICAL EVALUATION ON ORE DATA

Here we shall expand on our out-of-sample analysis to illustrate how modern approximate GP and kriging alternatives (Section 3) compare on two, real and large-scale ore data sets. The layout is as follows. Section 4.1 presents

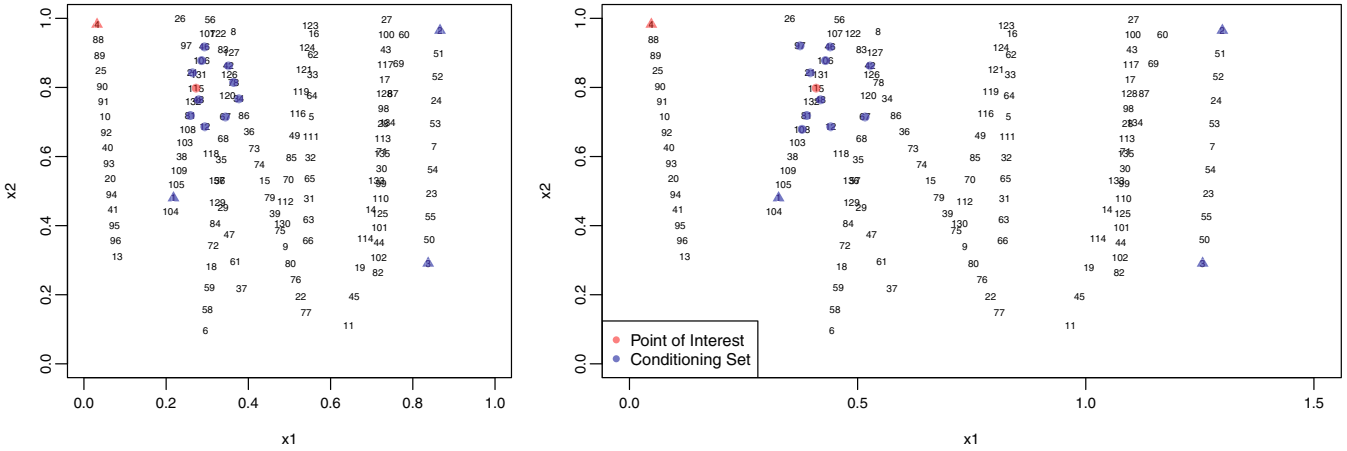


FIGURE 3 Conditioning sets for two inputs using $m = 10$: point 4 (triangles) and 115 (circles). Left shows the original space in coded inputs, and right shows x_1 pre-scaled by $1/\sqrt{\theta_1} = 2/3$.

the data and our validation apparatus, which offers a subtle twist on conventional methods to respect the borehole nature of data acquisition/measurement. Here we also provide results from our first sets of comparisons. Finally, Section 4.2 presents one of several potentially more nuanced analyses that, we believe, is only possible (with ease) in the fully probabilistic GP setting: coping with left-censoring prevalent in ore measurements. Implementation details are provided in section 4 of our Appendix S1.

4.1 | Validation exercise

Our ore data involve three-dimensional inputs, indicated as longitude, latitude and depth in standard units. Although these data record measurements (potential ore outputs) for multiple elements, our analysis here focuses on (log) gold concentration in parts per million. The two data sets are for geographically disparate mining sites that are characterized by different ore forming processes. The first one records more than 150,000 measurements from approximately 4000 boreholes; the second has $N \approx$ 500,000 from 8000 boreholes. The second data set also has a substantial number of left-censored values (i.e., thresholded measurements below the detection limits of the apparatus used to sample the core). For example, about 40% of the gold measurements in these data are recorded as 0.05. There are a smaller number of higher limiting values as well. We shall detail how we handle this with two different treatments in Sections 4.1 and 4.2. We are deliberately being vague about many aspects of our data to honor confidentiality agreements with mine operators.

We wish to draw an out-of-sample comparison between the methods in Section 3 on these data. In addition to RMSE and proper score, we also report time, considering both compute (machine) time and practitioner (human) time. Machine time is measured precisely, in seconds, for execution on an eight-core hyperthreaded Intel Core i9-9900K CPU at 3.6GHz with 128GB RAM and Intel MKL linear algebra subroutines. Human time is more subjective/imprecise, and we shall have more to say about that in due course. It is worth remarking that none of the small-data/exact methods from Section 2 are applicable when $N \gg 10{,}000$, as we have here. Approximation is essential. One simple option is to (randomly) subset the data to a manageable size and apply exact inference on that subset. We consider variously sized "subset GPs" as a benchmark.

As mentioned in Section 2, it is important to compare metrics on out-of-sample data. In practice, out-of-sample validation occurs by training the model on 90% of the data and testing on the other 10%. A simple example is provided in section 3 of our Appendix S1. Repeating that randomization multiple times mitigates the so-called Monte Carlo (MC) error for metrics like RMSE and score, which are shown in Section 2 of our supplement. Here we use $K =$ 10-fold *cross validation* (CV) [42] to average over train–test partitions while controlling MC error further by ensuring that each data element is used exactly once for testing, and complementarily exactly nine times for training. CV commences by first shuffling the data, and then evenly dividing it into a partition of K mutually-exclusive *folds*, then iterating over those folds $k = 1, \ldots, 10$, forming a testing set of the data in the k^{th} fold while taking the complement as the training set. In this way, K metric evaluations (like RMSE, score or time) can be calculated and summarized for comparison.

Early attempts at a CV evaluation of the methods in Section 3, after this fashion, revealed a shortcoming in the context of our borehole-driven ore data sets.

Namely, the best predictors of a particular held-out testing element were almost always comprised entirely of members of the training data coming from the same borehole. With boreholes "holding" approximately 30–60 data elements each, depending on the hole and the data set, this meant that it was highly probable that an accurate prediction could be made trivially just by those nearby evaluations. This conveyed a substantial advantage to OK and LAGP. We determined that it would be more realistic, and more fair, to hold out entire boreholes for testing, rather than partitioning the data on individual data elements regardless of which borehole they were in. The idea is to simulate what might happen if we were to predict measurements for a new borehole that has not been drilled yet.

Toward that end, we built a custom CV which randomly partitioned our data into K folds of roughly equally-sized boreholes instead. In this way, boreholes are "tested" all at once, without being able to lean on other data within the same borehole for training. Figure 4 shows one such training and testing partition via a single fold of this "borehole-preserving CV." Finally, it is worth remarking that all of our comparator methods use exactly the same CV folds.

4.1.1 | Ore data set one

With this setup, Figure 5 shows log RMSE, score and compute time for each of our methods for the first, smaller data set. The big takeaways are that SVecchia, OK and SLAGP are all competitive with each other in terms of RMSE; SLAGP and SVecchia are competitive in score with similar medians. Observe that SLAGP improves upon ordinary LAGP for both RMSE and score. Score for OK could not be calculated because GSLIB does not furnish predictive variances. GSLIB provides standard errors on the mean of the prediction, but those can substantially under-estimate out-of-sample variance. In terms of time, SVecchia takes seconds and LAGP takes a couple minutes to run, both with essentially zero "human time." We report that OK takes several hours of human time to perform a variography analysis, choosing between competing kernel formulations and parameterization and to determine an appropriate rotation and pre-scaling of the data to cope with an otherwise isotropic formulation. After that has been done, training and prediction takes about the same amount of time as (S)LAGP. It is interesting that SLAGP is faster than LAGP despite involving more algorithmic steps: first fit a global subset model, then local models on transformed inputs. The explanation is that, after pre-scaling, local MLE calculations are easier: they require many fewer iterations to converge. Computation time for the subset-GP methods explodes with $\mathcal{O}(m^3)$ flops as m grows.

Our conclusion from this experiment is that, although SLAGP edges out SVecchia on accuracy and UQ for this problem, SVecchia is slightly better overall because of its substantially lighter time commitment. However, for an individual (or entire borehole) prediction, SLAGP times are orders of magnitude faster—the times in the right panel of Figure 5 are for all boreholes in the fold—because each calculation is independent of others. For a one-off prediction it is the clear winner. Although raw accuracy is similar compared to OK, the modern GP methods are hands-off, provide full UQ, and are faster to train/predict.

4.1.2 | Ore data set two

A similar analysis for the second, larger data set, is nuanced because of the substantial left-censoring. One option is to ignore the censoring and treat the recorded values as the actual values. If there were a small number of such values, sporadically located in the input space, this might work well. However, there are a sizable number (more than 40%), and they cluster nearby one another.

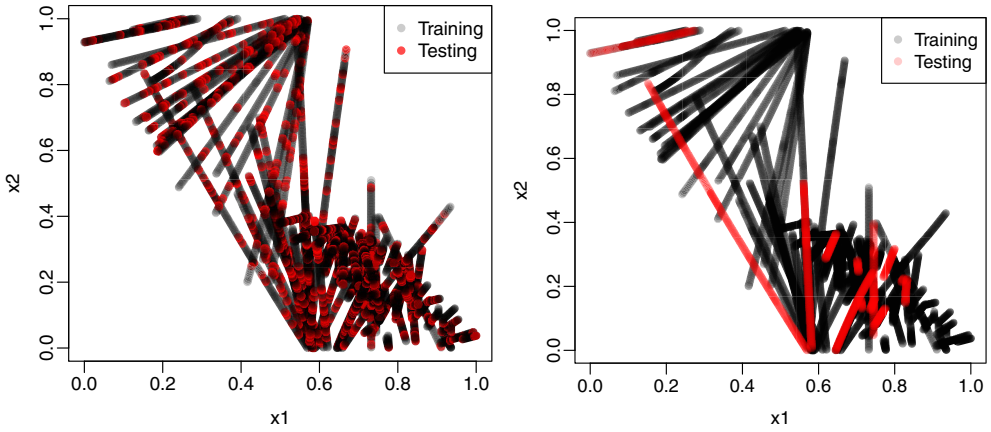


FIGURE 4 Testing and training sets for a 2d project with ordinary CV on the left and borehole-preserving CV on the right.

Having some responses smoothly vary in the input space, with others "flatlining" at 0.05, say, for most or all of a borehole represents an almost pathological contrast to typical smoothness assumptions underlying GP (and kriging) methods. So to start with, we removed these values, and dealt with borehole-preserving CV only on the remaining 259,555 data records. In Section 4.2 we shall discuss an imputation scheme for bringing these observations back into the fold. The remaining data still contain a moderate degree of left-censoring which we largely ignore except when an entire borehole contains the same (thresholded) gold response. In that case, we collapse those records into three data points—two ends and midway point—all with the same gold value. This collapsing is especially important for LAGP and OK because, due to their local nature, those methods occasionally have neighborhoods consisting of data from one or two boreholes only. If those measurements lack diversity due to thresholding, training can result in numerical singularities.

Even after such modifications, we found that LAGP and OK struggled to predict at some testing sites. `GSLIB`, implementing OK, would simply refuse to provide a prediction in these instances, or similarly when there are no training data points within a user-specified radius (regardless of m), returning an error code. In these data, that amounts to about 300 testing sites per fold. The laGP software would furnish a prediction, but when comparing the corpus of other predictions in a fold it was obvious that something was amiss, particularly with the estimated (local) nugget parameter and, consequently, the predicted variance. To investigate, we plotted a histogram of the estimated nuggets from all of the local fits, shown in the left panel of Figure 6, and compared these against the global nugget(s) provided by subset and SVecchia methods. We observed that occasionally, local nuggets were being estimated at the lower-threshold imposed by the `laGP` default search range (leftmost-bin in the histogram). The local neighborhood for one, representative member

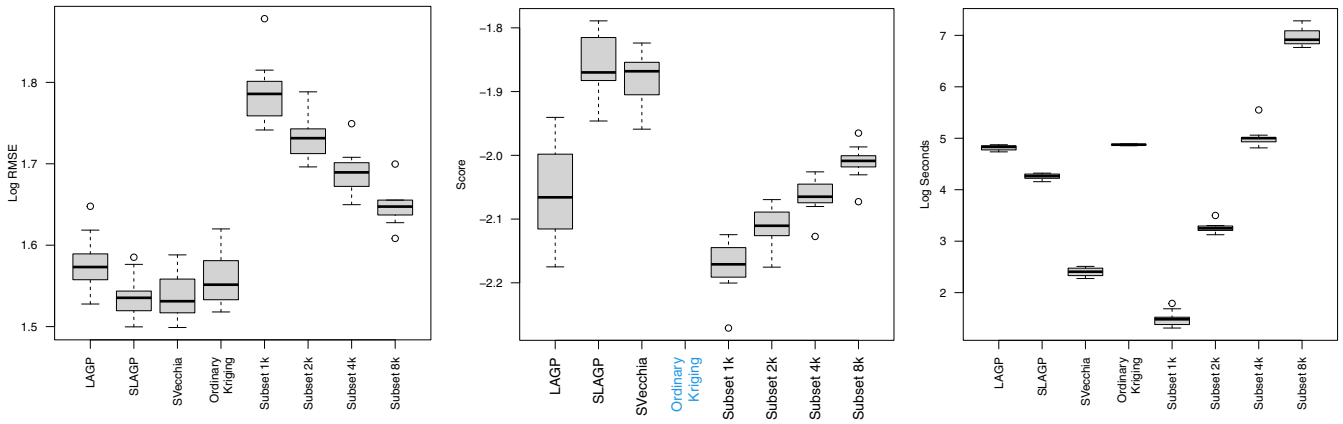


FIGURE 5 Drillhole-presererving 10-fold CV summary for the first data set. Left: RMSE (smaller is better); Middle: score$_p$ (higher is better). Right: compute time (smaller is better).

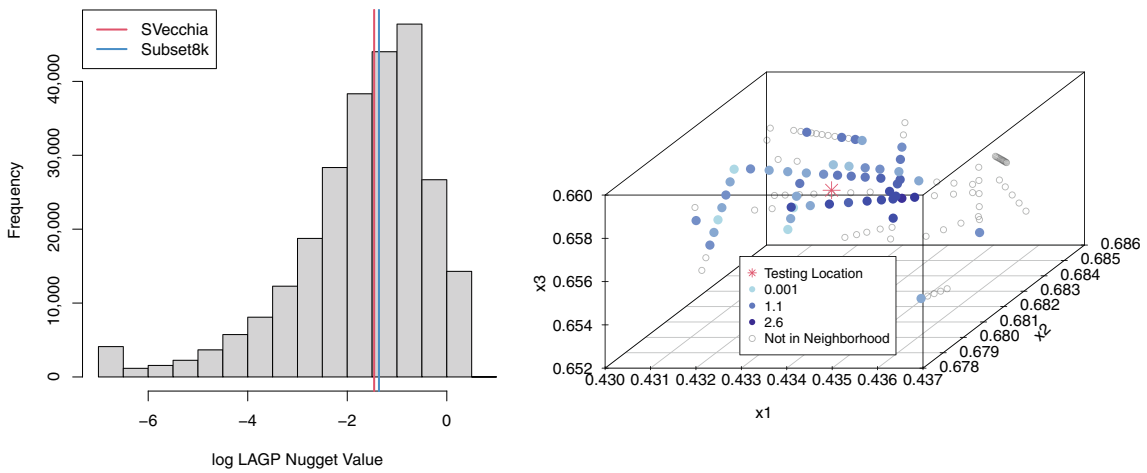


FIGURE 6 Left: Histogram of log LAGP nuggets with SVecchia and subset8k nuggets. Right: An example neighborhood using LAGP's ALC for a point with low estimated variance.

of this group is shown on the right panel of the figure. Observe that most of the points nearby have log gold values above 1 (are similar shades of blue) with some satellite points having lower log gold values. Thus a prediction of log gold a bit higher than 1 with low variance makes sense. However, the true log gold value for this point is about 0.1, meaning the prediction is confident and wrong—which would lead to a poor score evaluation.

Of course, we cannot know this location provides a bad prediction until we look at the testing value for this predictive location. So we decided to replace local nuggets estimated at the lower-bound of `laGP`'s' search range with the median of the nuggets from the rest of the distribution. This led to a substantial improvement in out-of-sample scores, described momentarily. If this sounds ad hoc, that is because it is. But a prediction with UQ that is based on compromise and a limited degree of post-hoc human intervention is better than no prediction at all (OK/ `GSLIB`).

Figure 7 shows these results with black boxplots. (The red ones involve a study on imputation in Section 4.2, so ignore those for now.) The story is similar to the first data set in Figure 5: SLAGP, SVecchia, and OK outperform subsetting in terms of RMSE. Here, OK appears to be the most accurate, but these RMSE calculations do not include any error-coded outputs (representing more than 300 presumably "bad" predictions per fold). So this is not a holistic assessment of OK accuracy. By score, which again cannot be calculated for OK, SVecchia is the clear winner, and the second-fastest in this comparison. (S)LAGP, which is similar in spirit to OK, has inferior scores despite modifications to address stability issues to do with the nugget (above).

4.2 | Imputation

It is unsatisfying to discard data. Even a coarsely left-censored value contains information, which can be used to enhance training. Perhaps even more importantly, one may wonder how accurately those censored values may be predicted, thereby increasing the resolution of those measurements, by borrowing information from higher-accuracy (training) data measurements nearby. This is a standard enterprise in statistical learning when fully probabilistic generative modeling, like the GP, is used. There are many options when handling "missing data," of which censoring is one example [43, 44].

One way to incorporate censored ore values, without destroying smoothness or stationarity assumptions underlying GP spatial models, is through *imputation*, e.g. [44]. In our context, imputation basically means generating a plausible response Y-value for censored locations that both respects the censored measurement, and the smoothness of the underlying spatial field learned through other, completely observed data. Once generated, the imputed value may be treated as if it were a completely observed value going forward, say for prediction. Of course, treating an imputed value as observed ignores the uncertainty in the imputation. *Multiple imputation* (MI) acknowledges that uncertainty by randomly imputing several possible values and performing inference based on the corpus of those imputed values, for example, through averaging.

Illustrating how this could work in our ore context requires some notational scaffolding. Let $D_N = (D_{\text{obs}}, D_{\text{cens}})$ represent the partition of the complete data set into its fully observed and censored components, respectively. For example, $D_{\text{obs}} = (X_{\text{obs}}, Y_{\text{obs}})$, may be the portion of the second data set we were working with in Section 4.1, and $D_{\text{cens}} = (X_{\text{cens}}, Y_{\text{cens}})$ was the part we (temporarily) discarded. Imputed values $Y_{\text{imp}}(X_{\text{cens}})$, may be used to augment D_{obs} to obtain $D_{\text{imp}} = \left(D_{\text{obs}}, \left(X_{\text{cens}}, Y_{\text{imp}}\right)\right)$ via truncated Gaussian simulation

$$Y_{\text{imp}} \sim \mathcal{N}_{N_{\text{cens}}}(\mu_{\text{obs}}(X_{\text{cens}}), \Sigma_{\text{obs}}(X_{\text{cens}}))\mathbb{I}_{\{Y_{\text{imp}} \le Y_{\text{cens}}\}}, \quad (14)$$

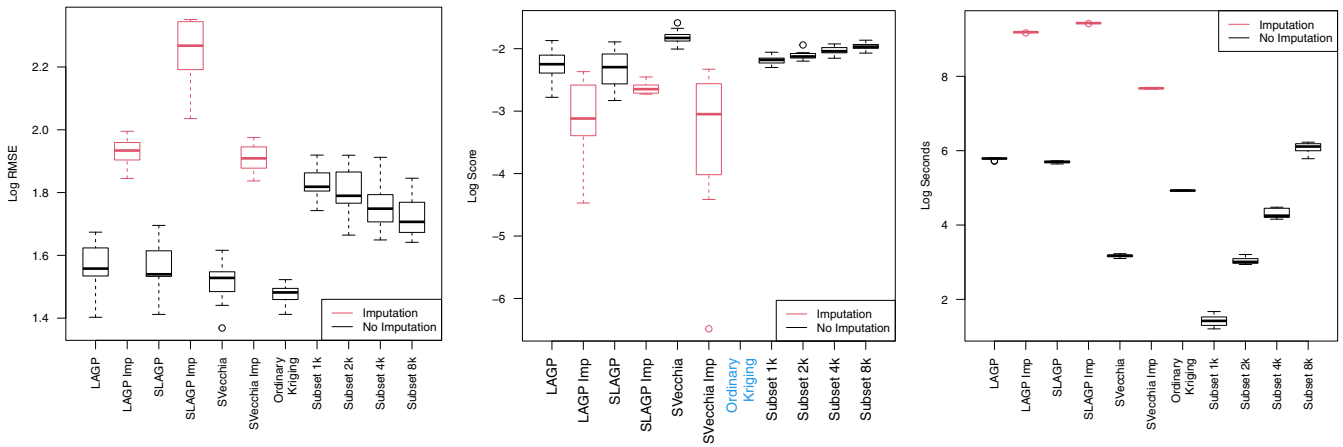


FIGURE 7 Drillhole-preserving 10-fold cross validation summary for the first data set. See Figure 5 caption. Red boxplots are discussion in Section 4.2.

where $\mathbb{I}_{\{Y_{\text{imp}} \leq Y_{\text{cens}}\}}$ is an indicator function, returning 1 if $Y_{\text{imp}} \leq Y_{\text{cens}}$ and 0 otherwise. Quantities $\mu_{\text{obs}}(X_{\text{cens}})$ and $\Sigma_{\text{obs}}(X_{\text{cens}})$ are the predictive moments (2) of a GP fit conditioned on D_{obs}.

Although software exists to sample from a truncated MVN directly, e.g. [45], such as those in (14), in practice it can be difficult to generate a sufficient number of values below Y_{cens} when N_{cens} is of modest size, for example in the hundreds [46]. A more customized approach that acknowledges the form of our (approximate/large-scale) spatial surrogates helps. In the (S)LAGP context, we may use Equation (9) to sample Y_{imp} from the truncated MVN (14) one at a time, conditioning on the previously sampled imputed values and the observed data. LAGP is designed to look at each location in the testing set independently which makes this setup work. On the other hand, SVecchia is designed to give a global model approximation, so doing a similar one-at-a-time conditional imputation is too crude. We instead prefer a bespoke rejection sampling [47] scheme that proceeds in epochs: first generate posterior samples from the MVN (14) unconstrained, keeping any values that satisfy the censoring threshold. Then condition on those imputations and the observed values, resampling at locations without an imputed value, repeating until Y_{imp} is completely filled in.

An algorithm is provided in Section 5 of our supplement with pseudo-code for concreteness, wrapping a single imputation with a `for` to obtain M imputations [48]. indicates that M between two and 10 works well, so we use $M = 5$ in our exercises. Each of the M imputed values are plausible realizations of the censored measurements, which correspond to M posterior/predictive Gaussian distributions for each testing location. Thus we may use Gaussian mixture moment equations [49] to report the mean and variance predictions for the testing set:

$$\mu_{\text{MI}}(\mathcal{X}) = \frac{1}{M}\sum_{i=1}^{M}\mu_i(\mathcal{X})$$
$$\sigma^2_{\text{MI}}(\mathcal{X}) = \frac{1}{M}\sum_{i=1}^{M}\sigma^2_i(\mathcal{X}) + \frac{1}{M}\sum_{i=1}^{M}\mu^2_i(\mathcal{X}) - \left(\frac{1}{M}\sum_{i=1}^{M}\mu_i(\mathcal{X})\right)^2 \quad (15)$$

While this cannot completely account for all possible uncertainties due to imputation, because we have not looked at all possible imputation values (only M), we can always increase M if desired. It may be shown that these equations give an unbiased estimate of mean and variance for any M.

4.2.1 | Imputation in practice

To begin with an illustration in a simple, controlled setting, the left panel of Figure 8 shows a classical GP model with and without imputation. The true function is $f(x) = 2\sin(4\pi x) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0,0.1)$ and an observation threshold of $y = 1$ with $n = 20$ randomly selected training points. There are two main regions of censoring, one in the center and one at the upper end of inputs. In the center, the model with imputation gets closer in mean to the truth. On the upper end, the variance of our predictions is much lower when conditioning on imputed values. Observe that the model with imputation is a better fit for the true curve.

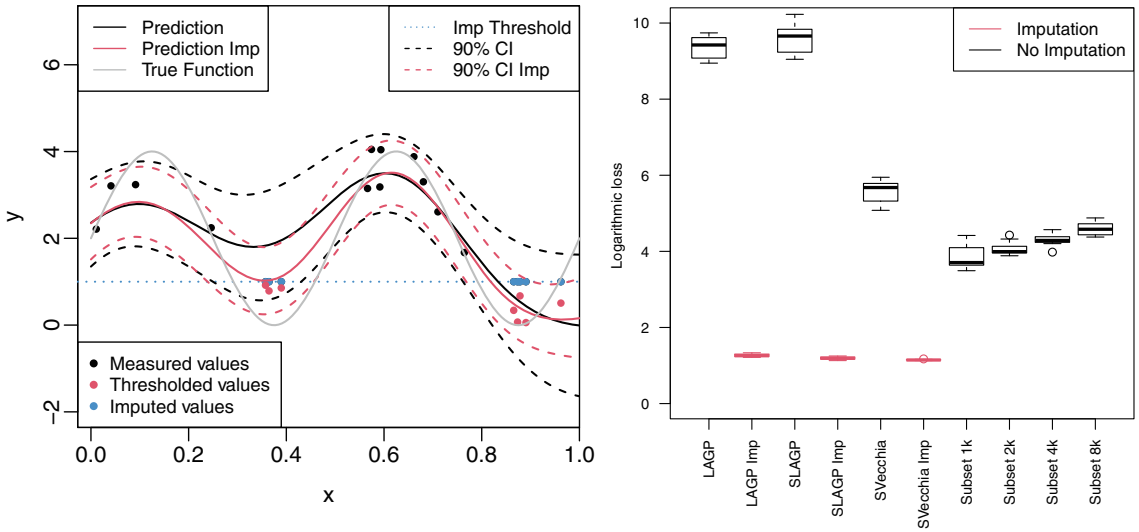


FIGURE 8 Left: Imputation illustration for 1d synthetic data. Right: CV log loss (16) from ore data set two that are below the detection threshold using big data methods with and without imputation.

When working with real data, we do not have the luxury of knowing what the "true curve" is. Consequently, it is much tougher to assess improvement in the quality of fit through imputation. RMSE and score on an out-of-sample testing set, say via CV, are problematic because censored data values—as elements of the testing set—are not realizations from the same population as the model/predictive quantities are ($\mu_n(x)$ and $\sigma_n^2(x)$). The only "truth" we know about these points is that they are measured to be below the threshold. The best we can to is determine if the model accurately predicted the "parity" of its recorded value, either above or below the threshold. The GP framework makes this transition simple. The probability of accurate prediction under a threshold may be obtained through inverse Gaussian CDF evaluated at the threshold. The proper scoring mechanism [21] for such probabilities is the logarithmic loss (LL) [50], also known as cross-entropy loss in the neural network literature. Lower LL is better. When all of the testing data are from one class (less than Y_{cens}), LL boils down to:

$$\begin{aligned}\text{LL}(\mathcal{X}_{\text{cens}}) &= -\frac{1}{N'_{\text{cens}}} \sum_{x \in \mathcal{X}_{\text{cens}}} \log(\mathbb{P}(Y(x) \leq Y_{\text{cens}}) \\ &= -\frac{1}{N'_{\text{cens}}} \sum_{x \in \mathcal{X}_{\text{cens}}} \log\big(\Phi^{-1}(Y_{\text{cens}}; \mu(x), \sigma^2(x))\big).\end{aligned} \tag{16}$$

Returning now to our second ore analysis from Section 4.1, we describe our experience with MI on these data. We only explore (S)LAGP and SVecchia in this context. When dealing with the subset methods, imputation is of limited additional value as completely observed data are plentiful relative to the subset size. It is cumbersome, but not impossible to entertain imputation under OK. Being faithful to the imputation scheme would require human intervention to re-fit variograms after each new imputation is obtained. That would result in over 100 K variogram fits in this example! In this context, we see LAGP as an equivalent, automatic variation on OK that can be more easily entertained when working with censored values and MI.

The right plot of Figure 8 demonstrates that our imputation scheme yields improved LL (16) across the board. For reference, $-\log(0.001) = 6.9$ and $-\log(0.2) = 1.6$, so fits leveraging imputation provide a probability of 0.2 for being below thresholds, on average, compared to 0.001 or worse for the models without imputation. At first, it is hard to square this improvement with what appears to be contrary messaging from the imputation results in Figure 7 (red boxplots). If a practitioner is certain that a particular region is high in gold concentration, a priori, then dropping the censored values (i.e., no imputation), which are all low-measurements, leads to more accurate results. Yet dropping thresholded values exposes the practitioner to confirmation bias: predictions appear more accurate in one part of the space at the expense of massive over-predictions in another. This is what the right panel of Figure 8 shows.

5 | CONCLUSION

We showed that for large data sets, the automated GP modeling approach is at least as accurate as kriging while eliminating much of the human intensive efforts, for example, variography. GPs and kriging produce predictions in a nearly identical manner with the main differences being in hyperparameter estimation: likelihood versus variography. Possibly in small data contexts, expert intervention through variography can lead to (slightly) better fits, so we are not suggesting the human should be eliminated entirely. Identifiability issues [51, 52] inherent in separating signal from noise (i.e., inferring the nugget), and in determining smoothness (p and ν), mean it is always sensible to inspect outputs and challenge downstream inferences against stylized facts about the system. But intimate human involvement challenges reproducibility and limits scope for extension, such as with censored data in our mining context. Thus, we advocate for automatic, likelihood-based approaches for hyperparameters using modern GP methodology.

ACKNOWLEDGMENTS

This work was conducted within the NSF I/UCRC Center for Advanced Subsurface Earth Resource Models (CASERM) which is an industry-university research center jointly managed by Colorado School of Mines and Virginia Tech under the NSF award numbers 1822108 and 1822146, respectively. The authors extend sincerest thanks to Alex Mason Apps at AngloGoldAshanti and Kelly Earle at Skeena Resources for supplying this project with mine assay data. This work was also supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, Scientific Discovery through Advanced Computing (SciDAC) Program under Award Number 0000231018.

DATA AVAILABILITY STATEMENT

Data for the illustrative examples in Section 2 is provided either in-line with our code (see git repo above), or as included with an open source library utilized by our code. The mining data from our industrial partners is proprietary and we do not have permission to share it. However, the data are in a standard CSV-column format, and our code for those examples (in the repo above) may be utilized with any data that can be supplied in that form.

ORCID

Ryan B. Christianson https://orcid.org/0000-0002-8669-0078

Ryan M. Pollyea https://orcid.org/0000-0001-5560-8601

REFERENCES

1. G. Matheron, *The theory of regionalized variables and its applications*, Les Cahiers du Centre de Morphologie Mathematique, Fontaineblue, Paris, 1971.
2. J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Stat. Sci. 4 (1989), no. 4, 409–423.
3. N. Cressie, *Statistics for spatial data*, Wiley Series in Probability and StatisticsWiley, New York, New York, 1993.
4. R. B. Gramacy, *Surrogates: Gaussian process modeling, design and optimization for the applied sciences*, Chapman Hall/CRC, Boca Raton, FL, 2020.
5. M. Katzfuss, J. Guinness, and E. Lawrence. Scaled Vecchia approximation for fast computer-model emulation. 2021.
6. R. B. Gramacy, *laGP: Large-scale spatial modeling via local approximate Gaussian processes in R*, J. Stat. Softw. 72 (2016), 72.
7. Banerjee S. *Geostatistical modeling for environmental processes*, 2017, 81–96.
8. C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, MIT Press, Cambridge, MA, 2006.
9. D. Kalpić and N. Hlupić, *Multivariate normal distributions*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, 907–910.
10. G. Matheron, *Principles of geostatistics*, Econ. Geol. 58 (1963), no. 8, 1246–1266.
11. P. Abrahamsen, *A review of Gaussian random fields and correlation functions*, Norsk Regnesentral/Norwegian Computing Center, Oslo, 1997. https://www.nr.no/directdownload/917_Rapport.pdf.
12. H. Wendland, *Scattered data approximation*, Cambridge University Press, Cambridge, UK, 2004.
13. M. L. Stein, *Interpolation of spatial data: Some theory for kriging*, Springer Science & Business Media, New York, New York, 1999.
14. A. Sauer, R. B. Gramacy, and D. Higdon, *Active learning for deep Gaussian process surrogates*, Technometrics. 65 (2021), no. 1, 4–18.
15. K. Liu, Y. Li, X. Hu, M. Lucu, and W. D. Widanage, *Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries*, IEEE Transact. Industr. Inform. 16 (2020), no. 6, 3767–3777.
16. N. Wycoff, M. Binois, and R. B. Gramacy. Sensitivity Prewarping for local surrogate modeling. *arXiv Preprint arXiv:2101.06296* 2021.
17. R. Gramacy and H. Lian, *Gaussian process single-index models as emulators for computer experiments*, Technometrics 54 (2012), no. 1, 30–41.
18. P. J. Diggle and P. J. Ribeiro, *Model-based Geostatistics*, Springer, New York, New York, 2007.
19. G. Casella and R. Berger, *Statistical inference*, Thomson Learning, Duxbury, 2001.
20. N. Cressie, *Fitting variogram models by weighted least squares*, J. Int. Assoc. Math. Geol. 17 (1985), 563–586.
21. T. Gneiting and A. E. Raftery, *Strictly proper scoring rules, prediction, and estimation*, J. Am. Stat. Assoc. 102 (2007), no. 477, 359–378.
22. M. J. Heaton, A. Datta, A. O. Finley, R. Furrer, J. Guinness, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D. Hammerling, M. Katzfuss, F. Lindgren, D. W. Nychka, F. Sun, and A. Zammit-Mangion, *A case study competition among methods for analyzing large spatial data*, J. Agric. Biol. Environ. Stat. 24 (2019), 398–425.
23. R. B. Gramacy and D. W. Apley, *Local Gaussian process approximation for large computer experiments*, J. Comput. Graph. Stat. 24 (2015), no. 2, 561–578.
24. H. Wackernagel, *Multivariate Geostatistics: An introduction with applications*, Springer, Berlin, 2003.
25. V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag New York, New York, NY, 2013.
26. F. Sun, R. Gramacy, B. Haaland, E. Lawrence, and A. Walker, *Emulating satellite drag from large simulation experiments*, SIAM/ASA J. Uncertain. Quantif. 7 (2019), no. 2, 720–759.
27. M. L. Stein, Z. Chi, and L. J. Welty, *Approximating likelihoods for large spatial data sets*, J. R. Stat. Soc. 66 (2004), no. 2, 275–296.
28. K. Wei, R. Iyer, and J. Bilmes, *Submodularity in data subset selection and active learning*, Proc. Mac. Learn. Res. 37 (2015), 1954–1963.
29. D. MacKay, *Information-based objective functions for active data selection*, Neural Comput. 4 (1992), no. 4, 590–604.
30. Seo S, Wallat M, Graepel T, Obermayer K. *Gaussian process regression: Active data selection and test point rejection*, 2000, 27–34.
31. D. A. Cohn, *Neural network exploration using optimal experiment design*, Neural Netw. 9 (1996), no. 6, 1071–1083.
32. Y. Liu and Y. Hung, *Latin hypercube design-based block bootstrap for computer experiment modeling*, tech. rep., Rutgers, New Brunswick, New Jersey, 2015.
33. A. V. Vecchia, *Estimation and model identification for continuous spatial processes*, J. R. Stat. Soc. Ser. B Methodol. 50 (1988), no. 2, 297–312.
34. A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand, *Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets*, J. Am. Stat. Assoc. 111 (2016), no. 514, 800–812.
35. M. Katzfuss, J. Guinness, W. Gong, and D. Zilber, *Vecchia approximations of Gaussian-process predictions*, J. Agric. Biol. Environ. Stat. 25 (2020), no. 3, 383–414.
36. J. Guinness, *Permutation and grouping methods for sharpening Gaussian process approximations*, Technometrics 60 (2018), no. 4, 415–429.
37. M. Katzfuss and J. Guinness, *A general framework for Vecchia approximations of Gaussian processes*, Stat. Sci. 36 (2021), no. 1, 124–141.
38. J. R. Stroud, M. L. Stein, and S. Lysen, *Bayesian and maximum likelihood estimation for Gaussian processes on an incomplete lattice*, J. Comput. Graph. Stat. 26 (2017), no. 1, 108–120.
39. L. Wu, G. Pleiss, and J. Cunningham. Variational nearest neighbor Gaussian processes. *arXiv Preprint arXiv:2202.01694* 2022.
40. M. R. Osborne, *Fisher's method of scoring*, Int. Stat. Rev. 60 (1992), 271–286.
41. M. Johnson, L. Moore, and D. Ylvisaker, *Minimax and maximin distance designs*, J. Stat. Plann. Infer. 26 (1990), no. 2, 131–148.
42. T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Springer New York Inc., New York, NY, 2001.
43. J. L. Schafer, *Analysis of incomplete multivariate data*, Chapman Hall/CRC, New York, NY, 1997.

44. R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, 2nd ed., Wiley-Interscience, New York, New York, 2002.
45. S. Wilhelm and B. G. Manjunath. *Tmvtnorm: Truncated multivariate Normal and student t distribution*. 2022 R Package Version 1.5.
46. Y. Li and S. K. Ghosh, *Efficient sampling methods for truncated multivariate Normal and student-t distributions subject to linear inequality constraints*, J. Stati. Theory Pract. 9 (2015), no. 4, 712–732.
47. G. Casella, C. P. Robert, and M. T. Wells, *Generalized accept-reject sampling schemes*, Lect. Notes Monogr. Ser. 45 (2004), 342–347.
48. D. B. Rubin, *Multiple imputation for nonresponse in surveys*, Wiley, New York, NY, 1987.
49. D. Reynolds, *Gaussian mixture models*, Springer US, Boston, MA, 2009, 659–663.
50. I. J. Good, *Rational decisions*, J. R. Stat. Soc. Ser. B Methodol. 14 (1952), no. 1, 107–114.
51. W. Tang, L. Zhang, and S. Banerjee, *On identifiability and consistency of the nugget in Gaussian spatial process models*, J. R. Stat. Soc. Ser. B Methodol. 83 (2021), no. 5, 1044–1070.
52. C. G. Kaufman and B. A. Shaby, *The role of the range parameter for estimation and prediction in geostatistics*, Biometrika 100 (2013), no. 2, 473–484.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: R. B. Christianson, R. M. Pollyea, and R. B. Gramacy, *Traditional kriging versus modern Gaussian processes for large-scale mining data*, Stat. Anal. Data Min.: ASA Data Sci. J. (2023), 1–19. https://doi.org/10.1002/sam.11635