UNIVERSITY OF CALIFORNIA
Los Angeles

Extending kriging methods to large datasets with applications to California groundwater data

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Brice Charles Randolph

2017

ABSTRACT OF THE THESIS

Extending kriging methods to large datasets
with applications to California groundwater data

by

Brice Charles Randolph
Master of Science in Statistics
University of California, Los Angeles, 2017
Professor Frederick R Paik Schoenberg, Chair

Spatial interpolation is performed to predict data values of unseen locations based on the distribution of known samples. In the field of geostatistics, the technique of unbiased linear interpolation, known as kriging, is used to predict data at unsampled locations. When working with large data sets or a large domain of interest, standard kriging methods such as ordinary and universal kriging can become computationally slow or require the domain to be partitioned with different models fit to different partitions. In this paper, we review common kriging methods as well as an extension known as fixed rank kriging that circumvents these problems. We apply the method of fixed rank kriging to a dataset of 2016 California groundwater, evaluating prediction accuracy under various model setups. We also compare fixed rank kriging to ordinary and universal kriging based on prediction accuracy and time taken to build the model and make predictions.

The thesis of Brice Charles Randolph is approved.

Hongquan Xu

Nicolas Christou

Frederick R Paik Schoenberg, Committee Chair

University of California, Los Angeles

2017

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Background

Statistical prediction in the spatial setting concerns the prediction of some variable quantity, e.g. a resource, in an unknown location $\mathbf{s}_0$ based on sample values collected at known locations $\mathbf{s}_1, ..., \mathbf{s}_n$. The statistical technique used to solve this spatial prediction problem is formally known as kriging. G. Matheron developed the statistical theory behind the prediction technique in the early 1960s. Matheron popularized the name kriging in honor of a well known South African mining engineer, D. G. Krige. Krige made several contributions to the field of mining, including the development of empirical methods for determining true ore-grade distributions from distributions based on sampled ore grades. Intuitively, the key insight in solving the spatial prediction problem was that observations closer to the prediction point should be given more weight in the predictor. This contrasts with using the arithmetic mean as a predictor, which gives equal weight to all samples, irrespective of proximity.

Although there are several methods of spatial prediction(interpolation) outside of the geostatistical domain such as inverse distance-based weighted interpolation(IDW) or trend surface analysis [BPG13], these methods do not take spatial correlation into account, nor do they give confidence interval estimates.

1.2 Introduction

We will be concerned with modeling data according to the stochastic process

$$\{Z(\mathbf{s}) : \mathbf{s} \in D_s\}$$

where the multivariate data point $Z(\mathbf{s})$ is observed at spatial location $\mathbf{s}$, which varies continuously over D_s (a subset of two-dimensional or three-dimensional space). $\mathbf{s}$ might be measured in latitude and longitude, for instance.

As opposed to the assumption of independence of samples, often employed in statistics, spatial data rarely obey this property. Data that are close together in space are often more alike than those that are far apart.[Cre93] Intuitively, if a neighbor finds gold on his property, you might start searching on your property, expecting more gold to exist in close proximity. This is formalized by the concept of spatial correlation. The quantification of this correlation will help guide the estimation of values at new unobserved locations.

Let $\mathbf{Z} = (Z(\mathbf{s}_1), Z(\mathbf{s}_2), ..., Z(\mathbf{s}_n))'$ be the vector of observed data at known spatial locations $\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n$. The goal is to predict the unobserved value $Z(\mathbf{s}_0)$ at the location $\mathbf{s}_0$.

1.3 Variogram

In geostatistics, spatial correlation is modeled by either the variogram or the covariance function. We can't simply estimate the spatial correlation between two realizations of a variable $z(\mathbf{s})$ at locations $\mathbf{s}_1$ and $\mathbf{s}_2$ as we only have a single data pair. Therefore, we must make some assumptions about the stochastic process that generated the data.

One typical assumption is intrinsic stationarity. Informally, this means that variational properties do not change throughout the region of interest D_s [Cre93]. Intrinsic stationarity is defined by the following two equations.

$$E[Z(\mathbf{s} + \mathbf{h}) - Z(\mathbf{s})] = 0 \tag{1.1}$$

$$Var[Z(\mathbf{s}+\mathbf{h}) - Z(\mathbf{s})] = 2\gamma(h) \tag{1.2}$$

Here, $\mathbf{s}$ and $\mathbf{s}+\mathbf{h}$ are two different locations in D_s. In 1.2, we are stating that the variance of Z is constant and does not depend on the location $\mathbf{s}$, only on separation distance h, a scalar. The quantity $2\gamma(h)$ is known as the variogram and can be estimated as

$$2\hat{\gamma}(h) = \frac{1}{|N(h)|}\sum_{N(h)}(Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2, \tag{1.3}$$

where the sum is over $N(h) = \{(i,j) : ||\mathbf{s}_i - \mathbf{s}_j||_{dist} = h\}$ and $|N(h)|$ is the number of distinct elements in $N(h)$. $\gamma(h)$ is often referred to as the semivariogram. Also, $||||_{dist}$ is the distance measure(usually euclidean or spherical).

The estimator $\hat{\gamma}(h)$ is unbiased, but it is heavily influenced by atypical observations.

A more robust approach to estimating the variogram was proposed in Cressie and Hawkins (1980)[CH80]. They propose the variogram estimator

$$2\bar{\gamma}(h) = \{\frac{1}{|N(h)|}\sum_{N(h)}|Z(\mathbf{s}_i) - Z(\mathbf{s}_j)|^{\frac{1}{2}}\}^4/(0.457 + \frac{0.494}{|N(h)|}). \tag{1.4}$$

Another assumption on the stochastic process $Z(\cdot)$ proceeds from the following two equations.

$$E[Z(\mathbf{s})] = \mu, \text{ for all } \mathbf{s} \in D_s \tag{1.5}$$

$$Cov(Z(\mathbf{s}_1), Z(\mathbf{s}_2)) = C(\mathbf{s}_1 - \mathbf{s}_2), \text{ for all } \mathbf{s}_1, \mathbf{s}_2 \in D_s. \tag{1.6}$$

Equation 1.5 states that the expected value of the random function is μ throughout the region of interest. A stochastic function $Z(\cdot)$ satisfying 1.5 and 1.6 is defined to be second-order stationary. Furthermore, if $C(\mathbf{s}_1 - \mathbf{s}_2)$ is a function of only $||\mathbf{s}_1 - \mathbf{s}_2||$ (the distance between the two points without mention of direction) then $C(\cdot)$ is called isotropic.

Outside of simulated data, we do not know the true variogram, and so we are forced to model it at different lag distances, h. There are many models for the variogram, but only a

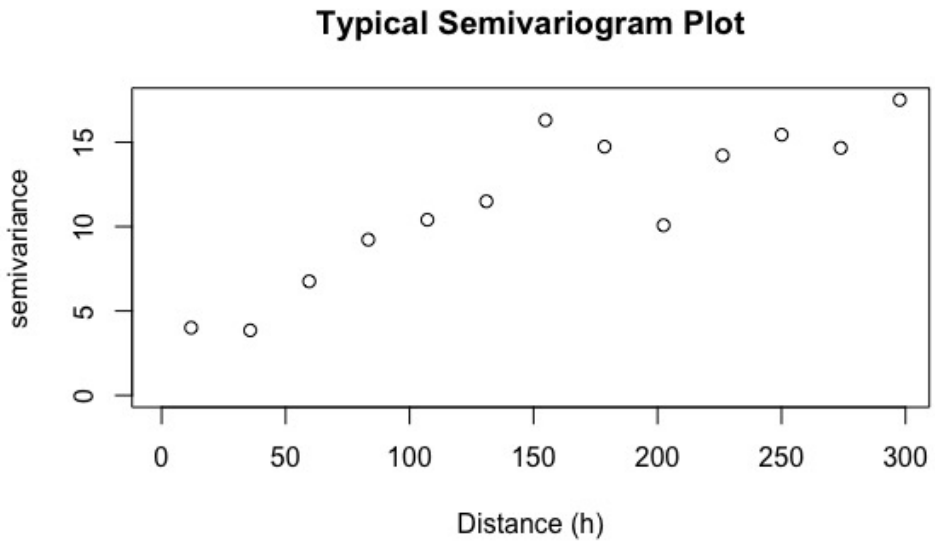

FIGURE 1.1: A typical sample variogram plot

few are typically used in practice. It takes a gross misspecification of the variogram model to have a dramatic impact on kriging estimates. However, the computed kriging variance is directly affected by the variogram fit. (Chiles Chp 3).

1.4 Variogram models

There are a myriad of different models for the variogram. Four of the most common models will be mentioned here. These models assume isotropy of the underlying process.

The simplest is the linear model:

$$\gamma(h;\theta) = \begin{cases} 0 & h = 0, \\ c_0 + bh, & h \neq 0. \end{cases} \tag{1.7}$$

$\theta = (c_0, b)'$, where $c_0 \geq 0$ and $b \geq 0$.

Another common model is the spherical model:

$$\gamma(h;\theta) = \begin{cases} 0 & h = 0, \\ c_0 + c_1\{(3/2)(h/\alpha) - (1/2)(h/\alpha)^3\}, & 0 < h \leq \alpha, \\ c_0 + c_1, & h \geq \alpha, \end{cases} \tag{1.8}$$

$\theta = (c_0, c_1, \alpha)'$, where $c_0 \geq 0$, $c_1 \geq 0$ and $\alpha \geq 0$.

The third model we present is the exponential model. This represents an exponential decay of influence between two samples.

$$\gamma(h;\theta) = \begin{cases} 0 & h = 0, \\ c_0 + c_1(1 - exp(-\frac{h}{\alpha})), & h \neq 0, \end{cases} \tag{1.9}$$

Another common model is the Gaussian semivariogram:

$$\gamma(h;\theta) = \begin{cases} 0 & h = 0, \\ c_0 + c_1(1 - exp(-\frac{h^2}{\alpha^2})), & h \neq 0, \end{cases} \tag{1.10}$$

For both the Gaussian and exponential models, $\theta = (c_0, c_1, \alpha)'$, where $c_0 \geq 0$, $c_1 \geq 0$ and $\alpha \geq 0$.

1.5 Variogram parameters and fitting

Nugget effect (c_0): The nugget effect refers to the nonzero intercept of the variogram. It can be interpreted as sampling error or inherent geological variability.

Range (α): As the separation distance (h) increases, the value of the variogram increases as well. At a certain distance, however, the variogram will plateau. The distance at which the variogram reaches the plateau is known as the range.

Sill ($c_0 + c_1$): The value that the semivariogram model attains at the range.

In practice, we must use the data to first make the choice of a parametric family(e.g.

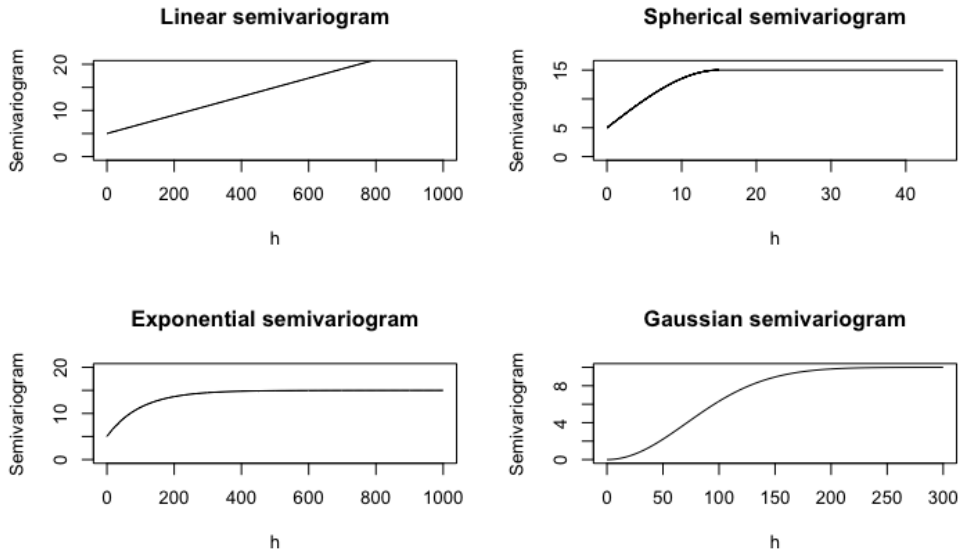

FIGURE 1.2: Basic variogram models

exponential) and then estimate parameters of this model. Conventional practice for model selection is to choose a model that is compatible with a plot of the classical variogram estimator or the robust estimator given in equations 1.3 or 1.4 respectively.

As a compromise between simplicity and statistical efficiency, Noel Cressie proposes using the method of weighted least-squares to fit the model variogram. [Cre85] So, we minimize

$$\sum_{k=1}^{K}\{\frac{\hat{\gamma}(h(k))}{\gamma(h(k);\theta)}-1\}^2|N(h(k))|$$

with respect to the variogram parameters θ(e.g., sill, nugget effect, range, etc.). Here, the sequence $h(1), h(2), ..., h(K)$ denotes the lags at which the classical estimator was computed. We can also replace the classical estimator $\hat{\gamma}$ in the above expression with the robust estimator $\bar{\gamma}$.

1.6 Note on covariance functions

The derivation of the kriging equations can be performed with the use of either the variogram or the covariance function. However, some generalizations of the krigging methods(Fixed

Rank Kriging for example) are derived solely in terms of the covariance function. Therefore, we will briefly describe the link between the covariance function and the variogram.

We define the covariance function

$$C(h) = Cov(Z(\mathbf{s}), Z(\mathbf{s}+\mathbf{h})),\ h \in D_s \tag{1.11}$$

Although both tools serve the same purpose, variograms are more general than covariance functions. However, covariance functions are better understood theoretically and many important properties, characterizations, and decomposition theorems have been established for covariance functions only. [GSS01] (Gneiting 2001) Because of this, some authors prefer to work directly with covariance functions when deriving the kriging equations.

Almost all major properties of covariance functions carry over to variograms. However, one of the exceptions is boundedness. $|C(h)| \leq C(0)$ for covariance function C, yet variograms do not need to be bounded.

If γ is a bounded variogram, then

$$\gamma(h) = C(0) - C(h),\ h \in \mathbb{R}^d \tag{1.12}$$

for some covariance function C.

CHAPTER 2

Kriging

We will now introduce the common methods of spatial prediction in the context of geostatistics. The ordering of the methods presented is based on increasing generality.

2.1 Ordinary kriging

Ordinary kriging refers to spatial prediction under the following two assumptions.

Model assumption:

$$Z(\mathbf{s}) = \mu + \delta(\mathbf{s}), \text{ where } \mathbf{s} \in D_s, \mu \in \mathbb{R}, \text{ and } \mu \text{ unknown.} \tag{2.1}$$

Here, $\delta(\mathbf{s})$ is a zero mean stochastic process with correlated errors. We assume this model holds with variogram $2\gamma(h) = Var(Z(\mathbf{s}+\mathbf{h}) - Z(\mathbf{s}))$, as was discussed in the previous section.

Predictor assumption:

$$\hat{Z}(\mathbf{s}_0) = \sum_{i=1}^{n} w_i Z(\mathbf{s}_i) \ , \ \sum_{i=1}^{n} w_i = 1 \tag{2.2}$$

In 2.1, we are assuming that the predicted value of the process at new location $\mathbf{s}_0$ is a weighted average of the sample values. The second condition ensures that the predictor is unbiased, namely: $\mathbb{E}(Z(\mathbf{s}_0)) = \mu$.

Under assumptions 2.1 and 2.2, ordinary kriging proceeds by minimizing the mean squared prediction error:

$$\sigma_e^2 = \mathbb{E}[Z(\mathbf{s}_0) - \hat{Z}(\mathbf{s}_0)]^2 \tag{2.3}$$

Substituting in our assumed linear predictor from 2.2, we have:

$$\sigma_e^2 = \mathbb{E}[Z(\mathbf{s}_0) - \sum_{i=1}^{n} w_i Z(\mathbf{s}_i)]^2 \tag{2.4}$$

Subject to the constraint $\sum_{i=1}^{n} w_i = 1$.

We can rewrite the mean squared prediction error in terms of the variogram after some algebraic manipulation:

Look at $[Z(\mathbf{s}_0) - \sum_{i=1}^{n} w_i Z(\mathbf{s}_i)]^2$,

$$[z(\mathbf{s}_0) - \sum_{i=1}^{n} w_i z(\mathbf{s}_i)]^2 = z^2(\mathbf{s}_0) - 2z(\mathbf{s}_0)\sum_{i=1}^{n} w_i z(\mathbf{s}_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j z(\mathbf{s}_i) z(\mathbf{s}_j)$$

$$= \sum_{i=1}^{n} w_i z^2(\mathbf{s}_0) - 2\sum_{i=1}^{n} w_i z(\mathbf{s}_0) z(\mathbf{s}_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j z(\mathbf{s}_i) z(\mathbf{s}_j) - \frac{1}{2}\sum_{i=1}^{n} w_i z^2(\mathbf{s}_i)$$

$$-\frac{1}{2}\sum_{j=1}^{n} w_j z^2(\mathbf{s}_j) + \sum_{i=1}^{n} w_i z^2(\mathbf{s}_i) =$$

$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j [z(\mathbf{s}_i) - z(\mathbf{s}_j)]^2 + \sum_{i=1}^{n} w_i [z(\mathbf{s}_0) - z(\mathbf{s}_i)]^2$$

Now we can take expectations of the last expression, giving

$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \mathbb{E}[z(\mathbf{s}_i) - z(\mathbf{s}_j)]^2 + \sum_{i=1}^{n} w_i \mathbb{E}[z(\mathbf{s}_0) - z(\mathbf{s}_i)]^2 =$$

$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j Var[z(\mathbf{s}_i) - z(\mathbf{s}_j)] + \sum_{i=1}^{n} w_i Var[z(\mathbf{s}_0) - z(\mathbf{s}_i)]$$

The last equality proceeds under the intrinsic stationarity assumption, where $\mathbb{E}[Z(\mathbf{s})] = \mu$. This enables us to write $Var[z(\mathbf{s}+\mathbf{h}) - z(\mathbf{s})] = \mathbb{E}[z(\mathbf{s}+\mathbf{h}) - z(\mathbf{s})]^2$.

Now, with the variogram defined as $2\gamma(\mathbf{s}_i - \mathbf{s}_j) = Var[z(\mathbf{s}_i) - z(\mathbf{s}_j)]$, we can write the

previous expression as: $2\sum_{i=1}^{n} w_i\gamma(\mathbf{s}_0 - \mathbf{s}_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \gamma(\mathbf{s}_i - \mathbf{s}_j)$.

So, the kriging equations proceed from minimizing the mean squared prediction error σ_e^2.

$$\sigma_e^2 = \mathbb{E}[Z(\mathbf{s}_0) - \sum_{i=1}^{n} w_i Z(\mathbf{s}_i)]^2 =$$

$$2\sum_{i=1}^{n} w_i\gamma(\mathbf{s}_0 - \mathbf{s}_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \gamma(\mathbf{s}_i - \mathbf{s}_j)$$

$$\text{subject to } \sum_{i=1}^{n} w_i = 1$$

Using the method of Lagrange multipliers, we can encode the constraint in the objective function:

$$\ell(w_1, ..., w_n, \lambda) = 2\sum_{i=1}^{n} w_i\gamma(\mathbf{s}_0 - \mathbf{s}_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \gamma(\mathbf{s}_i - \mathbf{s}_j) - 2\lambda(\sum_{i=1}^{n} w_i - 1) \tag{2.5}$$

Now, minimization of $\ell(w_1, ..., w_n, \lambda)$ with respect to the parameters $w_1, ..., w_n$, and λ will minimize equation 2.4 under the constraint. After differentiating $\ell(w_1, ..., w_n, \lambda)$ with respect to the $(n+1)$ parameters and setting these derivatives equal to zero, we get:

$$-\sum_{j=1}^{n} w_j\gamma(\mathbf{s}_i - \mathbf{s}_j) + \gamma(\mathbf{s}_0 - \mathbf{s}_i) - \lambda = 0 \text{ , i = 1,...,n} \tag{2.6}$$

$$\text{and } \sum_{i=1}^{n} w_i = 1$$

This system of equations can be written using matrix notation as

$$\Gamma\mathbf{w} = \gamma$$

Thus, the weights $w_1, ..., w_n$ and Lagrange multiplier λ can be obtained as

$$\mathbf{w} = \Gamma^{-1}\gamma \tag{2.7}$$

where

$$\mathbf{w} = (w_1, w_2, ..., w_n, \lambda)$$

$$\gamma = (\gamma(\mathbf{s}_0 - \mathbf{s}_1), \gamma(\mathbf{s}_0 - \mathbf{s}_2), ..., \gamma(\mathbf{s}_0 - \mathbf{s}_n), 1)'$$

$$\Gamma = \begin{cases} \gamma(\mathbf{s}_i - \mathbf{s}_j), & i = 1, 2, ..., n, j = 1, 2, ..., n \\ 1, & i = n+1, j = 1, 2, ..., n, \\ 1, & j = n+1, i = 1, 2, ..., n, \\ 0, & i = n+1, j = n+1. \end{cases}$$

Now that we have calculated the weight vector $\mathbf{w}$, we can compute the estimator $\hat{Z}(\mathbf{s}_0) = \sum_{i=1}^{n} w_i Z(\mathbf{s}_i)$.

If we would like to construct confidence intervals around our predictor, we will need the variance of our estimator, σ_e^2. We can derive this by starting with equation 2.6

$$-\sum_{j=1}^{n} w_j \gamma(\mathbf{s}_i - \mathbf{s}_j) + \gamma(\mathbf{s}_0 - \mathbf{s}_i) - \lambda = 0$$

We multiply this by w_i and sum over $i = 1, 2, ..., n$ to get:

$$-\sum_{i=1}^{n} w_i \sum_{j=1}^{n} w_j \gamma(\mathbf{s}_i - \mathbf{s}_j) + \sum_{i=1}^{n} w_i \gamma(\mathbf{s}_0 - \mathbf{s}_i) - \sum_{i=1}^{n} w_i \lambda = 0$$

This gives

$$\sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \gamma(\mathbf{s}_i - \mathbf{s}_j) = \sum_{i=1}^{n} w_i \gamma(\mathbf{s}_0 - \mathbf{s}_i) - \sum_{i=1}^{n} w_i \lambda$$

Now, we can substitute this result into the equation for the variance:

$$\sigma_e^2 = 2\sum_{i=1}^{n} w_i\gamma(\mathbf{s}_0 - \mathbf{s}_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \gamma(\mathbf{s}_i - \mathbf{s}_j)$$

$$\sigma_e^2 = 2\sum_{i=1}^{n} w_i\gamma(\mathbf{s}_0 - \mathbf{s}_i) - (\sum_{i=1}^{n} w_i\gamma(\mathbf{s}_0 - \mathbf{s}_i) - \sum_{i=1}^{n} w_i\lambda)$$

Finally, we arrive at the the variance of our estimator in terms of the computed weights and Lagrange multiplier:

$$\sigma_e^2 = \sum_{i=1}^{n} w_i\gamma(\mathbf{s}_0 - \mathbf{s}_i) + \lambda \tag{2.8}$$

Equation 2.7, $\mathbf{w} = \Gamma^{-1}\gamma$, illustrates the need for us to choose a model variogram that captures the underlying stochastic process. Firstly, the vector $\gamma = (\gamma(\mathbf{s}_0 - \mathbf{s}_1), \gamma(\mathbf{s}_0 - \mathbf{s}_2), ..., \gamma(\mathbf{s}_0 - \mathbf{s}_n), 1)'$ might not be computable from the sample variogram estimates. This is because we only calculated a number of sample variogram estimates at certain lag distances (h). Secondly, solving the kriging equations (2.6) requires the computation of the inverse of a size $(n+1)x(n+1)$ matrix. This inverse is not guaranteed to exist using only the sample variogram since the sample version of Γ might not be positive definite. In order to get around these two problems, we select and fit an appropriate model variogram(covered in section 1.4) and proceed using the theoretical variogram values.

Now, we can summarize the kriging process. First we calculate the sample variogram under different lag distances h. We plot these sample values against the different lag distances and choose an appropriate class of theoretical variograms. We fit the theoretical variogram to the data by minimizing a weighted sum of squares as in section 1.5. This will give us our matrix Γ. Finally, we calculate the weight vector $\mathbf{w}$ by inverting the matrix Γ. For each location, $\mathbf{s}_0$, we predict the value of the underlying random process $Z(\mathbf{s}_0)$ using our predictor $\hat{Z}(\mathbf{s}_0) = \sum_{i=1}^{n} w_i Z(\mathbf{s}_i)$.

Usually, a raster map of the predicted values is created along with a second map showing the standard errors σ_e of the estimate. With this in mind, we will have a grid of values to predict. Luckily, we only need to calculate a new vector γ as we will use the same matrix Γ^{-1} for each new location.

2.2 Universal kriging

The goal of universal kriging is the same as ordinary kriging, namely, predict the value $Z(\mathbf{s}_0)$ of the stochastic process at an unsampled location $\mathbf{s}_0$. Ordinary kriging assumes stationarity, a constant mean of the underlying random function $Z(\cdot)$. In reality, the mean values may vary over the study area. Spatial trend or a drift represents any detectable tendency for the values to change as a function of the coordinate variables.

Therefore, if our goal is to krig unsampled locations under the presence of a trend, equation 2.1 is not appropriate and must be generalized. Universal kriging assumes that the mean $\mu(\cdot)$ has a functional dependence on the spatial location $\mathbf{s}$, ie. $\mu(\mathbf{s})$ may vary over the region of interest. We model this functional dependence with an unknown linear combination of known functions $\{f_0(\mathbf{s}), ..., f_p(\mathbf{s})\}$, $\mathbf{s} \in D_s$. Our model for the underlying stochastic process then becomes:

$$Z(\mathbf{s}) = \sum_{j=1}^{p+1} f_{j-1}(\mathbf{s})\beta_{j-1} + \delta(\mathbf{s}) \tag{2.9}$$

Here, $\beta = (\beta_0, ..., \beta_p)' \in \mathbb{R}^{p+1}$ is an unknown vector of parameters and $\delta(\cdot)$ is a zero-mean intrinsically stationary random process with variogram $2\gamma(\cdot)$.

We might model the trend as a polynomial function of the X and Y coordinates(in the case that $\mathbf{s} \in \mathbb{R}^2$) as follows:

$$Z(\mathbf{s}_i) = \beta_0 + \beta_1 X_i + \beta_2 Y_i + \beta_3 X_i^2 + \beta_4 X_i Y_i + \beta_5 Y_i^2 + \delta(\mathbf{s}_i).$$

However, we should avoid overfitting the trend and thus should restrict ourselves to low-order polynomials.

Regardless of the proposed trend, we will still have the same predictor assumptions of equation 2.2. However, we must also incorporate more constraints to ensure that our

estimator is unbiased. In the case of a linear trend, we have:

$$Z(\mathbf{s}_0) = \beta_0 + \beta_1 X_0 + \beta_2 Y_0 + \delta(\mathbf{s}_0)$$

We must therefore add the constraints $\sum_{i=1}^n w_i X_i = X_0$ and $\sum_{i=1}^n w_i Y_i = Y_0$ to our original constraint $\sum_{i=1}^n w_i = 1$. In the same fashion as ordinary kriging, we minimize the mean squared prediction error under these constraints in order to derive our kriging weights. We will use the notation $\mathbf{x} = (f_0(\mathbf{s}_0), ..., f_p(\mathbf{s}_0))'$ and X is an n x $(p+1)$ matrix whose (i,j)th element is $f_{j-1}(\mathbf{s}_i)$. Our weight vector

$$\mathbf{w}_u = \Gamma_u^{-1} \gamma_u,$$

where

$$\mathbf{w}_u = (w_1, ..., w_n, \lambda_0, ..., \lambda_p)',$$

$$\gamma_u = (\gamma(\mathbf{s}_0 - \mathbf{s}_1), ..., \gamma(\mathbf{s}_0 - \mathbf{s}_n), 1, f_1(\mathbf{s}_0), ..., f_p(\mathbf{s}_0))',$$

and

$$\Gamma = \begin{cases} \gamma(\mathbf{s}_i - \mathbf{s}_j), & i = 1, 2, ..., n, j = 1, 2, ..., n \\ 1, & i = n+1, j = 1, 2, ..., n, \\ 1, & j = n+1, i = 1, 2, ..., n, \\ 0, & i = n+1, j = n+1. \end{cases}$$

For universal kriging, we have assumed that the variogram is known. However, in practice, it is estimated using $2\hat{\gamma}$ or $2\bar{\gamma}$. This is inappropriate in the presence of a trend since

$$\mathbb{E}(Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2 = var(Z(\mathbf{s}_i) - Z(\mathbf{s}_j)) + \{\mu(\mathbf{s}_i) - \mu(\mathbf{s}_j)\}^2$$

where the first term on the right hand side is $2\gamma(\mathbf{s}_i - \mathbf{s}_j)$ and the second term is nonzero. If we knew the vector β characterizing out trend, we could estimate the variogram based on

the residuals. However, β is unknown and in order to estimate it (using generalized least squares), we need knowledge of $var(\mathbf{Z}) = \Sigma$, the covariance matrix of our data. But, Σ is unknown and we are back where we started. This circularity has lead to some dissatisfaction with universal kriging as we are introducing bias when we estimate the variogram using the residuals. However, the bias of the residuals-based variogram is small at lags near the origin and more substantial at distant lags. Provided that a variogram model is fit by weighted least squares, which puts most weight on the estimator at small lags, the effect of the bias should be small. (Cressie 1993 section 3.4.3) This bias does not have much influence on the universal kriging predictor, but the kriging variance may be smaller than it should be.

2.3 Kriging as a regression procedure

A. Stein and L. C. A. Corsten embed the kriging technique into regression procedures, leading to a more straightforward formulation than using Lagrange multipliers.[SC91]

Let $\mathbf{C}$ be a positive-definite matrix such that each element is assumed to be an isotropic function $c(h)$ only of the distance (h) between the pair of observation points concerned, ie. $\mathbf{C}$ is the covariance matrix that will take the place of the model variogram matrix Γ. We have the following linear model for the actual observations $\mathbf{y}$ and hypothetical observation y_0:

$$(\mathbf{y}\ y_0)' = (X\ \mathbf{x}_0)'\beta + \epsilon$$

where $\mathbb{E}(\epsilon) = 0$ and $Cov(\epsilon) = \mathbf{C}$. We require the best predictor $\hat{y} = \mathbf{w}'\mathbf{y}$, linear in the observations $\mathbf{y}$, to satisfy $\mathbb{E}(\mathbf{w}'\mathbf{y} - y_0) = 0$ and $Var(\mathbf{w}'\mathbf{y} - y_0)$ is minimal. Again, $\mathbf{C}$ and $\mathbf{c}_0$ are the known covariances among the y_i and those between the y_i and y_0. We note that the vector $y_0 - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y}$ is orthogonal to any linear combination of $y_1, ..., y_n$ and $Var(y_0 - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y}) = Var(y_0) - Var(\mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y}) = c_{00} - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{c}_0$, c_{00} being the variance of y_0.

Now, we split the prediction error into

$$\hat{y} - y_0 = (\mathbf{w}'\mathbf{y} - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y}) - (y_0 - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y}),$$

the difference between two orthogonal terms, and $Var(\hat{y} - y_0) = Var(\mathbf{w}'\mathbf{y} - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y}) + c_{00} - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{c}_0$.

Furthermore, for $\nu'\mathbf{y}$, with ν' equal to $\mathbf{w}' - \mathbf{c}_0'\mathbf{C}^{-1}$, we require that $\mathbb{E}(\nu'\mathbf{y}) = \mathbf{x}_a'\beta$ with $\mathbf{x}_a' = (\mathbf{x}_0' - \mathbf{c}_0'\mathbf{C}^{-1}X)$ to satisfy the unbiasedness requirement. Minimizing $Var(\hat{y} - y_0)$ is equivalent to minimizing $Var(\nu'\mathbf{y})$ while $\mathbb{E}(\nu'\mathbf{y}) = \mathbf{x}_a'\beta$. The minimum is attained by the GLS estimator $\hat{\beta} = \mathbf{V}X'\mathbf{C}^{-1}\mathbf{y}$, where $\mathbf{V} = (X'C^{-1}X)^{-1}$. Therefore, $\nu'\mathbf{y} = \mathbf{x}_a'\hat{\beta}$ and $\hat{y} = \nu'\mathbf{y} + \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y}$ will be:

$$\hat{y}(\mathbf{s}_0) = \mathbf{x}_a'\hat{\beta} + \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{y} = \mathbf{x}_0'\hat{\beta} + \mathbf{c}_0'\mathbf{C}^{-1}(\mathbf{y} - X\hat{\beta}) \tag{2.10}$$

The variance of our estimator will then be $Var(\hat{y} - y_0) = c_{00} - \mathbf{c}_0'\mathbf{C}^{-1}\mathbf{c}_0 + \mathbf{x}_a'\mathbf{V}\mathbf{x}_a$.

From equation 2.10, we can see that the predictor $\hat{y}$ is the sum of the estimated local expectation of the target value ($\mathbf{x}_0'\hat{\beta}$) and a linear combination of the observed residuals within $\mathbf{y} - X\hat{\beta}$, with the coefficients given by the best linear approximation of the target value by all n observations. Furthermore, since $\hat{y} - \mathbf{x}_0'\hat{\beta} = \mathbf{c}_0'\mathbf{C}^{-1}(\mathbf{y} - X\hat{\beta})$, the procedure can be interpreted as regression of the residuals of $\hat{y}$ with respect to $\mathbf{x}_0'\hat{\beta}$ on the residuals of $\mathbf{y}$ with respect to $X\hat{\beta}$.

2.4 Problems with kriging on large datasets

When fitting the ordinary and universal kriging equations to a dataset, the covariance structure is assumed to be the same over the whole domain. In cases where the domain of interest is large, the state of California, for example, this may be an inappropriate assumption. If we wish to model different covariance structures in different parts of the domain, we are forced to perform ad-hoc methods of partitioning the domain and fitting a different model for each partition. This brings up issues of discontinuity in the covariance function as well as a burden on the modeler. Ordinary and universal kriging require the inversion of an n x n covariance matrix (or variogram), with a computational cost on the order of n^3. This isn't a problem for data sets under about 5,000 rows, but can lead to major slow downs for

larger data sets.

2.5 Fixed rank kriging

When datasets are large, they are often defined over a large spatial domain, so the spatial process of interest usually exhibits non-stationary behavior over that domain. Fixed rank kriging (FRK) is universal kriging within a class of non-stationary covariance functions. The family of non-stationary covariance functions is defined using a set of basis functions that is fixed in number. FRK is written in terms of the covariance matrix Σ. Regardless of whether the kriging equations are written in terms of the variogram Γ or the covariance matrix Σ, we still need to invert these matrices to obtain our kriging predictor. In FRK, a class of n x n covariance matrices is selected such that Σ^{-1} can be obtained by inverting r x r matrices, where r is fixed and $r << n$.

We continue to model the underlying stochastic process as

$$Z(s_i) = Y(s_i) + \epsilon(s_i) \ , \ i = 1, ..., n,$$

We assume that the measurement-error process, $\epsilon(\cdot)$, is statistically independent of $Y(\cdot)$ and distributed as,

$$\epsilon(\cdot) \sim N(0, \sigma_\epsilon^2 \upsilon_\epsilon(\cdot))$$

where $\upsilon_\epsilon(\cdot)$ is a function that is usually assumed known. Then, $\mathbf{Z}|\mathbf{Y} \sim N_n(\mathbf{Y}, \sigma_\epsilon^2 V_\epsilon)$ where $V_\epsilon = \text{diag}(\upsilon_\epsilon(s_1), ..., \upsilon_\epsilon(s_n))$. The process $Y(\cdot)$ is decomposed into two components,

$$Y(\cdot) = \mu(\cdot) + \nu(\cdot)$$

Here, the first component $\mu(\cdot)$ is a deterministic large-scale trend, and the second component $\nu(\cdot)$ is a random spatial-variation component. As in universal kriging, we assume that the

deterministic trend is a linear function of spatial covariates,

$$\mu(\cdot) = \mathbf{x}(\cdot)'\beta$$

where $\mathbf{x}(\cdot)$ is a p-dimensional vector of known covariates. The random spatial-variation term $\nu(\cdot)$ is further assumed to follow the spatial-random-effects (SRE) model,

$$\nu(\cdot) = \mathbf{S}(\cdot)'\eta + \xi(\cdot),$$

where $\mathbf{S}(\cdot) = (S_1(\cdot), ..., S_r(\cdot))'$ is an r-dimensional ($r << n$) vector of spatial basis functions, and η is a spatial random-effects vector. Sometimes, $\mathbf{S}(\cdot)$ contains basis functions of multiple resolutions, with the goal of capturing many scales of spatial variation. (Cressie and Johannesson 2008) We also assume that η is distributed as

$$\eta \sim N_r(0, K),$$

where K is an unknown r x r symmetric positive-definite matrix. The fine-scale-variation process $\xi(\cdot)$ accounts for the error induced by dimension reduction and is distributed as,

$$\xi(\mathbf{s}) \sim N(0, \sigma_\xi^2 \upsilon_\xi(\mathbf{s})),$$

independently for all $\mathbf{s} \in D_s$, and independently of η, where $\upsilon_\xi(\cdot)$ is a known function. We can then write the data vector as,

$$\mathbf{Z} = \mathbf{Y} + \epsilon,$$

where

$$\mathbf{Y} = X\beta + S\eta + \delta$$

Here, the i-th row of S is $\mathbf{S}(\mathbf{s}_i)'$. The covariance matrix of the data vector has the form,

$$\Sigma = var(\mathbf{Z}) = var(S\eta) + var(\delta + \epsilon) = SKS' + D,$$

where $D = \sigma_\xi^2 V_\xi + \sigma_\epsilon^2 V_\epsilon$. Here, $V_\xi = diag(\upsilon_\xi(\mathbf{s}_1), ..., \upsilon_\xi(\mathbf{s}_n))$ and $V_\epsilon = diag(\upsilon_\epsilon(\mathbf{s}_1), ..., \upsilon_\epsilon(\mathbf{s}_n))$. If there is no reason to believe that the measurement-error variances should be different in different parts of D_s, we can assume $V_\epsilon = I_n$, the identity matrix. The same applies to $\upsilon_\xi(\cdot)$.

In the FRK framework, the domain of interest is discretized into a set of N_d non-overlapping tiles known as Basic Aerial Units or BAUs. The process $\{\mathbf{Y}(\mathbf{s}) : \mathbf{s} \in D_s\}$ is then averaged over the BAUs. When we use the model to predict at unobserved locations, our prediction will give one value for the BAU that overlays the location of interest.

We must also specify the type, the number (r), and the locations of the basis functions used in the matrix S. Common basis functions include bisquare functions, wavelets, indicator functions, Gaussian functions, Matérn functions, and exponential functions.

Before prediction can proceed, we must estimate the unknown parameters $\{\beta, K, \sigma_\xi^2\}$ and the measurement-error variance σ_ϵ^2, which is assumed known. The measurement-error variance can be estimated by interpolating variogram estimates at lags (h) closest to zero. (Cressie 1993 section 3.2.1) It is also sometimes assumed to be zero, but this is a matter of choice and often depends on the application. In implementations of FRK, namely the R package "FRK", the user has the ability to attribute fine-scale variation to the observation model or the process model. [Zam17] In the above exposition, we consider estimation where the fine-scale variation sits in the process model.

One option for parameter estimation is through maximum likelihood estimation via the expectation-maximization (EM) algorithm as in [KC11]. The EM algorithm attempts to find the value of the parameter vector that maximizes the likelihood function, defined as the probability density function of the observed data as a function of the unknown parameters. We can estimate the parameter β through ordinary or weighted least squares, then use the EM algorithm to estimate $\{K, \sigma_\xi^2\}$. Alternatively, we could use EM to estimate all three parameters simultaneously, however, we will pursue the former method in this case.

The EM algorithm begins with starting values, $K^{[0]}$ and $\sigma_\xi^{2[0]}$, for the two parameters of interest. We subsequently update both parameters iteratively for $t = 1, 2, ...$ until conver-

gence.

$$K^{[t+1]} = K^{[t]} - K^{[t]}S'\Sigma^{[t]-1}SK^{[t]} + (K^{[t]}S'\Sigma^{[t]-1}\tilde{\mathbf{Z}})(K^{[t]}S'\Sigma^{[t]-1}\tilde{\mathbf{Z}})'$$

$$\sigma_\xi^{2[t+1]} = \sigma_\xi^{2[t]} + (\sigma_\xi^{2[t]})^2 tr(\Sigma^{[t]-1}[\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}'\Sigma^{[t]-1} - I_n]V_\xi/n)$$

where $\Sigma^{[t]} = SK^{[t]}S' + \sigma_\xi^{2[t]}V_\xi + \sigma_\epsilon^2 V_\epsilon$, $\tilde{\mathbf{Z}} = \mathbf{Y} - X\hat{\beta}$, and $\Sigma^{[t]-1}$ is shorthand for $(\Sigma^{[t]})^{-1}$. Using the Sherman-Morrison-Woodbury formula, we can invert $\Sigma^{[t]}$ by inverting only the r x r matrix $K^{[t]}$ and the diagonal matrix $D^{[t]} = \sigma_\xi^{2[t]}V_\xi + \sigma_\epsilon^2 V_\epsilon$:

$$\Sigma^{[t]-1} = D^{[t]-1} - D^{[t]-1}S[K^{[t]-1} + S'D^{[t]-1}S]^{-1}S'D^{[t]-1}.$$

This reduces the order of computations from n^3 to nr^2, and therefore makes computations scalable to larger datasets.

From equation 2.10, we can then write the FRK predictor as:

$$\hat{y}(\mathbf{s}_0) = \mathbf{x}_0'\hat{\beta} + \mathbf{S}(\mathbf{s}_0)'KS'\Sigma^{-1}(\mathbf{y} - X\hat{\beta})$$

and the kriging variance as:

$$Var(\hat{y} - y_0) = \mathbf{S}(\mathbf{s}_0)'K\mathbf{S}(\mathbf{s}_0) - \mathbf{S}(\mathbf{s}_0)'KS'\Sigma^{-1}SK\mathbf{S}(\mathbf{s}_0) + \mathbf{p}'(X'\Sigma^{-1}X)^{-1}\mathbf{p}$$

where $\mathbf{p} = \mathbf{x}_0 - X'\Sigma^{-1}SK\mathbf{S}(\mathbf{s}_0)$.

CHAPTER 3

Kriging of California groundwater data

3.1 Background

California's groundwater provides approximately 30 to 46 percent of the State's total water supply, depending on wet and dry years.[CAg] As the demand of water for domestic, agricultural, and industrial uses increases, groundwater management becomes increasingly important. Various management measures need to know the spatial and temporal behavior of groundwater. Accurate maps of depth to groundwater are necessary for predicting net flow direction as well as monitoring groundwater recharge. In a scattered groundwater observation region such as California, geostatistical methods can be used to determine the values for the points where measurements are not made. In what follows, several different kriging methods were applied to a dataset of 2016 depth to groundwater measurements in California. Specifically, we wish to evaluate Fixed rank kriging as it compares to ordinary and universal kriging in terms of prediction accuracy and computation speed.

The original data consists of 4,727 measurement of depth to groundwater below the ground surface. These measurements span California, but have a nonuniform coverage of the state.

The measurements were taken in the Fall of 2016, between October and December. Of the 4,727 original measurements, 4,028 measurements with unique locations were kept. Furthermore, 42 locations with a negative depth to groundwater measurement were excluded from the kriging as they indicate that the groundwater level is above ground surface at those locations.

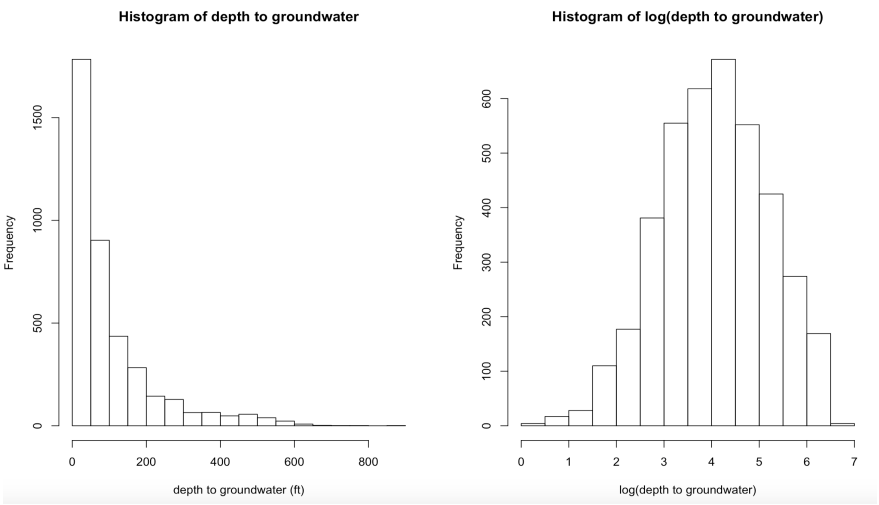

FIGURE 3.1: Histograms of original and transformed values

3.2 Methods

We wish to perform kriging on the groundwater data using the fixed rank kriging methodology. We will be using the R package "FRK" [Zam17] to perform the kriging as well as the package "gstat" [GPH16] when comparing FRK to ordinary and universal kriging. In comparing the different kriging methods, we will look at out-of-sample prediction accuracy based on the mean squared error. We will also compare standard error maps and the time it takes to fit the models and make predictions for each method.

We decided to work with the logarithm of the depth since this transformation gave values that more closely obeyed a normal distribution (see figure 3.1). As we have a moderately large dataset, some departure from normality is expected. Regardless of normality, fixed rank kriging is still justified as a spatial Best Linear Unbiased Predictor (BLUP).

Looking at figure 3.2, we can see what appears to be a latitude gradient in the data, as values seem to be decreasing as we move from southern California to northern California. Maps were created using the ggmap package in R.[KW13]

Recall that we first wish to characterize the deterministic trend $\mathbf{x}(\cdot)'\beta$, which models large-scale variation. We fit a linear model with latitude and longitude as a covariates (including an intercept), using least squares. As longitude was not a significant predictor

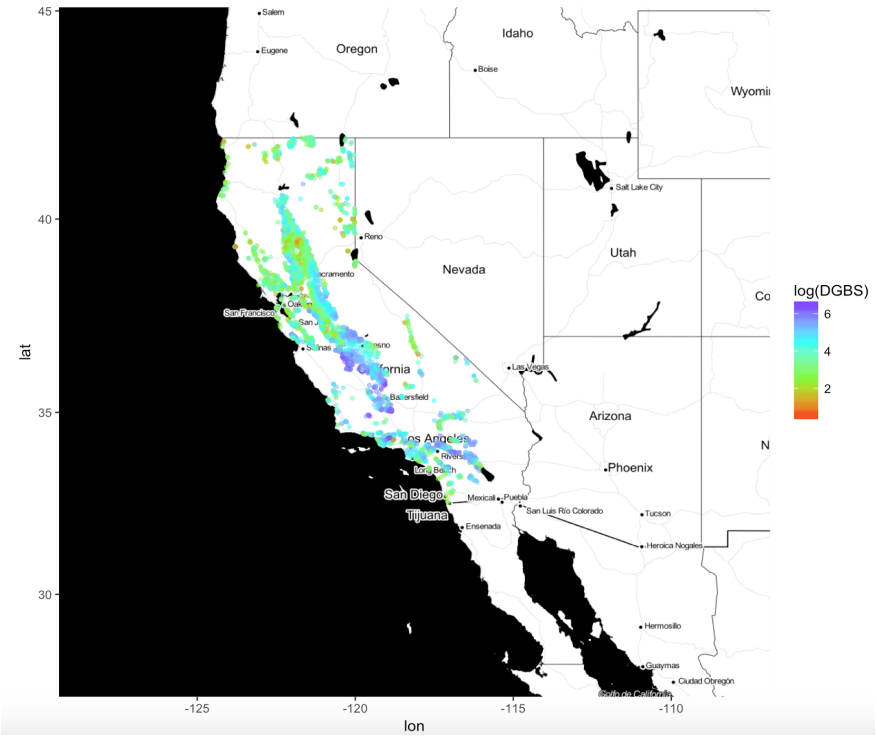

FIGURE 3.2: California log(depth to groundwater [ft])

in the model, we decided to model the trend with latitude and an intercept term. This gives a parameter vector of $\beta = (11.591 \quad -0.200)'$. Thus, in the notation of our model, $\mathbf{x} = (1 \text{ latitude})'$.

We then check the detrended data for normality (figure 3.3). Again, there is some departure from normality, but we will proceed with the analysis.

To model small-scale variation ($S\eta$ in the SRE model), we construct the basis matrix S. As Cressie and Johannesson recommend in [CJ08], we use multiple resolutions for the basis functions. We compare prediction accuracy of models constructed with Gaussian, bisquare, exponential, and Matérn covariance functions.

The form of these functions is as follows:

$$\phi_{Gaussian}(u) = \exp\left(-\frac{\|u\|^2}{2\sigma^2}\right)$$

$$\phi_{bisquare}(u) = \{1 - (\frac{\|u\|}{R})^2\}^2 I(\|u\| < R),$$

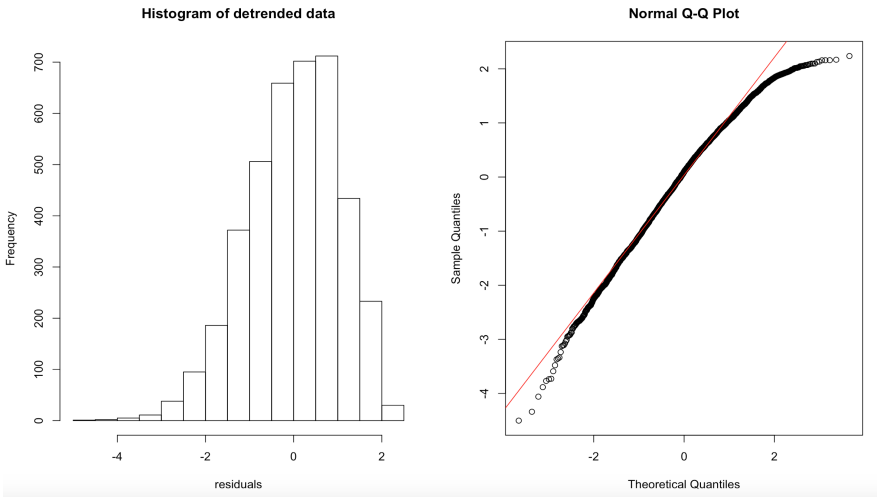

FIGURE 3.3: Histogram of detrended data with QQ plot

$$\phi_{exp}(u) = \exp\left(-\frac{\|u\|}{\tau}\right),$$

$$\phi_{Matern}(u) = \left(1 + \frac{\sqrt{3}\|u\|}{\kappa}\right)\exp\left(-\frac{\sqrt{3}\|u\|}{\kappa}\right)$$

where $\{\sigma, R, \tau, \kappa\}$ are scale coefficients set based on the minimum distance between the centroid locations following placement. u is the distance between the data point and the centroid of the basis function.

We used the functions FRK::auto_basis() and INLA::inla.nonconvex.hull() to place the basis functions on our domain D_s, the interior of California. Figure 3.4 shows the placement of the basis functions. For our comparisons, we used constructed the basis matrix with two different resolutions (plot b). In an attempt to increase prediction accuracy, we also tried using three different resolutions (plot a). 82 basis functions were used for the two resolution basis and 468 for the three resolution basis.

The convex hull of our data was discretized into BAUs using the function FRK::auto_BAUs(). Each BAU covered 0.2 by 0.2 degrees latitude/longitude, giving 1,406 units (Grid 1). We also used a finer grid with a 0.1 by 0.1 degrees area, giving 5,596 units (Grid 2). We found that making the grid any finer lead to a decrease in prediction accuracy. σ^2_ϵ was estimated to be 0.189 for the FRK model.

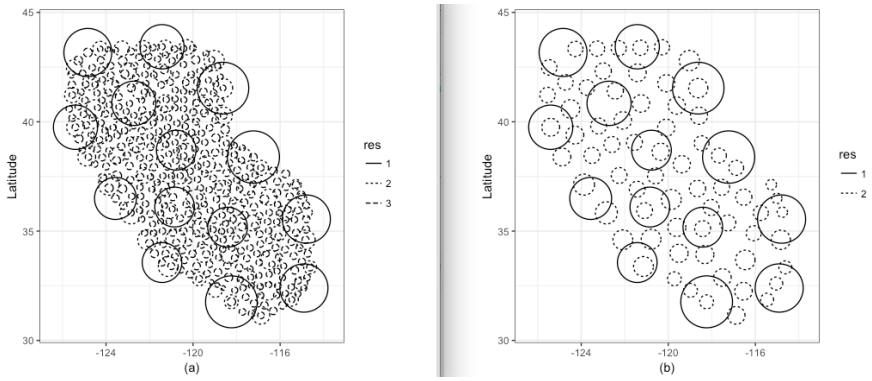

FIGURE 3.4: Constructed basis functions

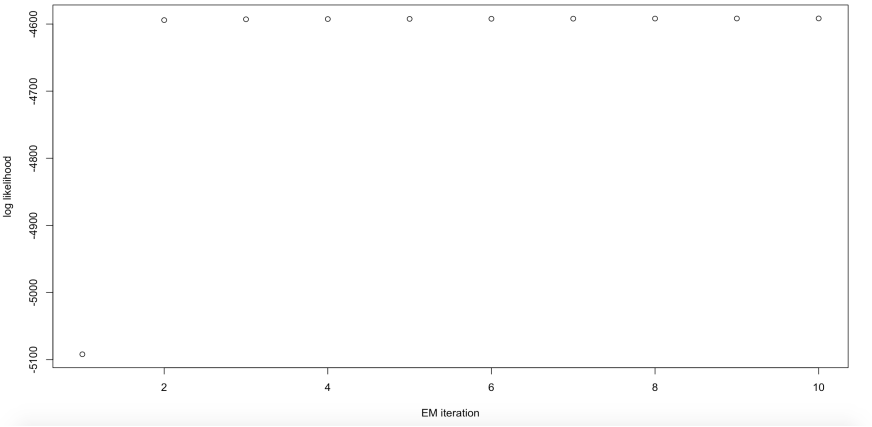

FIGURE 3.5: Convergence of the EM algorithm

We performed parameter estimation using the EM algorithm with a convergence tolerance of 0.1. This turned out to be the major bottleneck for fixed rank kriging. Time to fit the model and make predictions varied depending on the choice of number and type of basis functions, as well as the resolution of the grid of BAUs. Surprisingly, the algorithm converged much faster when using the finer grid. So, despite having more prediction locations, 5,596 vs. 1,406, the time taken to fit the model and make predictions decreased.

The universal kriging used the same trend coefficients as the FRK model, with the spherical variogram model fit in figure 3.6 (distance is in kilometers). The variogram had the following parameters, (Nugget c_0 , Sill (c_0+c_1), Range α)=(0.336,0.7854,100.22). Based on the directional variograms displayed in figure 3.7, we believe that the assumption of geometric

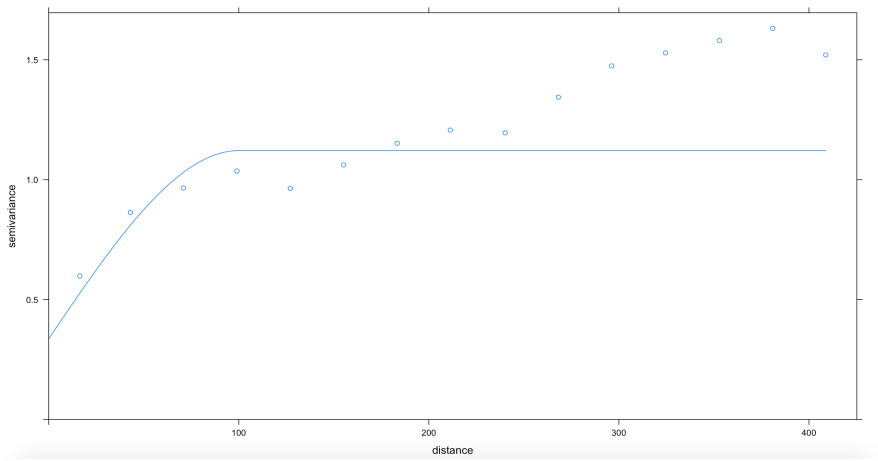

FIGURE 3.6: Spherical variogram model fit to data

isotropy holds for the domain of interest.

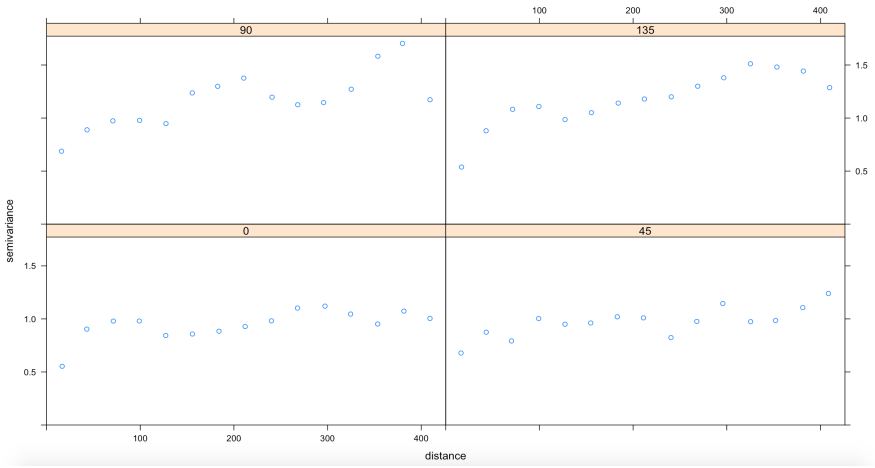

FIGURE 3.7: Directional variograms computed for directions 0, 45, 90, and 135 degrees

TABLE 3.1: FRK Mean squared prediction error over n=10 runs r=82 basis functions

MSE (mean,sd)				
Grid Used	Gaussian	Bisquare	Exponential	Matérn
Grid 1	(0.5629,0.02)	(0.5628,0.02)	(0.5636,0.02)	(0.5633,0.02)
Grid 2	(0.511,0.03)	(0.508,0.031)	(0.5138,0.032)	(0.513,0.03)

3.3 Results

Mean squared prediction error was calculated using a random sample of 75% of the data to train the models and 25% of the data to test the models. This procedure was performed 10 times, giving 10 different random samples of the data. For fixed rank kriging, the predicted value is taken to be the value of the BAU that contains the test data point. We list the mean and standard deviation of the resulting mean squared prediction error.

TABLE 3.2: Universal and Ordinary kriging compared to FRK with r=468 basis functions

Prediction accuracy results			
	Ordinary kriging	Universal kriging	FRK (r=468 basis functions)
Mean(MSPE)	0.4091	0.4038	0.4959
SD(MSPE)	0.006	0.006	0.028

TABLE 3.3: Time taken to build model and predict over BAUs

Average time taken (seconds)				
Grid Used	Gaussian	Bisquare	Exponential	Matérn
Grid 1	28	26	17.8	24
Grid 2	8	6.2	8	7.8

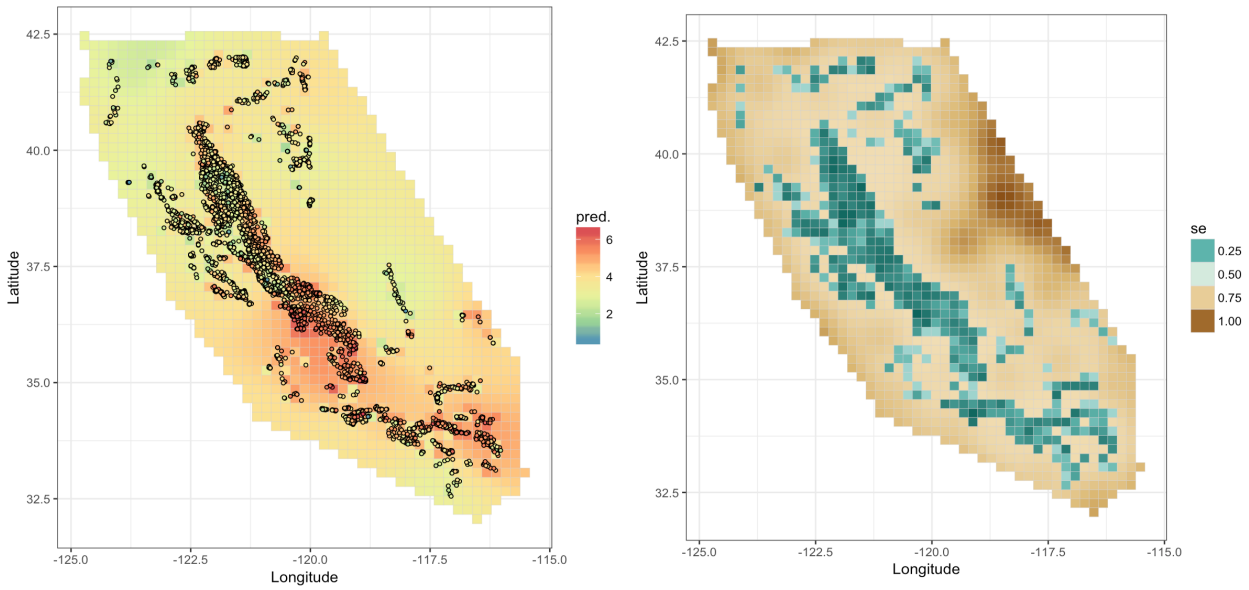

FIGURE 3.8: log(depth to groundwater) prediction map and standard errors

From table 3.1, we noticed that there was little difference between the basis functions chosen for the matrix S in terms of prediction accuracy. When comparing fixed rank kriging with universal and ordinary kriging, we used 468 bisquare basis functions of three different resolutions. From table 3.2, we can see that universal and ordinary kriging outperform fixed rank kriging in terms of prediction accuracy. Fixed rank kriging is much faster, however, averaging only 15 seconds to build the model and make predictions on 5,596 grid locations. Universal and ordinary kriging averaged 52 seconds, almost 4 times as long as fixed rank kriging. All times were recorded on a 2.9 GHz Intel Core i5 MacBook Pro (2015 model).

We can see the fixed rank kriging predictions and standard errors for the coarser resolution model (1,406 BAUs) in figure 3.8. We see that the standard error is higher in the areas without data points.

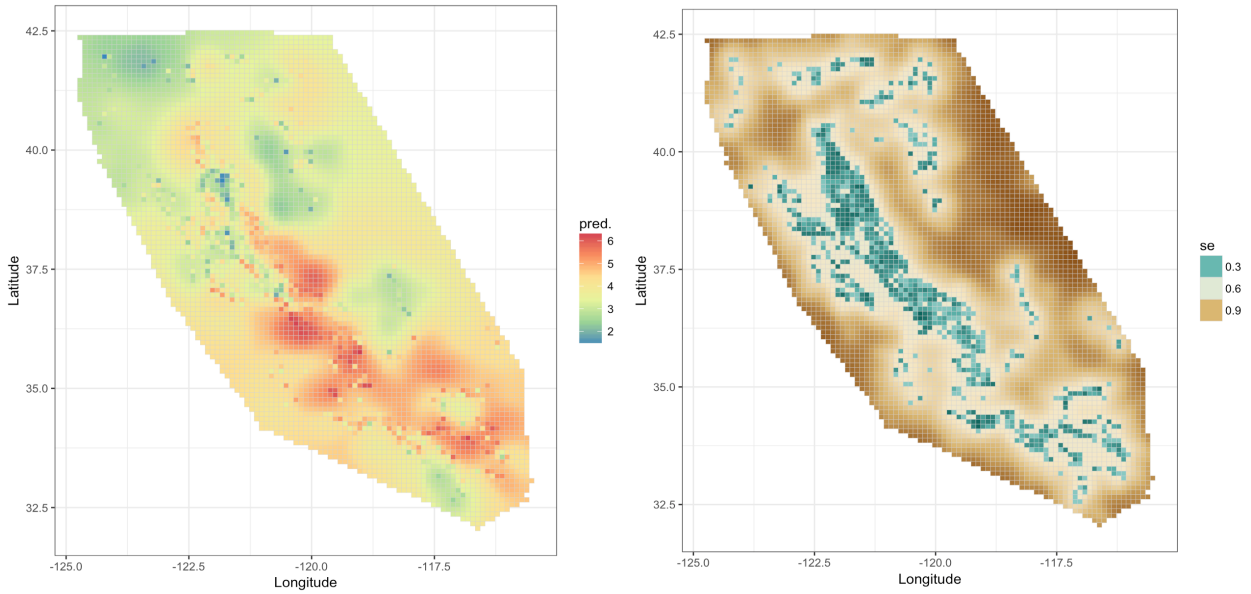

FIGURE 3.9: FRK prediction map (resolution 3) and standard errors

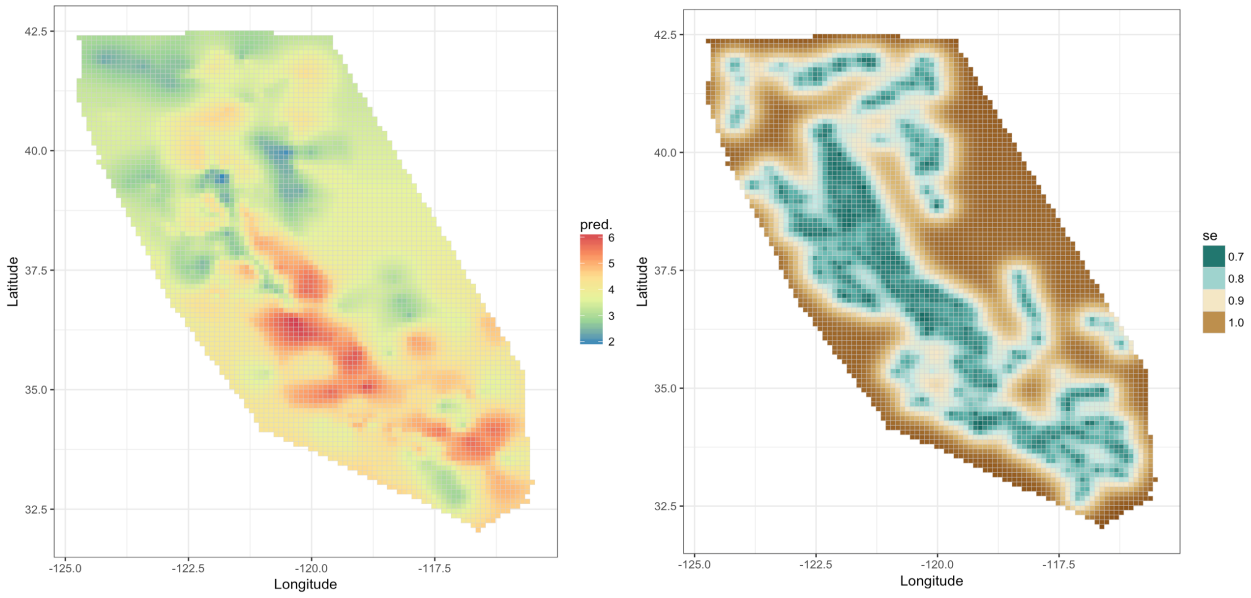

FIGURE 3.10: Universal kriging prediction map (resolution 3) and standard errors

From figures 3.9 and 3.10, we can see that the fixed rank kriging prediction map closely follows the universal kriging map. Both maps show that groundwater is furthest from the surface in the region of inland southern California and closest to the surface in the northwestern corner of California. The universal kriging map is much smoother than the fixed rank kriging map, however. Making the grid any finer, ie. increasing the number of BAUs past 5,596, led to a decrease in prediction accuracy for fixed rank kriging. Therefore, we settled on 5,596 BAUs for the FRK model.

3.4 Conclusion

For a dataset of this size (3,986 observations), the computational speedups of fixed rank kriging are not persuasive enough to support its use in place of universal kriging. As prediction accuracy is more important than computation time, we believe that universal kriging remains the de-facto prediction method at this scale. The smoothness of the universal kriging prediction and standard error maps are also preferred to the more choppy fixed rank kriging maps. For fixed rank kriging, we also found that varying the type of basis function did not have as much of an impact on increasing prediction accuracy as increasing the number of basis functions. Although fixed rank kriging remains a valid option for spatial prediction, we believe that the computational benefits of this approach do not outweigh the accuracy increase of maintaining the full covariance structure. In the case of predicting groundwater levels at this scale, universal kriging was shown to be the preferred method. In certain applications, where the domain of interest is very large and may include measurements with different supports, fixed rank kriging becomes a much more convincing alternative to universal kriging. For example, combining measurements from different orbiting satellites becomes a computational challenge for reasons of data size (upwards of 100,000 observations), nonstationarity, and varied support (some sensors use different grid sizes) [CJ08]. As an example use case, if kriging needs to be performed as part of a real-time system, on a global scale, we believe that fixed rank kriging offers advantages in terms of prediction speedup and preprocessing time.

REFERENCES

[BPG13] Roger S. Bivand, Edzer Pebesma, and Virgilio Gómez-Rubio. *Interpolation and Geostatistics*. Springer New York, New York, NY, 2013.

[CAg] "Groundwater Information." `http://www.water.ca.gov/groundwater/gwinfo/`. Accessed: 2017-06-2.

[CH80] Noel Cressie and Douglas M. Hawkins. "Robust estimation of the variogram: I." *Journal of the International Association for Mathematical Geology*, **12**(2):115–125, 1980.

[CJ08] Noel Cressie and Gardar Johannesson. "Fixed rank kriging for very large spatial data sets." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70**(1):209–226, 2008.

[Cre85] Noel Cressie. "Fitting variogram models by weighted least squares." *Journal of the International Association for Mathematical Geology*, **17**(5):563–586, 1985.

[Cre93] N.A.C. Cressie. *Statistics for spatial data*. Wiley series in probability and mathematical statistics: Applied probability and statistics. J. Wiley, 1993.

[GPH16] Benedikt Grler, Edzer Pebesma, and Gerard Heuvelink. "Spatio-Temporal Interpolation using gstat." *The R Journal*, **8**:204–218, 2016.

[GSS01] Tilmann Gneiting, Zoltn Sasvri, and Martin Schlather. "Analogies and Correspondences between Variograms and Covariance Functions." *Advances in Applied Probability*, **33**(3):617–630, 2001.

[KC11] Matthias Katzfuss and Noel Cressie. "Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets." *Journal of Time Series Analysis*, **32**(4):430–446, 2011.

[KW13] David Kahle and Hadley Wickham. "ggmap: Spatial Visualization with ggplot2." *The R Journal*, **5**(1):144–161, 2013.

[SC91] A. Stein and L. C. A. Corsten. "Universal Kriging and Cokriging as a Regression Procedure." *Biometrics*, **47**(2):575–587, 1991.

[Zam17] Andrew Zammit-Mangion. *FRK: Fixed Rank Kriging*, 2017. R package version 0.1.3.

CHAPTER 4

R Code

```
# Brice Randolph UCLA Statistics MS Thesis Code
# last edit: June 29 2017

# Data acquisition: 1. Visit
# https://gis.water.ca.gov/app/gicima/
# 2. Under'Select Layer Group',
# choose Fall 2016 Depth
#3. Select 'Download'

# Load necessary packages:

library(readr)
library(FRK)
library(sp)
library(ggplot2)
library(ggmap)
library(geoR)
library(gstat)
library(INLA)
library(foreign)
library(rgdal)
library(gridExtra)
```

```
######
# replace with path to dataset
h20 <- read.dbf(
#"~yourpathhere/Fall_2016_Depth_Points/
#F2016_DBGS_Points_20170327_102311.dbf")
# WSEL = groundwater surface elevation DGBS =
# groundwater depth below ground surface
# Preprocessing
h20 <- h20[, c("Site_Code", "WSEL", "Latitude", "Longitude",
    "DGBS")]
#remove duplicate reading from same site

h20 = h20[unique(h20$Site_Code), ]
h20 <- h20[, c("Latitude", "Longitude", "DGBS")]

coordinates(h20) = ~Longitude + Latitude
# Fixes distance calculations
proj4string(h20) = CRS("+proj=longlat")
h20 <- remove.duplicates(h20)
# Working with log transformed data with values > 1
h20Edit <- h20[h20$DGBS > 1, ]



# checking normality assumptions
hist(log(h20Edit$DGBS), main = "Log␣Depth␣to␣Groundwater",
    xlab = "")  #Appears normal
lmodel <- lm(log(h20Edit$DGBS) ~ (Latitude + 1), data = h20Edit)
h20DF <- data.frame(h20Edit)
```

```
summary(lmodel)
par(mfrow = c(1, 2))
hist(h20Edit$DGBS, main = "Histogram␣of␣depth␣to␣groundwater",
    xlab = "depth␣to␣groundwater␣(ft)")
hist(log(h20Edit$DGBS), main =
"Histogram␣of␣log(depth␣to␣groundwater)",
    xlab = "log(depth␣to␣groundwater)")
### looking at normality in detrended values
hist(lmodel$residuals, main = "Histogram␣of␣detrended␣data",
    xlab = "residuals")
qqnorm(lmodel$residuals)
qqline(lmodel$residuals, col = 2)
######### Procedure for creating single train/test set to
######### evaluate prediction Code at end contains the
######### procedure for replicating this 10 different times
set.seed(1)
trainIndices <- sample(1:length(h20Edit), length(h20Edit)/4,
    replace = FALSE)
test <- h20Edit[trainIndices, ]
train <- h20Edit[-trainIndices, ]
##### Creating the Basic Aerial Units(BAUs) Essentially
##### discretizes the domain of interest

# @ arguments: 2D plane, BAU cellsize, grid (not
# hex), data around which to create BAUs, border
# buffer factor,
GridBAUs1 <- auto_BAUs(manifold = plane(), cellsize = c(0.2,
    0.2), type = "grid", data = h20Edit, convex = -0.05,
    nonconvex_hull = FALSE)
```

```
h20Pts <- h20Edit
h20Pts$DGBS <- NULL
GridBAUs2 <- BAUs_from_points(h20Pts)
# this (GridBAUs2) has terrible performance
# compared to the other BAU methods and is left out
# of the report
GridBAUs3 <- auto_BAUs(manifold = plane(), cellsize = c(0.1,
    0.1), type = "grid", data = h20Edit, convex = -0.05,
    nonconvex_hull = FALSE)
GridBAUs4 <- auto_BAUs(manifold = plane(), cellsize = c(0.05,
    0.05), type = "grid", data = h20Edit, convex = -0.05,
    nonconvex_hull = FALSE)

plot(GridBAUs1)
plot(GridBAUs2)
plot(GridBAUs3)
plot(GridBAUs4)

GridBAUs1$fs <- 1  # fine-scale variation at BAU level
GridBAUs2$fs <- 1
GridBAUs3$fs <- 1
GridBAUs4$fs <- 1

# Types of basis functions used: 'bisquare',
# 'Gaussian','exp', 'Matern32'
G <- auto_basis(manifold = plane(), data = h20Pts,
    nres = 2, type = "Matern32", regular = 0)
show_basis(G) + coord_fixed() + xlab("Longitude") +
    ylab("Latitude")
```

```
## Note: show basis assumes spherical distance
## functions when plotting
G2 <- auto_basis(manifold = plane(), data = h20Pts,
    nres = 3, type = "bisquare", regular = 0)
show_basis(G2) + coord_fixed() + xlab("(b)") + ylab("Latitude")
## Note: show basis assumes spherical distance
## functions when plotting

f1 <- log(DGBS) ~ 1 + Latitude  # formula for SRE model
f2 <- log(DGBS) ~ 1  # formula for SRE model
# @ parameters: formula, list of datasets, BAUs,
# basis functions, estimation measurement error
S1 <- SRE(f = f1, data = list(train), BAUs = GridBAUs1,
    basis = G, est_error = TRUE, average_in_BAU = FALSE)
# @ parameters: model, max. num EM iterations,
# tolerance at which EM is assumed to have
# converged, bool print log-likelihood at each
# iteration
S1 <- SRE.fit(SRE_model = S1, n_EM = 100, tol = 0.1,
    print_lik = TRUE)
GridBAUs1 <- SRE.predict(SRE_model = S1, obs_fs = FALSE)


MSE_FRK1<-mean((log(test$DGBS)-over(test, GridBAUs1)$mu)^2)
#returns the GridBAUs1 entry with closest pixel
MSE_FRK1
head(over(test, GridBAUs1)$Latitude)
head(test$Latitude)  # sanity check
```

```
################ second FRK BAU setting
S2 <- SRE(f = f1, data = list(train), BAUs = GridBAUs2,
    basis = G, est_error = TRUE, average_in_BAU = FALSE)

S2 <- SRE.fit(SRE_model = S2, n_EM = 100, tol = 0.01,
    print_lik = TRUE)
GridBAUs2 <- SRE.predict(SRE_model = S2, obs_fs = FALSE)
#returns the GridBAUs1 entry with closest pixel
MSE_FRK2 <- mean((log(test$DGBS) - over(test, GridBAUs2)$mu)^2)
MSE_FRK2

################ third FRK BAU setting
S3 <- SRE(f = f1, data = list(train), BAUs = GridBAUs3,
    basis = G, est_error = TRUE, average_in_BAU = FALSE)

S3 <- SRE.fit(SRE_model = S3, n_EM = 100, tol = 0.01,
    print_lik = TRUE)
GridBAUs3 <- SRE.predict(SRE_model = S3, obs_fs = FALSE)

#returns the GridBAUs1 entry with closest pixel

MSE_FRK3 <- mean((log(test$DGBS) - over(test, GridBAUs3)$mu)^2)
MSE_FRK3
################################## Other kriging methods

### variogram for ordinary kriging
train_g <- gstat(id = "log_dist", formula = log(DGBS) ~
    1, data = train)
vg <- variogram(train_g)
```

```
# robust variogram calculation

vgRobust <- variogram(train_g, cressie = TRUE)
plot(vg)
plot(vgRobust)
lm(vgRobust$gamma ~ vgRobust$dist)  # sigma2e = 0.664
v.fit <- fit.variogram(vgRobust, vgm(1.5, "Sph", 300,
    0.5))
v.fit
plot(vg, v.fit)
##### variogram for universal kriging
train_g_U <- gstat(id = "log_dist", formula = log(DGBS) ~
    1 + Latitude, data = train)
vg_U <- variogram(train_g_U)
dir.vgm <- variogram(train_g_U, alpha = c(0, 45, 90,
    135))
plot(dir.vgm)
v.fit_U <- fit.variogram(vg_U, vgm(1.5, "Sph", 400,
    0.5))
plot(vg_U, v.fit_U)


### ordinary and universal kriging
OK <- krige(id = "logDist", formula = log(DGBS) ~ 1,
    train, newdata = test, model = v.fit)

#returns the GridBAUs1 entry with closest pixel
MSE_OK <- mean((log(test$DGBS) - OK$logDist.pred)^2)
UK <- krige(id = "logDist", formula = log(DGBS) ~ 1 +
```

```
    Latitude, train, newdata = test, model = v.fit_U)
MSE_UK <- mean((log(test$DGBS) - UK$logDist.pred)^2)

# UK2 is used to produce the analogous kriging map
# of california using OK and UK as opposed to FRK
UK2 <- krige(id = "logDist", formula = log(DGBS) ~
    1 + Latitude, train, newdata = GridBAUs3, model = v.fit_U)


# compare MSE of single trial: Actual comparisons
# are done after replicating the procedure 10 times
# (shown at end of code)
MSE_FRK1
MSE_FRK2
MSE_FRK3
MSE_OK
MSE_UK
MSE_OKL
### Plotting the predictions and standard errors

# need to coerce the BAUs to a data frame for ggplot2

BAUs_df <- as(GridBAUs3, "data.frame")
g_grid2FRK <- ggplot() + geom_tile(data = BAUs_df,
    aes(Longitude, Latitude, fill = mu), colour = "light␣grey") +
    scale_fill_distiller(palette = "Spectral", name = "pred.") +
    coord_fixed() + xlab("Longitude") + ylab("Latitude") +
    theme_bw()
# the following can be added to plot data over the
```

```
# map geom_point(data=data.frame(h20Edit), # Plot
# data aes(Longitude,Latitude,fill=log(DGBS)), #
# Colour <-> log(zinc) colour='black', # point
# outer colour pch=21, size=1) # size of point
g_grid2FRK  # to view the map

# Similar to above but with standard errors
g2 <- ggplot() + geom_tile(data = BAUs_df, aes(Longitude,
    Latitude, fill = sqrt(var)), colour = "light␣grey") +
    scale_fill_distiller(palette = "BrBG", name = "s.e.",
        guide = guide_legend(title = "se")) + coord_fixed() +
    xlab("Longitude") + ylab("Latitude") + theme_bw()
g2
grid.arrange(g1, g2, ncol = 2)
########## plotting the data over california
myLocation <- c(lon = -119, lat = 36)
### myMap = watercolor for just showing locations
### myMap2 = toner for contrasting color grad
myMap <- get_map(location = myLocation, source = "stamen",
    maptype = "watercolor", crop = FALSE, zoom = 6)
myMap2 <- get_map(location = myLocation, source = "stamen",
    maptype = "toner", crop = FALSE, zoom = 6)
ggmap(myMap2) + geom_point(aes(x = Longitude, y = Latitude,
    colour = log(DGBS)), size = 1, data = data.frame(h20Edit),
    alpha = 0.5) + scale_colour_gradientn(colours = rainbow(4))
####

# Plotting universal kriging predictions and
# standard errors
```

```
UK_df <- as(UK2, "data.frame")  #logDist.pred logDist.var
gUKp <- ggplot() + geom_tile(data = UK_df, aes(Longitude,
    Latitude, fill = logDist.pred), colour = "light␣grey") +
    scale_fill_distiller(palette = "Spectral", name = "pred.") +
    coord_fixed() + xlab("Longitude") + ylab("Latitude") +
    theme_bw()
gUKp

gUKse <- ggplot() + geom_tile(data = UK_df, aes(Longitude,
    Latitude, fill = sqrt(logDist.var)), colour = "light␣grey") +
    scale_fill_distiller(palette = "BrBG", name = "s.e.",
        guide = guide_legend(title = "se")) + coord_fixed() +
    xlab("Longitude") + ylab("Latitude") + theme_bw()
gUKse
grid.arrange(g_grid2FRK, gUKp, ncol = 2)

############ Replicating the procedure on 10 different test
############ sets Seed changes 10 times

mseListOK <- list()
mseListUK <- list()

mseListFRK1 <- list()
time <- list()

for (i in 1:10) {
    set.seed(i)
    trainIndices <- sample(1:length(h20Edit), length(h20Edit)/4,
```

```
        replace = FALSE)
    testi <- h20Edit[trainIndices, ]
    traini <- h20Edit[-trainIndices, ]

    start.timei <- Sys.time()
    S1i <- SRE(f = f1, data = list(traini), BAUs = GridBAUs1,
        basis = G, est_error = TRUE, average_in_BAU = FALSE)

    S1i <- SRE.fit(SRE_model = S1i, n_EM = 100, tol = 0.1,
        print_lik = TRUE)
    GridBAUs1i <- SRE.predict(SRE_model = S1i, obs_fs = FALSE)

    end.timei <- Sys.time()
    time.takeni <- end.timei - start.timei
    #(over) returns the GridBAUs1 entry with closest pixel
    MSE_FRK1i <- mean((log(testi$DGBS) - over(testi,
        GridBAUs1i)$mu)^2)



    time <- c(time, time.takeni)
    mseListFRK1 <- c(mseListFRK1, MSE_FRK1i)
}
# results of first FRK procedure (larger BAUs)
mean(unlist(mseListFRK1))
sd(unlist(mseListFRK1))
mean(unlist(time))
sd(unlist(time))
```

```
####
mseListFRK3 <- list()
timeFRK3 <- list()

for (i in 1:10) {
    set.seed(i)
    trainIndices <- sample(1:length(h20Edit), length(h20Edit)/4,
        replace = FALSE)
    testi <- h20Edit[trainIndices, ]
    traini <- h20Edit[-trainIndices, ]



    start.timei <- Sys.time()
    S3i <- SRE(f = f1, data = list(traini), BAUs = GridBAUs3,
        basis = G2, est_error = TRUE, average_in_BAU = FALSE)

    S3i <- SRE.fit(SRE_model = S3i, n_EM = 100, tol = 1,
        print_lik = TRUE)
    GridBAUs3i <- SRE.predict(SRE_model = S3i, obs_fs = FALSE)
    end.timei <- Sys.time()
    time.takeni <- end.timei - start.timei
    MSE_FRK3i <- mean((log(testi$DGBS) - over(testi,
        GridBAUs3i)$mu)^2)



    timeFRK3 <- c(timeFRK3, time.takeni)
    mseListFRK3 <- c(mseListFRK3, MSE_FRK3i)
}
# results of finer grid model FRK
```

```
mean(unlist(timeFRK3))
sd(unlist(timeFRK3))
mean(unlist(mseListFRK3))
sd(unlist(mseListFRK3))


####
mseListFRK1 <- list()
time <- list()

# Universal kriging and Ordinary kriging MSE are
# computed as before, using
# MSE_UK<-mean((log(test$DGBS)-UK$logDist.pred)^2)
# for the different test sets

for (i in 1:10) {
    set.seed(i)
    trainIndices <- sample(1:length(h20Edit), length(h20Edit)/4,
        replace = FALSE)
    testi <- h20Edit[trainIndices, ]
    traini <- h20Edit[-trainIndices, ]

    start.timei <- Sys.time()
    ##### variogram for universal kriging
    train_g_U <- gstat(id = "log_dist", formula = log(DGBS) ~
        1 + Latitude, data = traini)
    vg_U <- variogram(train_g_U)
    v.fit_U <- fit.variogram(vg_U, vgm(1.5, "Sph",
        400, 0.5))
```

```
    UK2 <- krige(id = "logDist", formula = log(DGBS) ~
        1 + Latitude, traini, newdata = GridBAUs3,
        model = v.fit_U)

    end.timei <- Sys.time()
    time.takeni <- end.timei - start.timei
    time <- c(time, time.takeni)
}

mean(unlist(time))  # avg time = 32 sec. sd 19.6
```