6. Simple Spatial Prediction Models

In this section we consider the simplest spatial prediction models that incorporate random effects. These spatial prediction models are part of a larger class of models known as *kriging models* [in honor of the South African mining engineer, D.G. Krige, who pioneered the use of statistical methods in ore-grade sampling in the early 50's].[1] So before launching into the details of the specific models developed in this section, it is appropriate to begin with a general overview of kriging models.

6.1 An Overview of Kriging Models

From a formal viewpoint, kriging models are closely related to the kernel smoothing models developed in Sections 5.1 and 5.2 above. In particular, the fundamental idea of predicting values based on local information is exactly the same. In fact, a slight modification of Figure 5.2, as in Figure 6.1 below, serves to illustrate the main ideas.

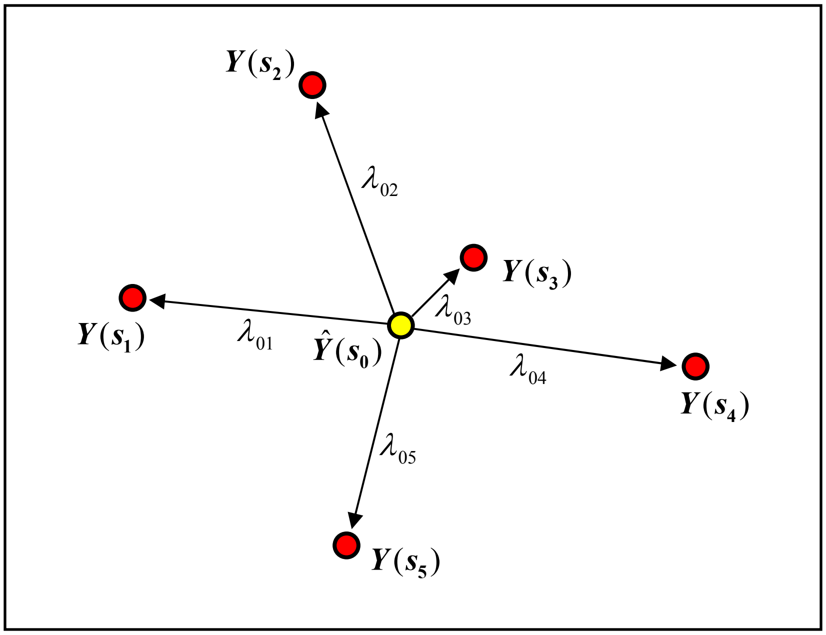


Figure 6.1 Basic Kriging Framework

Given spatial data, $y(s)$, at a set of locations, $\{s_i : i = 1,..,n\} \subset R$, we again consider the prediction of the unobserved value at some location, $s_0 \in R$. The first key difference is that we now treat the observed data as a finite sample from a spatial stochastic process $\{Y(s) : s \in R\}$. As in the case of deterministic interpolation, not all sample data is necessarily relevant for prediction at s_0. Hence, for the present, we again assume that some appropriate subset of sample locations,

[1] For further background discussion of kriging methods see Cressie (1990) and (1993, p.106).

(6.1.1) $$S(s_0) \subseteq \{s_i : i = 1,..,n\}$$

has been chosen for prediction, which for convenience we here designate as the *prediction set* at s_0 (rather than "interpolation set"). The choice of $S(s_0)$ will of course play a major role in determining the final prediction value at s_0. But it will turn out that the best way to choose these sets is first to determine a "best prediction" for any given set, $S(s_0)$, and then determine a "best prediction set" by comparing these predictions. This procedure, known as *cross validation,* will be developed in Section 6.4 below.

So given prediction set, $S(s_0) = \{s_1,..,s_{n_0}\}$, the next question is how to determine a prediction, $\hat{y}(s_0)$, based on the sample data, $\{y(s_1),..,y(s_{n_0})\}$. Given the present stochastic framework, this question is more properly posed by treating this prediction as a *random variable*, $\hat{Y}(s_0)$, and asking how it can be determined as a function of the random variables, $\{Y(s_1),..,Y(s_{n_0})\}$, associated with the observed data. As with kernel smoothers, we again hypothesize that $\hat{Y}(s_0)$ can be represented as some *linear combination* of these random variables, i.e., that $\hat{Y}(s_0)$ is of the form:

(6.1.2) $$\boxed{\hat{Y}(s_0) = \sum\nolimits_{i=1}^{n_0} \lambda_{0i} Y(s_i)}$$

where the weights λ_{0i} are yet to be determined. This fundamental hypothesis shall be referred to as the *linear prediction hypothesis*.

6.1.1 Best Linear Unbiased Predictors

In contrast to kernel smoothing, the unknown weights λ_{0i} in (6.1.2) need not be simple functions of distance (so that λ_{0i} in Figure 6.1 now replaces d_{0i} in Figure 5.2).[2] In any case, the key strategy of kriging models is to choose weights that are "statistically optimal" in an appropriate sense. To motivate this approach in the simplest way, we begin by designating the difference between $\hat{Y}(s_0)$ and the unknown true random variable, $Y(s_0)$, as the *prediction error*,

(6.1.3) $$e(s_0) = Y(s_0) - \hat{Y}(s_0)$$

This prediction error will play a fundamental role in the analysis to follow. But before proceeding, it is important to distinguish prediction error, $e(s_0)$, from the random effects

[2] One would expect that points s_i closer to s_0 will tend to have larger weights, λ_{0i}. However we shall see in Section 6.2.3 below that is not true, even when spatial correlations decrease with distance.

term, $\varepsilon(s_0)$, in our basic stochastic model, $Y(s_0) = \mu(s_0) + \varepsilon(s_0)$. While they can both viewed as "random errors", the random effects term, $\varepsilon(s_0)$, describes the deviation of $Y(s_0)$ from its mean, so that by definition, $E[\varepsilon(s_0)] = 0$. This is certainly *not* part of the definition of prediction error.

However, it is clearly desirable that prediction errors satisfy this zero-mean property, i.e., that prediction error on average be zero. Indeed, this is our first statistical optimality criterion, usually referred to as the *unbiasedness criterion*:

$$(6.1.4) \qquad \boxed{E[e(s_0)] = E[Y(s_0) - \hat{Y}(s_0)] = 0}$$

All predictors, $\hat{Y}(s_0)$, satisfying both (6.1.2) and (6.1.4) are referred to as *linear unbiased* predictors of $Y(s_0)$. In these terms, our single most important optimality criterion is that among all possible linear unbiased predictors, the prediction error of $\hat{Y}(s_0)$ should be as "close to zero" as possible. While there are many ways to define "closeness to zero", for the case of random prediction error it is natural to require that the *mean squared error*, $E[e(s_0)^2]$, be as small as possible.[3] Hence our third criterion, designated as the *efficiency criterion* is that $\hat{Y}(s_0)$ have *minimum mean squared error among all linear unbiased predictors*.

This criterion is so pervasive in the statistical literature that it is given many different names. On the one hand, if we abbreviate "minimum mean squared error" as MMSE, then such predictors are often called *MMSE predictors*. In addition, notice that since unbiasedness ($E[e(s_0)] = 0$) implies

$$(6.1.5) \qquad \text{var}[e(s_0)] = E[e(s_0)^2] - \left(E[e(s_0)]\right)^2 = E[e(s_0)^2] \ ,$$

such predictors are also instances of *minimum variance predictors*. However, to emphasize their optimality among all linear unbiased predictors, it is most accurate to designate them as *best linear unbiased predictors*, or *BLU predictors*. It is this latter terminology that we shall use throughout.

6.1.2 Model Comparisons

Within this general framework we consider four different kriging models, proceeding from simpler to more general models. These models are each characterized by the specific assumptions made about the properties of the underlying spatial stochastic process, $\{Y(s) = \mu(s) + \varepsilon(s) : s \in R\}$. For all such models, we start with a fundamental

[3] Another possibility would be to require that the mean absolute error, $E[\,|\,e(s_0)\,|\,]$, be as small as possible. However, since the absolute-value function is not differentiable at zero, this criterion turns out to be much more difficult to analyze.

normality assumption about spatial random effects. In particular, for each finite set of locations $\{s_i : i = 1,..,n\}$ in region *R*, it will be assumed that the associated spatial random effects $[\varepsilon(s_i) : i = 1,..,n]$ are *multi-normally distributed*.[4] Since $E[\varepsilon(s)] \equiv 0$, by definition, this distribution is determined entirely by the covariances, $\text{cov}[\varepsilon(s_i), \varepsilon(s_j)]$, $i, j = 1,..,n$. Hence the assumptions characterizing each model can be summarized in terms of assumptions about (*i*) the spatial trend, $\mu(s)$, and (*ii*) the covariances, $\text{cov}[\varepsilon(s), \varepsilon(s')]$, between pairs of random errors.

Before stating these assumptions, it is important to make one additional clarification. When a given parameter such a mean value, μ, is assumed to be "known" or "unknown", these terms have very specific meanings. In particular, one almost never actually "knows" the value of any parameter. Rather, a phrase like "μ known" is taken to mean that the value of this parameter is determined *outside* of the given model. Similarly, "μ unknown" is taken to mean that the value of this parameter is to be determined *inside* (i.e., as part of) the given model.[5]

Simple Kriging Model

Here "simple" refers to the (rather heroic!) assumption that underlying stochastic process itself is entirely *known*. In addition, it is also assumed that the spatial trend is *constant*. More formally, this amounts to the assumptions:

$$\boxed{\begin{array}{ll} \mu(s) = \mu \;\; known \; , \;\; s \in R & \qquad (6.1.6) \\[1em] \text{cov}[\varepsilon(s), \varepsilon(s')] \;\; known \; , \;\; s, s' \in R & \qquad (6.1.7) \end{array}}$$

Before proceeding, it is reasonable to ask why one would even want to consider this model. Since all parameters of the stochastic process are determined outside the model, it would appear that there is nothing left to be done. But remember that the underlying stochastic process model serves only as a statistical framework for carrying out *spatial prediction*. In particular, given any location, $s_0 \in R$, and associated prediction set, $S(s_0) = \{s_1,..,s_{n_0}\}$, the basic task is to predict a value for $Y(s_0)$ given observed values of $\{Y(s_1),..,Y(s_{n_0})\}$. So in terms of the linear prediction hypothesis in (6.1.2), the key *prediction weights*, $(\lambda_{0i} : i = 1,..,n_0)$, are still *unknown*, i.e., are yet to be determined. Hence the chief advantage of this *simple* kriging model from a conceptual viewpoint is to

[4] In addition there is an obvious "consistency" condition that must also be satisfied. For example, if $\{Y(s_1), Y(s_2)\}$ is bivariate normal, then the univariate normal distributions for subsets $\{Y(s_1)\}$ and $\{Y(s_2)\}$ must of course be the *marginal distributions* of $\{Y(s_1), Y(s_2)\}$. More generally each subset of size *k* from the *n*-variate normal, $\{Y(s_1),..., Y(s_n)\}$ must have precisely the corresponding *k*-variate marginal normal distribution.

[5] A somewhat more accurate terminology would be to use "μ exogenous" and "μ endogenous". But the terms "known" and "unknown" are so widely used that we choose stay with this convention.

allow us to derive optimal prediction weights without having to worry about estimating other unknown parameters at the same time.

Ordinary Kriging Model

The only difference between this model and simple kriging is that the constant mean, μ, is now assumed to be unknown, and hence must be estimated within the model. More formally, it is assumed that

$$\begin{array}{l}(6.1.8)\\[1em](6.1.9)\end{array}\qquad\boxed{\begin{array}{l}\mu(s)=\mu\ \ \textit{unknown}\ ,\ \ s\in R\\[1em]\operatorname{cov}[\varepsilon(s),\varepsilon(s')]\ \ \textit{known}\ ,\ \ s,s'\in R\end{array}}$$

This *ordinary kriging model* in fact the simplest kriging model that is actually used in practice. As will be seen below, the constant-mean assumption (6.1.8) allows both the mean and covariances to be estimated in a direct way from observed data. So a practical estimation procedure is available for this model. However, one may still ask why this model is of any interest from a *spatial* viewpoint when all variations in spatial trends are assumed away. The key point to keep in mind here is that spatial variation is still present in this model, but all such variation is assumed to be captured by the *covariance structure* of the model. We shall return to this issue in Section 6.3 below.

Universal Kriging Model

We turn now to kriging models that do allow for explicit variation in the trend function, $\mu(s)$. The simplest of these, designated as the *universal kriging model*, allows the trend function to be modeled as a linear function of spatial attributes, but maintains the assumption that all covariances are known. More formally, if we now let $x(s)=[x_1(s),..,x_k(s)]'$ denote a (column) vector of *spatial attributes* [which may include the coordinate attributes, $s=(s_1,s_2)$, themselves], and let $\beta=(\beta_1,..,\beta_k)'$ denote a corresponding vector of coefficients, then this model is characterized by the assumptions:

$$\begin{array}{l}(6.1.10)\\[1em](6.1.11)\end{array}\qquad\boxed{\begin{array}{l}\mu(s)=x(s)'\beta,\ \ \beta\ \ \textit{unknown},\ \ s\in R\\[1em]\operatorname{cov}[\varepsilon(s),\varepsilon(s')]\ \ \textit{known}\ ,\ \ s,s'\in R\end{array}}$$

Here it should be emphasized that "linear" means *linear in parameters* (β). For example, if $x(s)=[1,s_1,s_2,s_1^2,s_2^2,s_1s_2]'$ so that

$$(6.1.12)\qquad \mu(s)=\beta_0+\beta_1\,s_1+\beta_2\,s_2+\beta_3\,s_1^2+\beta_4\,s_2^2+\beta_5\,s_1s_2\ \ ,$$

then the trend, $\mu(s)$, is a *quadratic* function of the coordinates, $s=(s_1,s_2)$, but is *linear* in the parameter vector, $\beta=(\beta_0,\beta_1,\beta_2,\beta_3,\beta_4,\beta_5)'$.

Geostatistical Kriging Model

Our final kriging model relaxes the assumption that covariances are known. More formally, this *geostatistical kriging model* (or simply, *geo-kriging model*) is characterized by the following assumptions:

$$\boxed{\begin{array}{ll} \mu(s) = x(s)'\beta\,,\;\; \beta \;\, unknown\,,\;\; s \in R & (6.1.13) \\[2ex] \mathrm{cov}[\varepsilon(s),\varepsilon(s')] \;\, unknown\;,\;\; s,s' \in R & (6.1.14) \end{array}}$$

In this model, the spatial trend parameters, β, as well as all covariance parameters must be *simultaneously* estimated. While this procedure is clearly more complex from an estimation viewpoint, it provides the most general framework for spatial prediction in terms of prior assumptions. Hence our ultimate goal in this part of the NOTEBOOK is to develop this geostatistical kriging model in full, and show how it can be estimated.

6.2 The Simple Kriging Model

To develop the basic idea of kriging, we start by assuming as in (6.1.6) and (6.1.7) above that the relevant spatial stochastic process, $\{Y(s) = \mu(s) + \varepsilon(s) : s \in R\}$ has a *constant* mean, $E[Y(s)] = \mu(s) \equiv \mu$, and that this mean value, μ, together with all covariances, $\mathrm{cov}[\varepsilon(s),\varepsilon(s')]$, $s,s' \in R$ have already been estimated. We shall return to such estimation questions below. But for the present we simply take all these values to be given. In this setting, observe that if we want to predict a value, $Y(s_0)$, at some location, $s_0 \in R$, then since $\mu(s_0) = \mu$ is already known, we see from the identity,

$$Y(s_0) = \mu + \varepsilon(s_0) \tag{6.2.1}$$

that it suffices to predict the associated *error*, $\varepsilon(s_0)$. Moreover, if we are given a finite set of sample points, $\{s_1,..,s_n\} \subset R$ where observations, $\{y(s_1),..,y(s_n)\}$ have been made, then in fact we have already "observed" values of the associated errors, namely,

$$\varepsilon(s_i) \;=\; y(s_i) - \mu \;\;,\;\; i = 1,..,n \tag{6.2.2}$$

Hence if $S(s_0) = \{s_1,..,s_{n_0}\} \subseteq \{s_1,..,s_n\}$ denotes the relevant prediction set at s_0, then the *linear prediction hypothesis* for $\varepsilon(s_0)$ in this setting reduces to finding a linear combination,

$$\boxed{\hat{\varepsilon}(s_0) = \textstyle\sum_{i=1}^{n_0} \lambda_{i0}\, \varepsilon(s_i)} \tag{6.2.3}$$

which yields a *Best Linear Unbiased* (BLU) predictor of $\varepsilon(s_0)$. The corresponding predictor of $Y(s_0)$ is then defined to be

$$(6.2.4) \qquad \hat{Y}(s_0) = \mu + \hat{\varepsilon}(s_0)$$

Note since by definition all errors, $\varepsilon(s)$, $s \in R$, have zero means, it then follows at once from (6.2.1) and (6.2.4) together with the *linearity* of expectations that,

$$(6.2.5) \qquad E[Y(s_0) - \hat{Y}(s_0)] = E[\varepsilon(s_0) - \hat{\varepsilon}(s_0)]$$

$$= E[\varepsilon(s_0) - \sum\nolimits_{i=1}^{n_0} \lambda_{i0}\varepsilon(s_i)]$$

$$= E[\varepsilon(s_0)] - \sum\nolimits_{i=1}^{n_0} \lambda_{i0} E[\varepsilon(s_i)] \equiv 0$$

and hence that the *unbiasedness condition* is automatically satisfied for $\hat{Y}(s_0)$ [and indeed, for every possible linear estimator given by (6.2.3) and (6.2.4)]. This means that for simple kriging, BLU prediction reduces precisely to *Minimum Mean Squared Error* (MMSE) prediction. So the task remaining is to find the vector of weights, $\lambda_0 = (\lambda_{0i} : i = 1,..,n_0)'$ in (6.2.3) that minimize *mean squared error*:

$$(6.2.6) \qquad MSE(\lambda_0) = E\left([Y(s_0) - \hat{Y}(s_0)]^2\right) = E\left([\varepsilon(s_0) - \hat{\varepsilon}(s_0)]^2\right)$$

Here it might seem that without further information about the distributions of these errors, one could say very little. But surprisingly, it is enough to know their first and second moments [as *assumed* in (6.1.6) and (6.1.7) above]. To see this, we begin by introducing some simplifying notation. First, as in (1.1.1) above, we drop the explicit reference to locations and now write simply

$$(6.2.7) \qquad \varepsilon(s_i) = \varepsilon_i \ , \ i = 0,1,..,n_0$$

[Here it is worth noting that the choice of "0" for the prediction location is very convenient in that it often allows this location to be indexed together with it predictor locations, as in (6.2.7).] Next, recalling that $E(\varepsilon_i) \equiv 0$ it follows that variances and covariances for the predictor variables can be represented, respectively, as

$$(6.2.8) \qquad \text{var}(\varepsilon_i) = E(\varepsilon_i^2) = \sigma_{ii} \ , \ i = 1,..,n_0$$

$$(6.2.9) \qquad \text{cov}(\varepsilon_i, \varepsilon_j) = E(\varepsilon_i \varepsilon_j) = \sigma_{ij} \ , \ i, j = 1,..,n_0 \ (j \neq i)$$

In addition, the corresponding variance and covariances for unknown error, ε_0, to be predicted can be written as

$$\text{(6.2.10)} \qquad \text{var}(\varepsilon_0) = E(\varepsilon_0^2) = \sigma^2$$

$$\text{(6.2.11)} \qquad \text{cov}(\varepsilon_0,\varepsilon_i) = \sigma_{0i}\,,\; i = 1,..,n_0$$

Notice in particular that in the *variance* expression (6.2.10) we have omitted subscripts and written simply $\sigma_{00} \equiv \sigma^2$. This variance will play a special role in many of the expressions to follow. Moreover, since only *stationary* models of covariance will actually be used in our kriging applications, this variance will be independent of location s_0.[6] In these terms, we can now write mean squared error explicitly in terms of these parameter values as follows:

$$\text{(6.2.12)} \qquad MSE(\lambda_0) = E\left([\varepsilon(s_0) - \hat{\varepsilon}(s_0)]^2\right) = E\left[\left(\varepsilon_0 - \sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_i\right)^2\right]$$

$$= E\left[\varepsilon_0^2 - 2\,\varepsilon_0\sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_i + \left(\sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_i\right)^2\right]$$

$$= E\left(\varepsilon_0^2\right) - 2\,E\left(\varepsilon_0\sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_i\right) + E\left[\left(\sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_i\right)^2\right]$$

But since

$$\text{(6.2.13)} \qquad E\left(\varepsilon_0\sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_i\right) = E\left(\sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_0\varepsilon_i\right) = \sum\nolimits_{i=1}^{n_0}\lambda_{0i}E(\varepsilon_0\varepsilon_i) = \sum\nolimits_{i=1}^{n_0}\lambda_{0i}\,\sigma_{0i}$$

and since the product identity

$$\text{(6.2.14)} \qquad \left(\sum\nolimits_{i=1}^{n}x_i\right)^2 = \left(\sum\nolimits_{i=1}^{n}x_i\right)\left(\sum\nolimits_{j=1}^{n}x_j\right) = \sum\nolimits_{i=1}^{n}x_i\left(\sum\nolimits_{j=1}^{n}x_j\right) = \sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n}x_ix_j$$

implies that

$$\text{(6.2.15)} \qquad E\left[\left(\sum\nolimits_{i=1}^{n_0}\lambda_{0i}\varepsilon_i\right)^2\right] = E\left(\sum\nolimits_{i=1}^{n_0}\sum\nolimits_{j=1}^{n_0}\lambda_{0i}\lambda_{0j}\varepsilon_i\varepsilon_j\right)$$

$$= \sum\nolimits_{i=1}^{n_0}\sum\nolimits_{j=1}^{n_0}\lambda_{0i}\lambda_{0j}E(\varepsilon_i\varepsilon_j) = \sum\nolimits_{i=1}^{n_0}\sum\nolimits_{j=1}^{n_0}\lambda_{0i}\lambda_{0j}\,\sigma_{ij}\;\;,$$

[6] Note that this also implies subscripts could be dropped on all predictor variances, σ_{ii}. But here it is convenient to maintain these subscripts so that expressions involving all predictor variances and covariances can be stated more easily

it follows by substituting (6.2.13) and (6.2.15) into (6.2.12) that

$$\boxed{MSE(\lambda_0) = \sigma^2 - 2\sum_{i=1}^{n_0} \lambda_{0i}\sigma_{0i} + \sum_{i=1}^{n_0}\sum_{j=1}^{n_0} \lambda_{0i}\lambda_{0j}\sigma_{ij}} \tag{6.2.16}$$

Thus mean squared error, $MSE(\lambda_0)$, is seen to be a simple *quadratic function* of the unknown vector of weights, $\lambda_0 = (\lambda_{0i} : i = 1,..,n_0)'$, with known coefficients given by the variance-covariance parameters in (6.2.8) and (6.2.9). This means that one can actually minimize this function *explicitly* and determine the desired unknown weights. As shown in Appendix A2, such quadratic minimization problems are easily solved in terms of vector partial differentiation. But to illustrate the main ideas, it is instructive to consider a simple case not requiring vector analysis.

6.2.1 Simple Kriging with One Predictor

Consider the one-predictor case shown in Figure 6.2 below. Here the task is to predict $Y(s_0)$ on the basis of a *single* observation, $Y(s_1)$, at a nearby location, s_1 [so the relevant prediction set is simply $S(s_0) = \{s_1\}$].

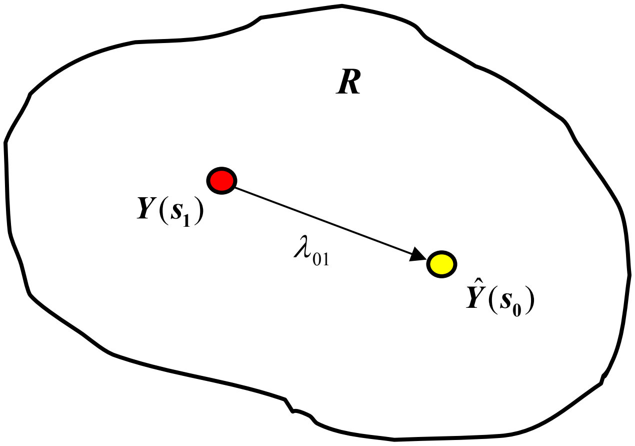


Figure 6.2 Single Predictor Case

While such "sparse" predictions are of little interest from a practical viewpoint, the derivation of a BLU predictor in this case is completely transparent. If we let

$$Y(s_i) = \mu + \varepsilon(s_i) = \mu + \varepsilon_i \ , \qquad i = 0,1 \ , \tag{6.2.17}$$

then by (6.2.3), the linear prediction hypothesis reduces to

(6.2.18) $$\hat{\varepsilon}_0 = \lambda_{01}\varepsilon_1 \ ,$$

so that the expression for mean squared error takes the simple form

(6.2.19) $$MSE(\lambda_{01}) = E\left[(\varepsilon_0 - \hat{\varepsilon}_0)^2\right] = E\left[(\varepsilon_0 - \lambda_{01}\varepsilon_1)^2\right]$$

$$= \sigma^2 - 2\lambda_{01}\sigma_{01} + \lambda_{01}^2\sigma_{11}$$

where in this case, $\sigma^2 = \text{var}(\varepsilon_0)$, $\sigma_{01} = \text{cov}(\varepsilon_0,\varepsilon_1)$, and $\sigma_{11} = \text{cov}(\varepsilon_1,\varepsilon_1) = \text{var}(\varepsilon_1)$. A representative plot of this simple quadratic function in λ_{01} is shown in Figure 6.2 below. Here it should be clear that mean squared error, $MSE(\lambda_{01})$, is minimized at the point, $\hat{\lambda}_{01}$, shown in the figure. [7]

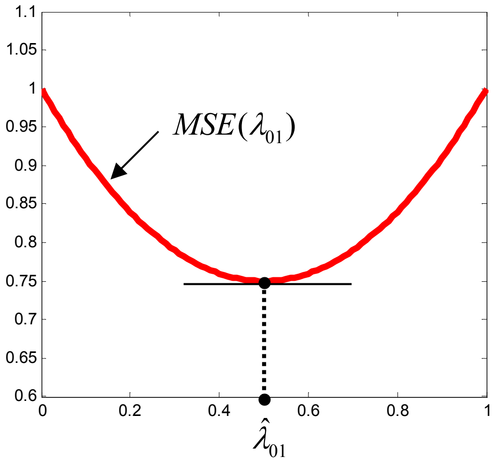


Figure 6.3 Optimal Weight Estimate

Mathematically, this minimum point, $\hat{\lambda}_{01}$, is characterized by the usual *first-order condition* that the derivative (slope) of $MSE(\lambda_{01})$ be zero (as shown in the figure), along with the *second-order condition* that that this slope be increasing, i.e., that the second derivative of $MSE(\lambda_{01})$ be positive. By differentiating (6.2.19) twice, we see that

(6.2.20) $$\tfrac{d}{d\lambda_{01}}MSE(\lambda_{01}) = -2\sigma_{01} + 2\sigma_{11}\lambda_{01} \quad \textit{and} \quad \tfrac{d^2}{d\lambda_{01}^2}MSE(\lambda_{01}) = 2\sigma_{11} > 0$$

Hence the second derivative is positive everywhere (as in the figure), and it follows that the unique optimal weight, $\hat{\lambda}_{01}$, is given by the solution of the first-order condition,

[7] In this example, $\sigma^2 = 1 = \sigma_{11}$ and $\sigma_{01} = 0.5$, so that the resulting optimal estimate in (6.2.21) is $\hat{\lambda}_{01} = 0.5$.

$$-2\sigma_{01} + 2\sigma_{11}\hat{\lambda}_{01} = 0 \;\;\Rightarrow\;\; \boxed{\hat{\lambda}_{01} = \sigma_{01}/\sigma_{11} = (\sigma_{11})^{-1}\sigma_{01}} \tag{6.2.21}$$

In this simple case, the interpretation of this optimal weight is also clear. Note first that if the covariance, $\sigma_{01} = \mathrm{cov}(\varepsilon_0, \varepsilon_1)$, between ε_0 and ε_1 is zero (so that these random variables are uncorrelated), then $\hat{\lambda}_{01} = 0$. In other words, if they are uncorrelated then ε_1 provides *no information* for predicting ε_0, and one can do no better than to ignore ε_1 altogether.[8] Moreover, as this covariance increases, ε_1 is expected to provide more information about ε_0, and the optimal weight on ε_1 increases. On the other hand, as the variance, $\sigma_{11} = \mathrm{var}(\varepsilon_1)$, of this predictor increases the optimal weight, $\hat{\lambda}_{01}$, decreases. This reflects the fact that a larger variance in ε_1 decreases its reliability as a predictor.

Finally, given this optimal weight, $\hat{\lambda}_{01}$, it then follows from (6.2.4) together with (6.2.18) that the resulting *optimal prediction*, $\hat{Y}(s_0)$, in Figure 6.2 is given by

$$\boxed{\hat{Y}(s_0) = \mu + \hat{\varepsilon}(s_0) = \mu + \hat{\lambda}_{01}\varepsilon_1 = \mu + (\sigma_{11})^{-1}\sigma_{01}\,\varepsilon_1} \tag{6.2.22}$$

As we shall see below, these results are mirrored in the general case of more than one predictor.

6.2.2 Simple Kriging with Many Predictors

Given the above results for a single predictor, we now generalize this setting to many predictors. The main objective of this section is to reformulate (6.2.16) in vector terms, and to use this formulation to extend expression (6.2.22) to the general the vector of optimal prediction weights, $\hat{\lambda}_0 = (\hat{\lambda}_{0i} : i = 1,..,n_0)'$, for Simple Kriging. A complete mathematical derivation of this result is given in Section A2.7.1 of Appendix A2. To begin with, let the full covariance matrix for $\varepsilon_0 = \varepsilon(s_0)$ together with its corresponding prediction set of error values, $\varepsilon_i = \varepsilon(s_i)$, be denoted by

$$C_0 = \left(\begin{array}{c|ccc} \sigma^2 & \sigma_{01} & \cdots & \sigma_{0n_0} \\ \hline \sigma_{01} & \sigma_{11} & \cdots & \sigma_{1n_0} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{0n_0} & \sigma_{n_01} & \cdots & \sigma_{n_0n_0} \end{array}\right) \tag{6.2.23}$$

[8] Note in particular that for the present case of multi-normally distributed errors, zero correlation is equivalent to *statistical independence*.

The partitioning shown in this matrix identifies its relevant components. Given the ordering, $i = 0,1,..,n_0$ of both rows and columns, the upper left hand corner denotes the variance of ε_0. The column vector below this value (and the row vector to the right) identifies the covariances of ε_0 with each predictor variable, ε_i, $i = 1,..,n_0$, and is now denoted by

$$c_0 = \begin{pmatrix} \sigma_{01} \\ \vdots \\ \sigma_{0n_0} \end{pmatrix} \tag{6.2.24}$$

Finally, the matrix to the lower right is the covariance matrix for all predictor variables, ε_i, $i = 1,..,n_0$, and is now denoted by

$$V_0 = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1n_0} \\ \vdots & \ddots & \vdots \\ \sigma_{n_0 1} & \cdots & \sigma_{n_0 n_0} \end{pmatrix} \tag{6.2.25}$$

In these terms, the full covariance matrix, C_0, can be given the compact form,

$$\boxed{C_0 = \begin{pmatrix} \sigma^2 & c_0' \\ c_0 & V_0 \end{pmatrix}} \tag{6.2.26}$$

It is the components of this partitioned matrix that form the basic elements of all kriging analysis. In particular, for the vector of unknown weights, $\lambda_0 = (\lambda_{0i} : i = 1,..,n_0)'$, the *mean squared error* function, $MSE(\lambda_0)$, in (6.2.16) can now be written in vector terms as follows

$$\boxed{MSE(\lambda_0) = \sigma^2 - 2c_0'\lambda_0 + \lambda_0' V_0 \lambda_0} \tag{6.2.27}$$

[which can be checked by applying (6.2.24) and (6.2.25) together with the rules of matrix multiplication]. By minimizing this function with respect to the components of λ_0, it is shown in expression (A2.7.20) of the Appendix that the *optimal weight vector*, $\hat{\lambda}_0 = (\hat{\lambda}_{0i} : i = 1,..,n_0)'$, is given by

$$\boxed{\hat{\lambda}_0 = V_0^{-1} c_0} \tag{6.2.28}$$

Hence, letting $\varepsilon = (\varepsilon_1,..,\varepsilon_{n_0})'$ denote the vector of *predictors* for ε_0, it follows that the *BLU predictor* of ε_0 is given by

(6.2.29) $$\boxed{\hat{\varepsilon}_0 \;=\; \hat{\lambda}_0{}'\varepsilon \;=\; c_0' V_0^{-1}\varepsilon}$$

and that [as a generalization of (6.2.22)] the corresponding *BLU predictor* of $Y(s_0)$ is given by

(6.2.30) $$\boxed{\hat{Y}(s_0) = \mu + \hat{\varepsilon}_0 = \mu + c_0' V_0^{-1}\varepsilon}$$

This predictor will generally be referred to as the *Simple Kriging predictor* of $Y(s_0)$.

6.2.3 Interpretation of Prediction Weights

By way of comparison with the single-predictor case above, note that in the present setting, this case takes the form,

(6.2.31) $$C_0 = \begin{pmatrix} \sigma^2 & \sigma_{01} \\ \sigma_{01} & \sigma_{11} \end{pmatrix}$$

so that by (6.2.24), $c_0 = (\sigma_{01})$ and $V_0 = (\sigma_{11})$. Hence it should now be clear that (6.2.21) is simply a special case of (6.2.29). Conversely, the simple interpretation of (6.2.21) can be (at least partially) extended to the present case. In particular, if the covariances between ε_0 and *all* predictor variables, ε_i, $i = 1,..,n_0$, are zero, i.e., if $c_0 = (0,..,0)'$, then by (6.2.28) we see that $\hat{\lambda}_0 = (0,..,0)'$. Hence in this case it is again clear that these predictors provide no information. More generally, suppose that all predictors are *uncorrelated*, i.e., the $\sigma_{ij} = 0$ for all $i, j = 1,..,n_0\,(i \neq j)$. Then V_0 reduces to a positive *diagonal* matrix with inverse given by the diagonal of reciprocals, i.e.,

(6.2.32) $$V_0 = \begin{pmatrix} \sigma_{11} & & \\ & \ddots & \\ & & \sigma_{n_0 n_0} \end{pmatrix} \;\Rightarrow\; V_0^{-1} = \begin{pmatrix} \sigma_{11}^{-1} & & \\ & \ddots & \\ & & \sigma_{n_0 n_0}^{-1} \end{pmatrix}$$

(which can be checked by simply multiplying to obtain $V_0 V_0^{-1} = I_{n_0}$). Hence by (6.2.24) and (6.2.29) we see that in this case all weights are the same as in the *single-predictor* case, i.e., that

(6.2.33) $$\hat{\lambda}_{0i} \;=\; (\sigma_{ii})^{-1}\sigma_{0i} \;,\; i = 1,..,n_0$$

So if all predictors are uncorrelated, then the contribution of each predictor, ε_i, to $\hat{\varepsilon}_0$ in (6.2.3) is the same as if it were a single predictor. In particular, it has zero contribution if and only if it is uncorrelated with ε_0.

However, if such predictors are to some degree *correlated*, then optimal prediction involves a rather complex interaction between the covariances, V_0 , among predictors and their covariances, c_0, with ε_0. In particular, if $\sigma_{0i} = 0$ then it is possible that interactions between both ε_0 and ε_i with other predictors may result in either positive or negative values for $\hat{\lambda}_{0i}$. As one illustration, suppose there are two predictors, $(\varepsilon_1, \varepsilon_2)$, with

$$(6.2.34) \qquad V_0 = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix} \quad , \quad c_0 = \begin{pmatrix} 0 \\ 1/2 \end{pmatrix}$$

so that ε_1 is uncorrelated with ε_0, but both have positive covariance (1/2) with ε_2. Then it can be verified in this case that

$$(6.2.35) \qquad \hat{\lambda}_0 = V_0^{-1} c_0 = \begin{pmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{pmatrix} \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} = \begin{pmatrix} -1/3 \\ 2/3 \end{pmatrix}$$

So even though all covariances (and hence correlations) are *nonnegative*, the optimal weight on ε_1 is actually *negative*. This shows that in the general case the interpretation of individual weights is much more complex. Indeed, it turns out in this case that the only quantity than can meaningfully be interpreted is the full *linear combination* of predictors in (6.2.29), i.e.,

$$(6.2.36) \qquad \hat{\varepsilon}_0 = \hat{\lambda}_0' \varepsilon = \sum_{i=1}^{n_0} \hat{\lambda}_{0i} \varepsilon_i$$

which in the above example, takes the form,

$$(6.2.37) \qquad \hat{\varepsilon}_0 = -(1/3)\varepsilon_1 + (2/3)\varepsilon_2$$

As expected, we see that ε_2 contributes positively to the prediction, $\hat{\varepsilon}_0$, and makes a more influential contribution than ε_1. But the negative influence of ε_1 is less intuitive. To gain further insight here, notice that by definition,

$$(6.2.38) \qquad \mathrm{cov}(\hat{\varepsilon}_0, \varepsilon_0) = E(\hat{\varepsilon}_0 \varepsilon_0) = E(\lambda_0' \varepsilon \varepsilon_0) = \lambda_0' E(\varepsilon \varepsilon_0) = \lambda_0' c_0 \ ,$$

and similarly that

$$(6.2.39) \qquad \mathrm{var}(\hat{\varepsilon}_0) = \mathrm{var}(\lambda_0' \varepsilon) = \lambda_0' \mathrm{cov}(\varepsilon) \lambda_0 = \lambda_0' V_0 \lambda_0$$

Hence mean squared error, $MSE(\lambda_0)$, can also be written as

(6.2.40) $$\boxed{MSE(\lambda_0) = \sigma^2 - 2\,\text{cov}(\hat{\varepsilon}_0,\varepsilon_0) + \text{var}(\hat{\varepsilon}_0)}$$

But since σ^2 is a constant not involving $\hat{\varepsilon}_0$, it becomes clear that minimization of $MSE(\lambda_0)$ essentially involves a tradeoff between the *covariance* of the predictor $\hat{\varepsilon}_0$ with ε_0 and the *variance* of the predictor itself. Indeed, this is the proper generalization of the original interpretation given in the single predictor case, where the relevant covariance and variance in that case were simply σ_{01} and σ_{11}, respectively. Moreover, the form of this tradeoff in (6.2.38) makes it clear that to minimize $MSE(\lambda_0)$, one needs a predictor $\hat{\varepsilon}_0$ with *positive* covariance, $\text{cov}(\hat{\varepsilon}_0,\varepsilon_0)$, as *large* as possible while at the same time having a variance, $\text{var}(\hat{\varepsilon}_0)$, as *small* as possible. It is from this viewpoint that the negativity of $\hat{\lambda}_{01}$ in (6.2.35) can be made clear. To see this observe that since $\sigma_{01} = 0$, covariance in this case takes the form

(6.2.41) $$\text{cov}(\hat{\varepsilon}_0,\varepsilon_0) = \lambda_0' c_0 = \lambda_{01}\sigma_{01} + \lambda_{02}\sigma_{02} = \lambda_{02}\sigma_{02}$$

But since $\sigma_{02} = 1/2 > 0$, it follows that this covariance can only be positive if $\lambda_{02} > 0$. Turning next to variance, observe that for any two-predictor case,

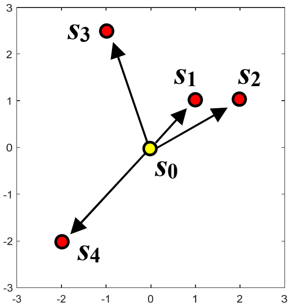
(6.2.42) $$\text{var}(\hat{\varepsilon}_0) = \lambda_0' V_0 \lambda_0 = (\lambda_{01}\,\lambda_{02})\begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}\begin{pmatrix} \lambda_{01} \\ \lambda_{02} \end{pmatrix}$$

$$= \lambda_{01}^2\sigma_{11} + 2\lambda_{01}\lambda_{02}\sigma_{12} + \lambda_{02}^2\sigma_{22}$$

$$= (\lambda_{01}^2\sigma_{11} + \lambda_{02}^2\sigma_{22}) + 2\lambda_{01}\lambda_{02}\sigma_{12}$$

But since the first term is always positive and since $\sigma_{12} = 1/2 > 0$, we see from the positivity of λ_{02} above that $\text{var}(\hat{\varepsilon}_0)$ can only be made small by requiring that $\lambda_{01} < 0$. In short, since ε_1 has no effect on the correlation of the predictor, $\hat{\varepsilon}_0$, with ε_0, its best use for prediction is to *shrink* the variance of $\hat{\varepsilon}_0$ by setting $\lambda_{01} < 0$.

Before using these kriging weights for prediction, it is of natural interest to consider their *spatial* nature. In particular, referring again to our initial illustration in Figure 6.1, it would seem reasonable that points, s_i, closer to s_0 should have larger weight, $\hat{\lambda}_{0i}$. In particular, if invoke the "standard covariogram" assumption of Figure 4.1 in Section 4, namely that covariances *decrease* with distance, then points further away should contribute less to the prediction of $Y(s_0)$. But for Simple Kriging predictors this is simply not the case. One simple example is shown in Figure 6.4 below:



point	s_1	s_2	s_3	s_4
distance	1.41	2.24	2.69	2.83
weight	.53	.05	.13	.30
rank	1	4	3	2

Figure 6.4 Weighting versus Distance

Here points (s_1,s_2,s_3,s_4) are ordered in terms of their distance from prediction point, s_0, as shown in the second row of the table.[9] To calculate weights in this case, a simple exponential covariogram was used.[10] So in this *spatially stationary* setting, covariances are strictly decreasing in distance. Hence the key point to notice is that the kriging weights $(\hat{\lambda}_{01},\hat{\lambda}_{02},\hat{\lambda}_{03},\hat{\lambda}_{04})$ in the third row of the table are *not* decreasing in distance. Indeed the second closest point, s_2, here is the *least* influential of the four (as depicted by the ranking of weights in the last row of the table). Notice that since s_1 and s_2 are closer to each other than to s_0, and since distances are in this case inversely related to correlations,[11] the errors $\varepsilon(s_1)$ and $\varepsilon(s_2)$ are more correlated with each other than either is to $\varepsilon(s_0)$. So it might be argued here that $\varepsilon(s_2)$ is adding little prediction information for $\varepsilon(s_0)$ beyond that in $\varepsilon(s_1)$. But notice that the influence of points s_3 and s_4 is also reversed, and that no such relative correlation effects are present here. So even in this simple monotone-covariance setting, it is difficult to draw general conclusions about the exact relation between distance and kriging weights.

While these illustrations are necessarily selective in nature, they do serve to emphasize the complexity of possible interaction effects in MMSE prediction. Given this development of Simple Kriging predictors, we turn now to the single most important justification for such stochastic predictors, namely the construction of meaningful *prediction intervals* for possible realized values of $Y(s_0)$.

6.2.4 Construction of Prediction Intervals

Note that up to this point we have relied only on knowledge of the means and covariances of the spatial error process $\{\varepsilon(s): s \in R\}$ to derive optimal predictors. But to develop prediction intervals for these errors, we must now make explicit use of the

[9] The actual point coordinates are $s_0 = (0,0)$, $s_1 = (1,1)$, $s_2 = (2,1)$, $s_3 = (-1,2.5)$ and $s_4 = (-2,-2)$.

[10] With respect to the notation in expression (4.6.6) of Section 4, the range, sill, and nugget parameters used were $(r = 30,\ s = 1,\ a = 0)$.

[11] Recall from (3.3.13) in Section 3 that *spatially stationary* correlations are proportional to covariances.

distributional assumption of *multi-normality* . In terms of (6.2.26) this assumption implies in particular that for any prediction site, $s_0 \in R$, and corresponding prediction set, $S(s_0) = \{s_i : i = 1,..,n_0\}$, the random (column) vector of errors,

$$(6.2.43) \qquad \begin{pmatrix} \varepsilon(s_0) \\ \varepsilon(s_1) \\ \vdots \\ \varepsilon(s_{n_0}) \end{pmatrix} = \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_{n_0} \end{pmatrix} = \begin{pmatrix} \varepsilon_0 \\ \varepsilon \end{pmatrix}$$

is *multi-normally distributed* as[12]

$$(6.2.44) \qquad \begin{pmatrix} \varepsilon_0 \\ \varepsilon \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0_{n_0} \end{pmatrix}, \begin{pmatrix} \sigma^2 & c_0' \\ c_0 & V_0 \end{pmatrix}\right]$$

Our primary application of this distribution will be to derive the distribution of the associated prediction error in (6.1.3), which we now write simply as $e_0 = e(s_0)$. But before proceeding it is important to emphasize once again the *distinction* between ε_0 and e_0 . Recall that ε_0 is the deviation of $Y(s_0)$ about its mean [$Y(s_0) = \mu + \varepsilon_0$], while e_0 is the difference between $Y(s_0)$ and its predicted value [$e_0 = Y(s_0) - \hat{Y}(s_0)$].

To derive the distribution of e_0 from that of random error vector, $(\varepsilon_0, \varepsilon)$,[13] we begin by using (6.2.1), (6.2.4) and (6.2.29) to write e_0 in terms of $(\varepsilon_0, \varepsilon)$ as follows,

$$(6.2.45) \qquad e_0 = Y(s_0) - \hat{Y}(s_0) = \varepsilon(s_0) - \hat{\varepsilon}(s_0) = \varepsilon_0 - \hat{\varepsilon}_0$$

$$= \varepsilon_0 - \hat{\lambda}_0'\varepsilon = (1, -\hat{\lambda}_0')\begin{pmatrix} \varepsilon_0 \\ \varepsilon \end{pmatrix}$$

Hence e_0 is seen to be a *linear compound* of $(\varepsilon_0, \varepsilon)$. This, together with the multi-normality of $(\varepsilon_0, \varepsilon)$, implies at once from the Invariance Theorem in Section 3.2.2 above that e_0 must also be *normally distributed*. Moreover, since we have already seen in (6.1.4) that $E(e_0) = 0$, it follows that if we can calculate the *variance* of e_0 , then its distribution will be completely determined.

[12] Here 0_{n_0} denotes the n_0 -dimensional *zero vector*.

[13] Note that technically this vector should be written inline as $(\varepsilon_0, \varepsilon')'$ to indicate that it is column vector. But for sake of visual clarity, we write simply $(\varepsilon_0, \varepsilon)$.

In view of the importance of this particular variance, we derive it in two ways. First we derive it directly from the covariance-transformation identity in (3.2.21) of Section 3. In particular, for any linear compound, $a'X$, of a random vector, X, with covariance matrix, Σ, it follows at once from (3.2.21) [with $A = a'$] that

(6.2.46) $$\mathrm{var}(a'X) = a'\Sigma a$$

Hence by letting

(6.2.47) $$X = \begin{pmatrix} \varepsilon_0 \\ \varepsilon \end{pmatrix} \;,\; a = \begin{pmatrix} 1 \\ -\hat{\lambda}_0 \end{pmatrix} \;,\; \Sigma = \begin{pmatrix} \sigma^2 & c_0' \\ c_0 & V_0 \end{pmatrix}$$

it follows from (6.2.45) and (6.2.46) that

(6.2.48) $$\mathrm{var}(e_0) = (1\,, -\hat{\lambda}_0{}') \begin{pmatrix} \sigma^2 & c_0' \\ c_0 & V_0 \end{pmatrix} \begin{pmatrix} 1 \\ -\hat{\lambda}_0 \end{pmatrix}$$

$$= (1\,, -\hat{\lambda}_0{}') \begin{pmatrix} \sigma^2 - c_0'\hat{\lambda}_0 \\ c_0 - V_0\hat{\lambda}_0 \end{pmatrix} = (\sigma^2 - c_0'\hat{\lambda}_0) - \hat{\lambda}_0{}'c_0 + \hat{\lambda}_0{}'V_0\hat{\lambda}_0$$

But since for any vectors, $x' = (x_1,..,x_n)$ and $y' = (y_1,..,y_n)$, it must be true that $x'y = \Sigma_i x_i y_i = \Sigma_i y_i x_i = y'x$, we see that (6.2.48) can be reduced to

(6.2.49) $$\mathrm{var}(e_0) = \sigma^2 - 2c_0'\hat{\lambda}_0 + \hat{\lambda}_0{}'V_0\hat{\lambda}_0$$

The form of the right hand side should look familiar. In particular, the representation of mean squared error, $MSE(\lambda_0)$, in (6.2.27) now yields the identity,

(6.2.50) $$\boxed{\mathrm{var}(e_0) = MSE(\hat{\lambda}_0)}$$

This relation is no coincidence. Indeed, recall from (6.1.5) that for any unbiased predictor, $\hat{\varepsilon}_0$,

(6.2.51) $$E[(\varepsilon_0 - \hat{\varepsilon}_0)^2] = E(e_0^2) = \mathrm{var}(e_0)\,,$$

so that its mean squared error is identically equal to the variance of its associated prediction error. So for the optimal predictor in particular, this variance must be given by the mean squared error evaluated at $\hat{\lambda}_0$. Indeed we could have derived (6.2.49) through this line of reasoning. Hence the direct derivation in (6.2.45) through (6.2.48) offers an instructive confirmation of this fact.

To complete this derivation, it suffices to substitute the solution for $\hat{\lambda}_0$ in (6.2.28) [i.e., $\hat{\lambda}_0 = V_0^{-1}c_0$] into (6.2.49) to obtain,

(6.2.52)
$$\text{var}(e_0) = \sigma^2 - 2c_0'[V_0^{-1}c_0] + [V_0^{-1}c_0]'V_0[V_0^{-1}c_0]$$
$$= \sigma^2 - 2c_0'V_0^{-1}c_0 + c_0'V_0^{-1}(V_0V_0^{-1})c_0$$
$$= \sigma^2 - 2c_0'V_0^{-1}c_0 + c_0'V_0^{-1}(I_{n_0})c_0$$
$$= \sigma^2 - 2c_0'V_0^{-1}c_0 + c_0'V_0^{-1}c_0$$

By combining the last two terms, we obtain the final expression for *prediction error variance* (also called *Kriging variance*),

(6.2.53)
$$\boxed{\sigma_0^2 \;=\; \text{var}(e_0) \;=\; \sigma^2 - \; c_0'V_0^{-1}c_0}$$

where we have now introduced the simplifying notation (σ_0^2) for this important quantity. While this expression for σ_0^2 is most useful for computational purposes, it is of interest to develop an alternative expression that is easier to interpret. To do so, if we now use the simplifying notation, $Y_0 = Y(s_0)$, for the variable to be predicted at location $s_0 \in R$, then [as a consequence of (3.2.21)] it follows that the first term in (6.2.53) is simply the variance of Y_0, since

(6.2.54)
$$\text{var}(Y_0) = \text{var}(\mu + \varepsilon_0) = \text{var}(\varepsilon_0) = \sigma^2$$

Similarly, if we also represent the corresponding *predictor* in (6.2.30) by $\hat{Y}_0 = \hat{Y}(s_0)$, then the second term in (6.2.53) turns out to be precisely the variance of $\hat{Y}_0$. To see this, note simply from (6.2.3) together with (6.2.5) and (3.2.21) that

(6.2.55)
$$\text{var}(\hat{Y}_0) = \text{var}(\mu + c_0'V_0^{-1}\varepsilon) = \text{var}(c_0'V_0^{-1}\varepsilon)$$
$$= c_0'V_0^{-1}\,\text{cov}(\varepsilon)V_0^{-1}c_0 = c_0'V_0^{-1}(V_0)V_0^{-1}c_0$$
$$= c_0'V_0^{-1}(V_0\,V_0^{-1})c_0 = c_0'V_0^{-1}c_0$$

So the prediction error variance in (6.2.53) can be equivalently rewritten as

(6.2.56)
$$\boxed{\sigma_0^2 = \text{var}(Y_0) - \text{var}(\hat{Y}_0)}$$

In these terms, it is clear that prediction error variance, σ_0^2 is *smaller* that than the original variance of Y_0. Moreover, the amount of this reduction is seen to be precisely the variance "explained" by the predictor, $\hat{Y}_0$. Indeed, it can be argued that this reduction in variance is the *fundamental rationale* for kriging predictions, often referred to as "borrowing strength from neighbors".

Given this expression for prediction error variance, it follows at once from the arguments above the prediction error, e_0, must be *normally distributed* as

$$\boxed{e_0 \sim N(0,\sigma_0^2)} \qquad (6.2.57)$$

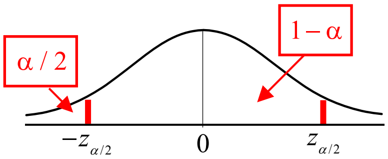
Hence the task remaining is to use this normal distribution of $e_0\ [= Y_0 - \hat{Y}_0]$ to construct prediction intervals for Y_0 in terms of $\hat{Y}_0$ and σ_0^2. To do so, we first recall from Sections 3.1.1 and 3.1.2 above that the standardization of e_0 must be distributed as $N(0,1)$. In particular, since the mean of e_0 is zero, and since it standard deviation of e_0 is given from (6.2.53) by $\sqrt{\text{var}(e_0)} = \sigma_0$, it follows that

$$\boxed{\frac{Y_0 - \hat{Y}_0}{\sigma_0} = \frac{e_0}{\sigma_0} \sim N(0,1)} \qquad (6.2.58)$$

Hence it now becomes clear that, together with $\hat{Y}_0$, the key distributional parameter is the standard deviation, σ_0, of e_0, which is usually designated as the *standard error of prediction.* Indeed, as will be seen below, the fundamental outputs of all kriging software are precisely estimates of the *kriging prediction*, $\hat{Y}_0$, and *standard error of prediction*, σ_0, at all relevant prediction locations, s_0.

To construct prediction intervals for Y_0 based on (6.2.52), we proceed in a manner paralleling the two-tailed Clark-even test procedure in Section 3.2.2 of Part I. In particular, by recalling from (3.1.32) that Φ denotes the cumulative distribution function for $N(0,1)$, and that for any probability, α, the α*-critical value*, z_α, is defined by $\Phi(-z_\alpha) = \alpha$ [as in the figure below for $\alpha/2$], it follows that

$$\boxed{\Pr\left(-z_{\alpha/2} \le \frac{Y_0 - \hat{Y}_0}{\sigma_0} \le z_{\alpha/2}\right) = 1-\alpha} \qquad (6.2.59)$$



But since the following events are equivalent:

$$(6.2.60)\qquad -z_{\alpha/2} \le \frac{Y_0 - \hat{Y}_0}{\sigma_0} \le z_{\alpha/2} \;\Leftrightarrow\; -\sigma_0 z_{\alpha/2} \le Y_0 - \hat{Y}_0 \le \sigma_0 z_{\alpha/2}$$

$$\Leftrightarrow\; \hat{Y}_0 - \sigma_0 z_{\alpha/2} \le Y_0 \le \hat{Y}_0 + \sigma_0 z_{\alpha/2}$$

it follows that their probabilities must be the same, and hence from (6.2.59) that,

$$(6.2.61)\qquad \boxed{\Pr\left(\hat{Y}_0 - \sigma_0 z_{\alpha/2} \le Y_0 \le \hat{Y}_0 + \sigma_0 z_{\alpha/2}\right) = 1-\alpha}$$

In other words, the probability that the value of Y_0 lies between $\hat{Y}_0 - \sigma_0 z_{\alpha/2}$ and $\hat{Y}_0 + \sigma_0 z_{\alpha/2}$ is $1-\alpha$. In terms of confidence levels, this means that we be $100(1-\alpha)\%$ confident that Y_0 lies in the *prediction interval*,

$$(6.2.62)\qquad \left[\hat{Y}_0 - \sigma_0 z_{\alpha/2}\,,\, \hat{Y}_0 + \sigma_0 z_{\alpha/2}\right] = \left[\hat{Y}_0 \pm \sigma_0 z_{\alpha/2}\right]$$

The single most common instance of (6.2.62) is for the case, $\alpha = 0.05$, with corresponding critical value $z_{\alpha/2} = z_{.025} = 1.96$. In this case, one can thus be 95% confident that Y_0 lies in the prediction interval,

$$(6.2.63)\qquad \boxed{\left[\hat{Y}_0 - (1.96)\sigma_0\,,\, \hat{Y}_0 + (1.96)\sigma_0\right] = \left[\hat{Y}_0 \pm (1.96)\sigma_0\right]}$$

As with all statistical confidence statements, the phrase “95% confident” here means that if we were able carry out this same prediction procedure many times (i.e., to take many random samples from the *joint distribution* of Y_0 and its kriging prediction, $\hat{Y}_0$) then we would expect the realized values of Y_0 to lie in the corresponding realized intervals $[\hat{Y}_0 \pm (1.96)\sigma_0]$ about 95% of the time.

Finally it should again be emphasized that it is the ability to make confidence statements of this type that distinguishes stochastic prediction methods from the deterministic methods of spatial interpolation developed in Section 5.

6.2.5 Implementation of Simple Kriging Models

Given the theoretical development of Simple Kriging above, the task remaining is to make this procedure operational. But before doing so, it should again be emphasized, as in Section 6.1.2 above, that from a practical viewpoint, *Ordinary Kriging* is almost always used in empirical situations where Simple Kriging is relevant. Hence the main relevance of this procedure for our purposes is to develop as many of the basic concepts as possible within this simple setting. It should also be noted that this Simple Kriging procedure is one of the options available in the **Geostatistical Analyst** extension of ARCMAP. So we will be able to use the implementation developed here to illustrate most of the operational procedures involved in the use of this software. With this in mind, we now proceed to operationalize Simple Kriging through a series of procedural steps. This will be followed in Section 6.2.6 below with an application of this procedure.

In the following development, we again postulate that the values of some variable *Y* defined over a relevant region *R* can be modeled by a spatial stochastic process, $\{Y(s) = \mu + \varepsilon(s) : s \in R\}$, with constant mean, μ. In addition, we assume the existence of a given set of *n* observations (data points), $\{y_i = y(s_i) : i = 1,..,n\}$ in *R*, where of course each data point, y_i, is taken to be a realization of the corresponding random variable, $Y_i = Y(s_i)$ in this spatial stochastic process. Also, for purposes of illustration, we shall again consider the problem of predicting, $Y(s_0)$, at a single given location, $s_0 \in R$, with respect to a given prediction set, $S(s_0) = \{s_1,..,s_{n_0}\} \subseteq \{s_1,..,s_n\}$. Within this framework, we can operationalize the Simple Kriging model as follows:

Step 1. Estimation of the Mean

Recall from the assumption in (6.1.6) that our first task is to produce an *estimate* of the mean, μ, *outside* the Simple Kriging model. Here the obvious choice is just to use the *sample mean* of the given data, i.e.,

$$(6.2.62) \qquad \hat{\mu} = \overline{y}_n = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} y(s_i) = \tfrac{1}{n}\sum_{i=1}^{n} y_i$$

One attractive feature of this estimate is that it is always *unbiased* since

$$(6.2.63) \qquad E(\hat{\mu}) = E\left(\tfrac{1}{n}\textstyle\sum_{i=1}^{n} Y_i\right) = \tfrac{1}{n}\sum_{i=1}^{n} E(Y_i) = \tfrac{1}{n}\sum_{i=1}^{n} \mu = \mu$$

So even though these random variables are spatially correlated, this has no effect on unbiasedness. What spatial correlation does imply is that the *variance* of this estimator is much larger than that of the classical sample mean under independence. We shall return to this issue in the development of Ordinary Kriging in Section 6.3 below.

Step 2. Estimation of Covariances

Recall next from assumption (6.1.7) that the covariances, $\text{cov}[\varepsilon(s),\varepsilon(s')]$ are assumed to be given for all locations, $s,s' \in R$. But we must of course provide some prior estimates of these covariances. This was in fact one of the primary motivations for the assumption of *covariance stationarity* in Section 3.3.2 above. Hence we now invoke this assumption in order to estimate spatial covariances in a manner that accounts for spatial correlation effects. Recall also from Section 4.10.1 that, unlike the mean above, the classical estimate of covariance is *biased* in the presence of spatial correlation. So our estimation procedure here will always start with variograms rather than covariograms. Fortunately, this basic estimation procedure is exactly the same as that used for Ordinary Kriging, and indeed, for all more advanced kriging models. So it is worthwhile to develop this procedure in detail here.

To do so, we begin by recalling from (3.3.7) and (3.3.11) in Section 3 that under covariance stationarity, all covariances can be summarized by a covariogram, $C(h)$. As emphasized in Section 4, this is best estimated by first estimating a *variogram*, $\gamma(h;r,s,a)$ with parameters, $r = \textit{range}$, $s = \textit{sill}$, and $a = \textit{nugget}$. Since the common variance, $C(0) = \sigma^2$, is precisely the sill parameter, s, one can then obtain the desired covariogram from the identity in (4.1.7) of Section 4, namely[14]

$$(6.2.64) \qquad C(h) = \sigma^2 - \gamma(h) = s - \gamma(h;r,s,a)$$

Hence, the estimation procedure starts by using the MATLAB program, **var_spher_plot**, together with the full sample data set above to obtain estimates, $(\hat{r},\hat{s},\hat{a})$, of the spherical variogram parameters. The estimated spherical variogram, $\gamma(h;\hat{r},\hat{s},\hat{a})$, is then used together with (6.2.64) to obtain an estimate, $\hat{C}(h)$, of the desired covariogram as follows:

$$(6.2.65) \qquad \hat{C}(h) = \hat{s} - \gamma(h;\hat{r},\hat{s},\hat{a})$$

Recall that for any pair of point, $s,s' \in R$ separated by distance, $\|s-s'\| = h$ the quantity, $\hat{C}(h)$, then yields an estimate of $\text{cov}[\varepsilon(s),\varepsilon(s')]$, i.e.,

$$(6.2.66) \qquad \widehat{\text{cov}}[\varepsilon(s),\varepsilon(s')] = \hat{C}(\|s-s'\|)$$

Using this identity, we can then estimate the full covariance matrix, C_0, relevant for prediction at s_0 [as in (6.2.23) above]. In particular, if we let $d_{ij} = \|s_i - s_j\|$ for each pair of points, $s_i,s_j \in \{s_0,s_1,..,s_{n_0}\}$, and [as instances of (6.2.66)] set

$$(6.2.67) \qquad \hat{\sigma}_{ij} = \hat{C}(d_{ij})$$

[14] Again, remember *not* to confuse the symbol, s, for "sill" with points, $s = (s_1,s_2) \in R$.

then we immediately obtain the following estimate, $\hat{C}_0$, of C_0,

$$(6.2.68) \qquad \hat{C}_0 = \left(\begin{array}{c|ccc} \hat{\sigma}^2 & \hat{c}_{01} & \cdots & \hat{c}_{0n_0} \\ \hline \hat{c}_{10} & \hat{c}_{11} & \cdots & \hat{c}_{1n_0} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{c}_{n_0 0} & \hat{c}_{n_0 1} & \cdots & \hat{c}_{n_0 n_0} \end{array}\right) = \begin{pmatrix} \hat{\sigma}^2 & \hat{c}_0{}' \\ \hat{c}_0 & \hat{V}_0 \end{pmatrix}$$

Note in particular, that the common variance, σ^2, of all random variables is again estimated by the sill, since

$$(6.2.69) \qquad \hat{\sigma}^2 = \hat{C}(0) = \hat{s}$$

Step 3. Estimation of Kriging Predictions

Finally, given these parameter estimates, we are ready to estimate the Simple Kriging prediction, $\hat{Y}(s_0)$, of $Y(s_0)$. To do so, begin by recalling that that the *deviation error*, $\varepsilon_i = y_i - \mu$, at each data point, $i = 1,..,n_0$, can now be estimated in terms of (6.2.62) by

$$(6.2.70) \qquad \hat{\varepsilon}_i = y_i - \hat{\mu}$$

So if we now designate the corresponding estimate of the deviation predictors, $\varepsilon = (\varepsilon_1,..,\varepsilon_{n_0})'$ for $\varepsilon_0 = \varepsilon(s_0)$ by

$$(6.2.71) \qquad \hat{\varepsilon} = [\hat{\varepsilon}_i : s_i \in S(s_0)]' = (\hat{\varepsilon}_1,..,\hat{\varepsilon}_{n_0})'$$

then it follows from (6.2.29) that *Simple Kriging prediction* of ε_0 is given by

$$(6.2.72) \qquad \hat{\varepsilon}_0 = \hat{c}_0' \hat{V}_0^{-1} \hat{\varepsilon}$$

Finally, by using (6.2.30) together with these estimates, it follows that the *Simple Kriging prediction* of $Y_0 = Y(s_0)$ is given by[15]

$$(6.2.73) \qquad \hat{Y}_0 = \hat{Y}(s_0) = \hat{\mu} + \hat{c}_0' \hat{V}_0^{-1} \hat{\varepsilon}$$

To complete the implementation of Simple Kriging, it remains only to estimate the corresponding *prediction error variance* (or *Kriging variance*) in (6.2.53) by

[15] Here it should be noted that for simplicity, we have used the same notation for the theoretical and estimated Simple Kriging prediction, $\hat{Y}_0$ (and $\hat{\varepsilon}_0$).

(6.2.74) $$\hat{\sigma}_0^2 = \hat{s} - \hat{c}_0{}'\hat{V}_0^{-1}\hat{c}_0$$

and take its square root,

(6.2.75) $$\hat{\sigma}_0 = \sqrt{\hat{s} - \hat{c}_0{}'\hat{V}_0^{-1}\hat{c}_0}$$

to be the relevant estimate of the *standard error of prediction* at location s_0. The pair of values $(\hat{Y}_0, \hat{\sigma}_0)$ can then be used as in (6.2.61) to estimate the (default) 95% prediction interval for Y_0, namely,

(6.2.76) $$[\hat{Y}_0 \pm (1.96)\hat{\sigma}_0]$$

One final comment should be made about these estimates. In the theoretical development of Section 6.2.2, the predictors $\hat{\varepsilon}_0$ and $\hat{Y}_0$ were derived as *Best Linear Unbiased* (*BLU*) predictors. This is only accurate if the *true* mean, μ, and covariances, C_0, are known – which is of course almost never the case. So to be accurate, the above values $\hat{\varepsilon}_0$ and $\hat{Y}_0$ are in fact only *estimates* of BLU predictors. This distinction is often formalized by designating them as *Empirical-BLU predictors.* Similarly, as with all prediction intervals or confidence intervals based on *estimated* parameters, the variation of these parameter estimates is of course *not* accounted for in these intervals themselves. So again, a more precise statement would be to designate (6.2.76) as an *estimated* 95% prediction interval.

6.2.6 An Example of Simple Kriging

Given the estimation procedure above, we now illustrate an application of *Simple Kriging* in terms of the Vancouver Nickel data in Section 4.9 above. But before developing this example, it is important to emphasize that the underlying *normality* assumption on all spatially-dependent random effects, $\varepsilon(s)$, is crucial for the estimation of meaningful prediction intervals. Moreover, since these random effects are not directly observable, this distributional assumption can only be checked indirectly. But by *assuming* that there are no global trends (as in Simple and Ordinary Kriging), it should be clear from the identity

(6.2.77) $$Y(s_i) = \mu + \varepsilon(s_i), \quad i = 1,..,n$$

that these random effects differ from the observed data, $\{y(s_i) : i = 1,..,n\}$, only by a (possibly unobserved) constant, μ. Moreover, since the variance, $\sigma^2 = \text{var}[\varepsilon(s_i)]$ $= \text{var}[Y(s_i)]$ is constant for all *covariance-stationary* processes, it follows that under this additional assumption, the *marginal* distributions must be the same for all Y data, namely

(6.2.78) $$Y(s_i) \sim N(\mu, \sigma^2) \ , \ i = 1,..,n$$

So even though these are not independent samples from this common distribution, it is still reasonable to expect that the histogram of this data should look approximately normal. This motivates the following simple test of normality.

Normal Quantile Plots and Transformations

A very simple and appealing test of normality is available in JMP, known as *Normal Quantile Plots* (also called *Normal Probability Plots*). The main appeal of this test is that it is *graphical*, and in addition, provides global information about possible failures of normality. The idea is very simple. Given a set of data $(y_1,...,y_n)$ from an unknown distribution, one first reorders the data (if necessary) so that $y_1 \le y_2 \le \cdots \le y_n$, and then standardizes it by subtracting the sample mean, $\overline{y}_n = \frac{1}{n}\Sigma_{i=1}^n y_i$, and dividing by the sample standard deviation, $s_n = \left[\frac{1}{n-1}\Sigma_{i=1}^n (y_i - \overline{y}_n)^2\right]^{1/2}$, to obtain:

(6.2.79) $$z_i = \frac{y_i - \overline{y}_n}{s_n} \ , \quad i = 1,..,n$$

Now if $(y_1,...,y_n)$ were coming from a normal distribution, then $(z_1,...,z_n)$ should be *approximately* distributed as, $Z_i \sim N(0,1), i = 1,..,n$ [we are using only *estimated* means and standard deviations here]. So for an independent sample $(Z_1,..,Z_n)$ of size n from $N(0,1)$, if we compute the *theoretical* expected values, $\phi_i = E(Z_i), \ i = 1,..,n$, then we would expect on average that the observed values z_i in (6.2.79) should be reasonably close to their expected values, ϕ_i. This in turn implies that if plot z_i against ϕ_i, the points should like close to the 45° line. This is illustrated in Figure 6.5 below, where a sample of size $n = 100$ has been simulated in JMP (using **Formula → Random → Random Normal**).

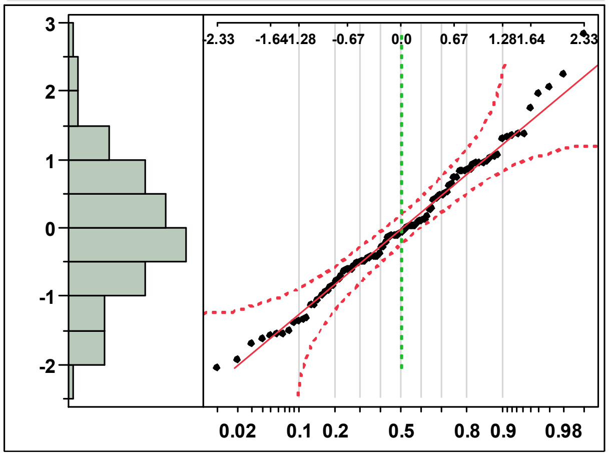


Figure 6.5 Normal Quantile Plot

The values on the vertical axis are exactly the z_i values together with their histogram shown on the left. The *Normal Quantile Plot* is displayed on the right (using the procedure detailed in Assignment 4). The values on the horizontal axis at the *top* of the figure are precisely the expected values, ϕ_i, for each z_i , $i = 1,..,100$.[16] Here it is clear that all point pairs are indeed close to the 45° line (shown in red). The dashed lines denote 95% *probability intervals* on the realized values z_i, so that if the sample were normal (as in this simulation) then each dot should lie between these bands about 95% of the time.[17] For example, the middle sample value, z_{50}, with expected value, $\phi_{50} = E(Z_{50}) \approx 0$, should lie in the interval between these two bands on the vertical green center line about 95% of the time. So this plot provides compelling evidence that this sample is indeed coming from a normal distribution.

We now apply this tool to the Nickel data, as shown in Figure 6.6 below. For ease of comparison with Figure 6.7, the histogram and corresponding normal quantile plot are show using the horizontal display option[18]. (The only difference here is that the Normal Quantile Plot is now above the histogram, with ϕ_i values on the vertical axis to the right.) Since most data observed in practice is *nonnegative* (i.e., is truncated at zero), the corresponding histograms tend to be "skewed to the right", as illustrated by this Nickel data.

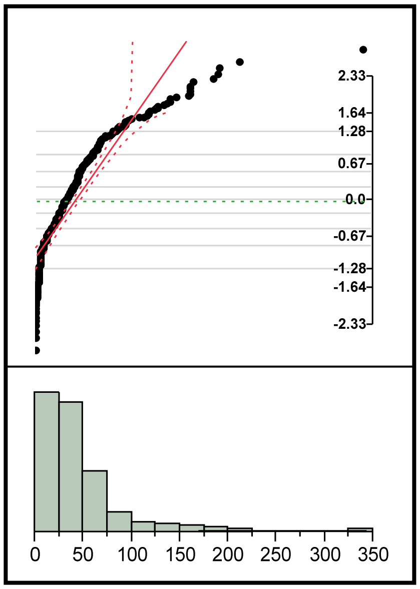


Figure 6.6. Nickel Data

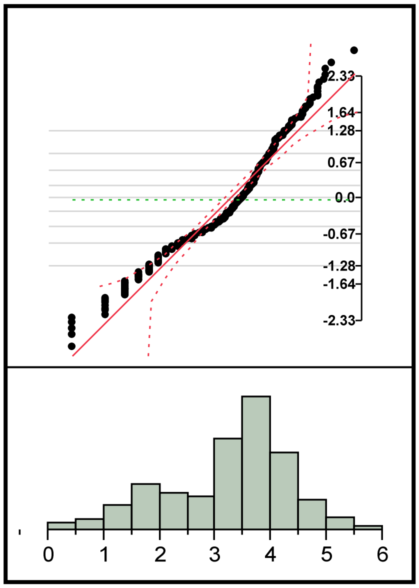


Figure 6.7. Log-Nickel Data

[16] The values on the bottom horizontal axis are the associated cumulative probabilities, so that "0" on the top corresponds to "$\Phi(0) = .5$" on the bottom.

[17] Note that such probability intervals are *different* from confidence intervals. In particular, their end points are fixed. Note also that these (Lilliefors) probability bounds actually account for the estimated mean and standard-deviation valued used [for more information, Google "Lilliefors test"].

[18] Right click on the label bar above the histogram and select **Display Options → Horizontal Layout**.

The degree of non-normality of this data is even more evident from the Normal Quantile Plot. Here the mid-range values are well above the 45° line (slightly distorted in this plot), indicating that there is "too much probability mass to the left of center" relative to the normal distribution. Hence it is difficult to krige this data directly, since the corresponding prediction intervals would have little validity.

However, if this data is transformed to *natural logs*, then the familiar "bell shaped" curve starts to appear, as seen in Figure 6.7 above. What is happening is that the log transformation "shrinks" the upper range of the distribution (above value one) and "expands" the lower range (below value one). While other transformations are possible here, (such as taking square roots rather than logs), the log transformation is by far the most common. It is also used for regression residuals, as we shall see in later sections.

To perform this log transformation in MATLAB, we start with original data set, **nickel**, the MATLAB file, **nickel.mat**. Next we replace the data column, **nickel(:,3)** with log data, and save as **log_nickel** using the command:

```
>> log_nickel = [nickel(:,1:2),log(nickel(:,3))];
```

This makes a new matrix consisting of the first two columns of nickel and the log of the third column.[19]

Estimation of the Spherical Variogram and Covariogram

Recall from Section 4.9.2 that the variogram and covariogram were estimated for the **nickel** data, as in Figures 4.22 and 4.23, respectively. We now redo this procedure for the **log_nickel** data in order to obtain initial covariance inputs for Kriging this data. To estimate a spherical variogram we start with the default value of maxdist:

```
>> var_spher_plot(log_nickel);
```

and obtain the results shown in Figure 6.8 below:

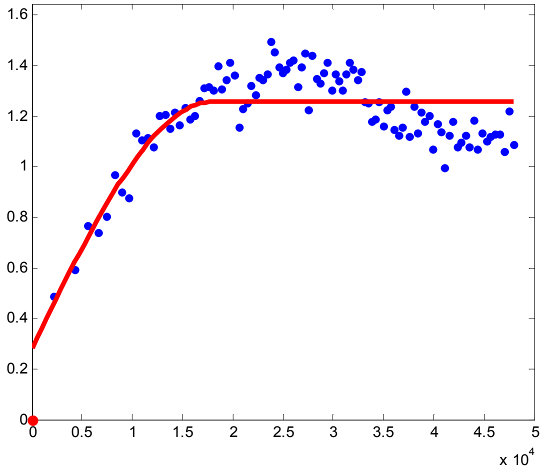


Figure 6.8. Log Nickel Variogram

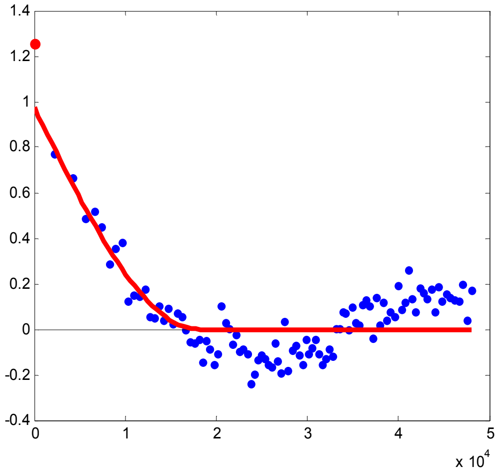


Figure 6.9. Log Nickel Covariogram

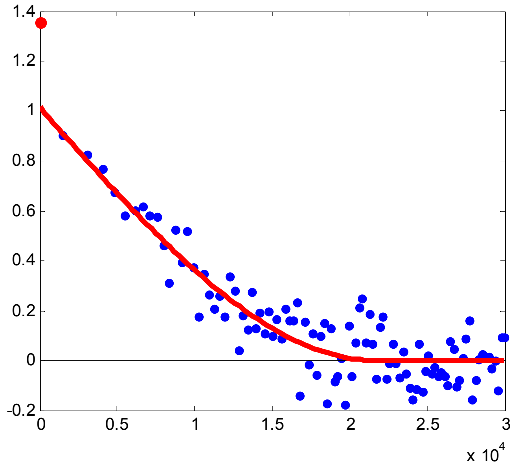
[19] Note that the **log** command uses *natural logs* by default. Logs to the base 10 are obtained with the command, **log10**.

The corresponding covariogram estimate is on the right in Figure 6.9. Here we again see a wave effect which is qualitatively very similar to that in Figure 4.22 for the raw **nickel** data. Here the reported maxdist value is 48,204. However, it appears that up to about 30,000 meters the empirical variogram is reasonably consistent with a classical spherical variogram. Hence to capture this range, we now rerun **var_spher_plot** with this specified maxdist value as follows:

```
>> opts.maxdist = 30000;
```

```
>> OUT = var_spher_plot(log_nickel,opts);
```

The new covariogram is plotted in Figure 6.10 below, and is seen to be quite in keeping with the classical model.



RANGE **21630.857**
SILL **1.356**
NUGGET **0.340**

MAXDIST = 30000

Figure 6.10. Final Log Nickel Covariogram

Here we no longer show the variogram, since its main purpose was to estimate the desired *covariogram*. By using the estimated *range*, *sill* and *nugget* parameters $(\hat{r},\hat{s},\hat{a}) = (21631,\ 1.356\ ,0.340)$ shown on the right, we can now construct estimates of all desired covariances as in (6.2.65) and (6.2.66) above.

To use these parameters in MATLAB, recall that the first cell of the **OUT** structure above contains these parameter values. So we may identify these for later use as:

```
>> p_log = OUT{1}
```

Note that by leaving off the semicolon on the command line, the new vector is automatically displayed as

```
    p_log =    21631      1.3561      0.34026
```

so that the correctness of this command is easily checked from the output above.

Simple Kriging at a Selected Point

Given this covariogram estimate, we first apply simple kriging to a single point in order to illustrate the procedure. In particular we choose the point, **s0** = (659000,586000),[20] shown as a red dot in Figure 6.11 below. Here the nickel values in Figure 4.18 have been replaced by log-nickel values. Notice that while the values have changed, the overall pattern is essentially the same. With respect to the particular point, **s0**, it appears that a bandwidth of **h0** = 5000 meters is sufficient to capture the (12) most important neighbors of this point, as shown in the enlarged portion of the map. So for purposes of this illustration we take the relevant prediction set, $S(s_0)$, to be given by these 12 points.

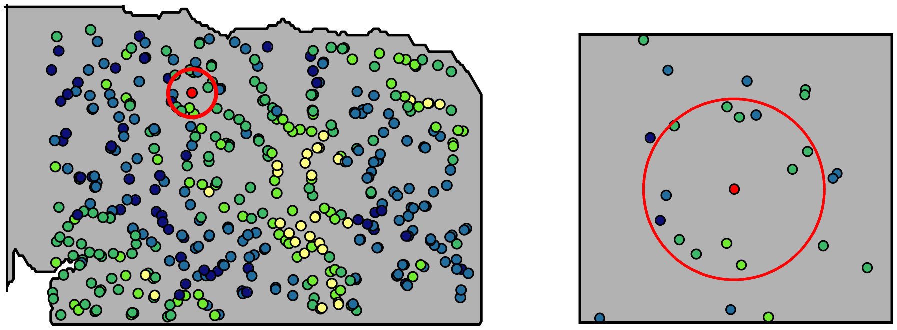


Figure 6.11. Point s0 and its Prediction Set

The rest of the simple kriging procedure is operationalized in the MATLAB program, **krige_simple.m** . So to obtain the desired *simple kriging prediction* and an associated estimate of the *standard error of prediction* at **s0**, one can use the command:

```
>> OUT = krige_simple(h0,p_log,log_nickel,s0)
```

Here the **OUT** matrix lists the krige prediction in the first column and the standard errors in the second column (see also the documentation at the beginning of the program). So in the present case, we can simply leave off the semicolon again and see the screen display:

```
>>  OUT =   3.0488      0.76697
```

If we now denote **nickel** values by the random variable, *Y*, and **log_nickel** values by *logY*, the *kriging prediction* of **log_nickel** at the point **s0** is seen to be

$$\widehat{\log Y}(s_0) = 3.0488 \qquad (6.2.80)$$

[20] In the following discussion we shall refer to the given *location* as **s0** when discussing input/output for MATLAB programs, and as s_0 when referring to the formal development above. The same is true of bandwidths, where **h0** and h_0 .will be used respectively.

where the "hat" notation, $\widehat{\log Y}$, is used to denote a *prediction* (or *estimate*) of the random variable, $\log Y$. The corresponding estimate of the *standard error of prediction* at location **s0** is then given by,

(6.2.81) $\hat{\sigma}_0 = 0.76697$

For our later purposes, it is important to note that as in Step 1 of the estimation procedure for simple kriging, this program uses the *sample mean* of the **log_nickel** data, which can be obtained directly in MATLAB with the command

>> mean(log_nickel(:,3))

which in this case yields the value, $\hat{\mu} = 3.252$.

Comparison with Geostatistical Analyst

Before analyzing this simple kriging output further, it is instructive to compare it with the output obtained by using the simple kriging procedure in Geostatistical Analyst. First it is necessary to construct log-nickel values in ARCMAP. This is easily accomplished by opening the attribute table for the **Vancouver_dat** shapefile, making a new field, say **LOGNI**, and using the Calculator to create the logs of Nickel values [written in the calculator window as **log([NI])**].[21] [These log values are shown in Figure 6.11 above.] To perform simple kriging start with the path:

Geostatistical Analyst → Geostatistical Wizard → Kriging

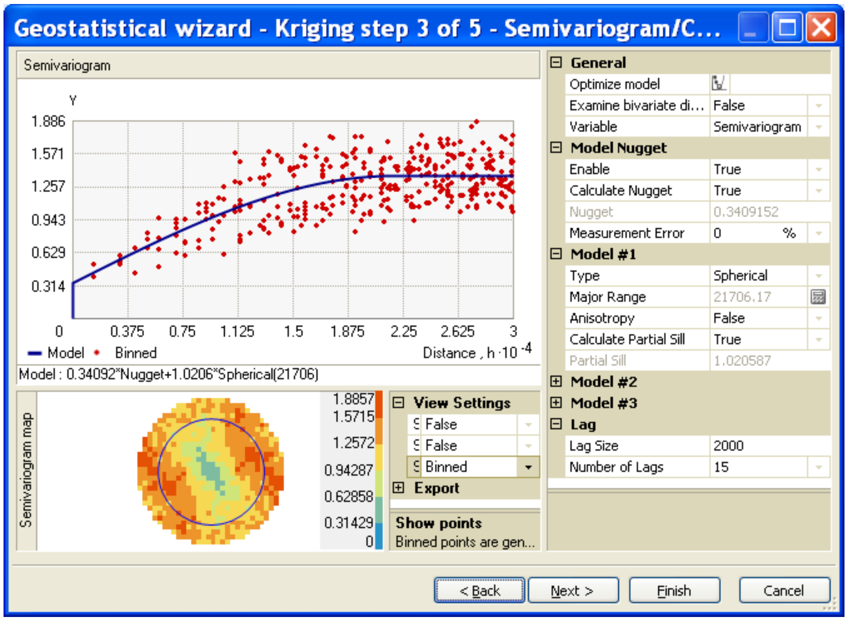
and use attribute **LOGNI** for input data **Vancouver_dat.** In the next window, select

Simple Kriging → Prediction Map

Notice that the mean value is displayed as 3.2515, which is precisely the (rounded) MATLAB value above. In the next window, be sure to select the "Semivariogram" option, to obtain a variogram plot. Recall that the maxdist above was chosen to be 30000 meters.

To obtain a fit that is roughly comparable in this case set the number of lags to 15 with a lag size of 2000 meters (yielding a maxdist of $15 \times 2000 = 30000$ meters) as shown in Figure 6.12 below. Here the estimated *range* of 21706 meters is remarkably close to the MATLAB value of 21630 meters in Figure 6.10 above. Similarly, the estimated *nugget* value, 0.3409, and *sill* value, $(.3409 + 1.0206 = 1.3615)$, are also very close to those in Figure 6.10. So in this case one expects the simple kriging results to be quite similar as well.

[21] As with MATLAB, the "log( )" function in ARCMAP calculates *natural logs*.



Geostatistical wizard - Kriging step 3 of 5 - Semivariogram/C...

Model : 0.34092*Nugget+1.0206*Spherical(21706)

General
Optimize model
Examine bivariate di... False
Variable Semivariogram

Model Nugget
Enable True
Calculate Nugget True
Nugget 0.3409152
Measurement Error 0 %

Model #1
Type Spherical
Major Range 21706.17
Anisotropy False
Calculate Partial Sill True
Partial Sill 1.020587

Model #2
Model #3

Lag
Lag Size 2000
Number of Lags 15

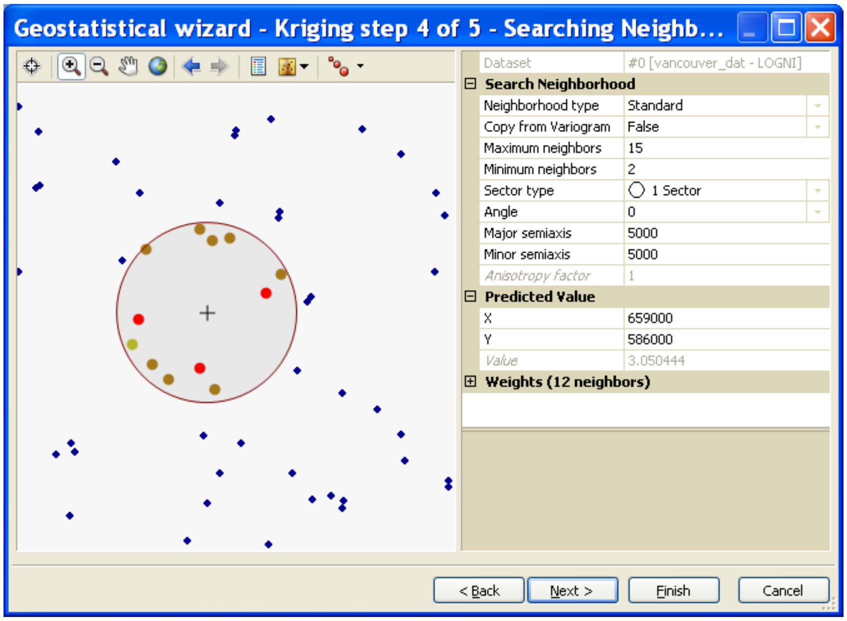
View Settings
False
False
Binned

Export

Show points
Binned points are gen...

Figure 6.12 Variogram for Log Nickel Data

This can be verified in the next window, shown in Figure 6.13 below. Here the sample point coordinates have been set to X = 659000 and Y = 586000 to agree with the point **s0** above. Similarly, to produce a circular neighborhood of 5000 meters, the "Sector type" is set to the simple ellipse form shown, and the axes are both set to 5000 to yield a circle.[22]



Geostatistical wizard - Kriging step 4 of 5 - Searching Neighb...

Dataset #0 [vancouver_dat - LOGNI]

Search Neighborhood
Neighborhood type Standard
Copy from Variogram False
Maximum neighbors 15
Minimum neighbors 2
Sector type ○ 1 Sector
Angle 0
Major semiaxis 5000
Minor semiaxis 5000
Anisotropy factor 1

Predicted Value
X 659000
Y 586000
Value 3.050444

Weights (12 neighbors)

Figure 6.13. Kriging Prediction at s0 = (X,Y)

[22] Be sure to set **Copy from Variogram** = "False" in order to set these axis values.

Notice also that the maximum "Neighbors to include" has been set to 15 to ensure that all points in the circle around point (X,Y) in the preview window will be included.[23]

The kriging prediction for log nickel is then displayed in Figure 6.13 as "Prediction = 3.0504`" located below the (X,Y) coordinate values. [Notice also that exactly the 12 points inside the circle have been used for this kriging prediction.] As expected, this value is seen to be quite close to the MATLAB prediction in (6.2.80) above.

Finally, to produce an estimate of the standard error of prediction at (X,Y), click "Back" twice to return to the "Step 1" window and now select

Simple Kriging → Prediction Standard Error Map

With this selection, return to the "Step 3" window by clicking "Next" twice. Notice that that all settings in Steps 2 and 3 have remained constant, so that prediction standard errors are now being calculated under the same settings as the kriging prediction. The only change is that "Prediction = 3.0504" is now replaced by "Error = 0.7676". Again, this value is quite close to the MATLAB standard error estimate in (6.2.81) above. As mentioned above, this close agreement is largely due to the similarity of the variogram parameter estimates in this case. Hence such close agreement cannot be expected in general.

Analysis of the Simple Kriging Results

By applying the prediction interval result in expression (6.2.61) above, we can immediately obtain a (default) prediction interval for the log-nickel value at **s0**. However, this is not particularly appropriate, since it is *nickel values* (in parts per million, *ppm*) that we are really interested in. Indeed the only reason for using log-nickel values was to obtain a better normal approximation, so that prediction intervals will have some statistical validity. But having obtained such a prediction interval, we now wish to *transform* this interval back to nickel values. Here the idea is very simple. Notice first that if $g(Y)$ is any monotone increasing function of a random variable [such as $\log(Y)$] then the function g has a well-defined *inverse*, g^{-1}, which is also monotone increasing. So for any three random variables (Z_1, Z_2, Z_3) the following "inequality events" must be identical

$$(6.2.82) \qquad Z_1 \le g(Z_2) \le Z_3 \iff g^{-1}(Z_1) \le g^{-1}[g(Z_2)] \le g^{-1}(Z_3)$$

$$\iff g^{-1}(Z_1) \le Z_2 \le g^{-1}(Z_3)$$

[23] Note also in Figure 6.13 that the "Enlarge" tool for the preview window has been used to focus in on the point (X,Y).

where the last line follows from the identity, $g^{-1}[g(Z_2)] \equiv Z_2$. This in turn implies that the probabilities of these events must be identical, so that

$$(6.2.83) \qquad \Pr[Z_1 \le g(Z_2) \le Z_3] = \Pr[g^{-1}(Z_1) \le Z_2 \le g^{-1}(Z_3)]$$

Now in the present case, recall from (6.2.59) that the 95% prediction interval for $\log Y(s_0)$ is defined by the relation:

$$(6.2.84) \qquad \Pr[\widehat{\log Y}(s_0) - (1.96)\hat{\sigma}_0 \le \log Y(s_0) \le \widehat{\log Y}(s_0) + (1.96)\hat{\sigma}_0] = .95$$

Hence if we now let $Z_1 = \widehat{\log Y}(s_0) - (1.96)\hat{\sigma}_0, Z_2 = \log Y(s_0),\ Z_3 = \widehat{\log Y}(s_0) + (1.96)\hat{\sigma}_0$ and let $g(\cdot) = \log(\cdot)$ so that $g^{-1}(\cdot) = \exp(\cdot)$, then it follows at once from (6.2.83) and (6.2.84) that

$$(6.2.85) \qquad \Pr\left[\exp\left(\widehat{\log Y}(s_0) - (1.96)\hat{\sigma}_0\right) \le Y(s_0) \le \exp\left(\widehat{\log Y}(s_0) + (1.96)\hat{\sigma}_0\right)\right] = .95$$

This yields the desired *prediction interval for* $Y(s_0)$. In the present case we have the estimated values,

$$(6.2.86) \qquad \left[\exp\left(\widehat{\log Y}(s_0) - (1.96)\,\hat{\sigma}_0\right), \exp\left(\widehat{\log Y}(s_0) + (1.96)\,\hat{\sigma}_0\right)\right]$$

$$= \left[\exp\left(3.0504 - (1.96)(.7676)\right), \exp\left(3.0504 + (1.96)(.7676)\right)\right]$$

$$= [\ \exp(1.5459)\ ,\ \exp(4.5549)] = [4.6922, 95.097]$$

and hence can be 95% confident that the true value of $Y(s_0)$ lies in the interval $[4.6922, 95.097]$. Note finally that [as stated following expression (6.2.61)] this result can be interpreted to mean that if we were able to perform this same estimation procedure many times, then $Y(s_0)$ would lie in the *estimated interval* about 95% of the time. So in the present case, one can be reasonably confident that the interval obtained (namely $[4.6922, 95.097]$) does indeed contain $Y(s_0)$.

Full Kriging of Log Nickel

While the restriction to a single point, **s0**, was valuable as an illustration of the Simple Kriging procedure, typically one wishes to predict (estimate) the entire sample area based on the observed data points $\{y(s_i): i = 1,.., N\}$. In ARCMAP this is precisely the "default" option (where predictions are restricted to the smallest box in the sample area containing the observed data). But in MATLAB one must actually specify the set of points where predictions are desired. So a simple procedure here is to use the program, **grid_form.m**, to construct a reasonably fine grid of points in the smallest box containing the data. To display this visually, one can then import this data to ARCMAP and use some appropriate (non-statistical) interpolation method to interpolate this grid to every pixel. In the MATLAB file, **nickel.mat**, the coordinates of all 437 data points are in the matrix, **L0**. So to form a *bounding box*, write:

```
>> Xmin=min(L0(:,1));
>> Xmax=max(L0(:,1));
>> Ymin=min(L0(:,2));
>> Ymax=max(L0(:,2));
```

Next, to choose a grid cell size, observe from the map display in ARCMAP that a division of the box sides into about 25 segments yields a reasonably fine grid for interpolation. So we now set,

```
>> Xcell = (Xmax-Xmin)/25;
>> Ycell = (Ymax-Ymin)/25;
```

and use the command (recall the application on p.4-26 of Part I):

```
>> G = grid_form(Xmin,Xmax,Xcell,Ymin,Ymax,Ycell);
```

to construct an appropriate grid, **G**. This grid is shown in Figure 6.14 below, and is seen to just cover the region of the data points. Using grid **G** as an input rather than the single point, **s0**, we can then obtain a full kriging of all grid points with the command:

```
>> OUT_G = krige_simple(h0,p_log,log_nickel,G);
```

[Here we use the semicolon to avoid screen output of all kriging values.] This data can then be imported to ARCMAP by making a data table,

```
>> DAT_G = [G,OUT_G];
```

in which the first two columns include the grid coordinate points and the last two include the krige and standard error estimates at each grid point. By saving this as an ASCII file;

```
>> save DAT_G.txt DAT_G -ascii
```

(and editing the file in EXCEL to include column labels) one can then import **DAT_G.txt** into ARCMAP, make a shapefile **Simple_Krige_Grid.shp**, and display this layer as shown in Figure 6.15 below.

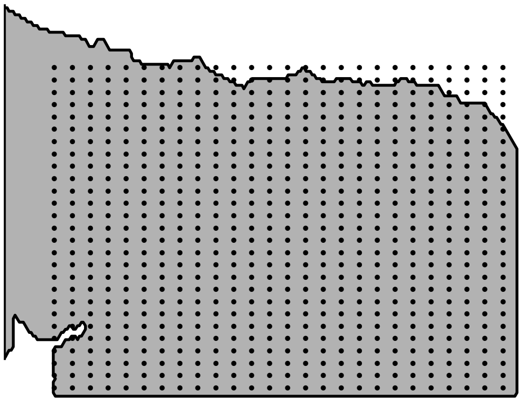


Figure 6.14. Interpolation Grid

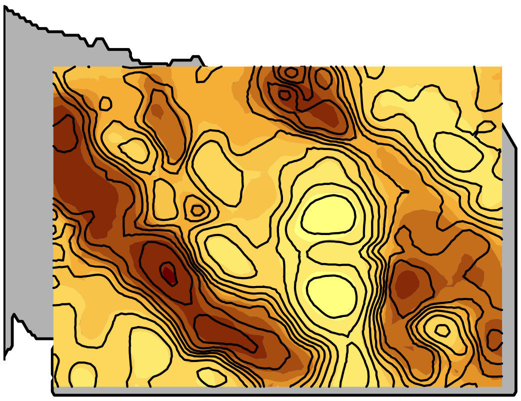


Figure 6.15. Simple Kriging Comparison

To display the simple kriging results from MATLAB, we can then use any of the interpolators in **Geostatistical Analyst**. The contours shown in Figure 6.15 are obtained by first interpolating the kriging data in **Simple_Krige_Grid** with the *radial basis functions* option, and then using the command, **Data → Export to Vector**. The layer produced contains precisely these contours. The reason why contours are used here is to allow a visual comparison with a simple kriging of log-nickel in **Geostatistical Analyst**. This is accomplished by completing the simple kriging procedure outlined above [that we terminated with Step 3 (Searching Neighborhood) shown in Figure 6.13]. If one places the contours above the kriging map displayed, then both can be seen together.[24]

Finally, this visual comparison shows that while these two kriging surfaces are not in perfect agreement, they are *qualitatively very similar*. Moreover, while the Geostatistical Analyst procedure is clearly easier to perform in this case, the MATLAB "grid" procedure will prove to be very useful for universal kriging, where the Geostatistical Analyst version is very limited in terms of applications. This will be illustrated by the "Venice example" in Section 7.3.5 below.

[24] To make the boundaries of the kriging map agree exactly with the contours (as seen in Figure 6.15), open the "properties" of the kriging map layer, select "Extent" and set this to "the rectangular extent of Simple_Krige_Grid".

6.3 The Ordinary Kriging Model

The procedural details of Ordinary Kriging are almost identical to those of Simple Kriging. Hence the present development focuses on those aspects that extend the above analysis by internalizing the estimation of the *unknown mean*, μ. Here again we start with a spatial stochastic process $\{Y(s) = \mu + \varepsilon(s) : s \in R\}$ where each finite set of sample variates, $\{Y(s_i) = \mu + \varepsilon(s_i) : i = 1,..,n\}$, is assumed to be multi-normally distribution with *known covariances*, $\mathrm{cov}[\varepsilon(s_i),\varepsilon(s_j)]$, $i, j = 1,..,n$. Given such a sample, we again consider the problem of predicting $Y(s_0)$ at some location, $s_0 \in R$, not in this sample. It is also assumed that the relevant prediction set, $S(s_0) = \{s_1,..,s_{n_0}\}$, for location s_0 has been identified within this set of sample locations. Hence the basic task is to predict a value for $Y(s_0)$ in terms of observed values of the variates $\{Y(s_1),..,Y(s_{n_0})\}$. By the linear prediction hypothesis in (6.1.2) we then seek a best linear unbiased (BLU) predictor,

$$\hat{Y}(s_0) = \sum\nolimits_{i=1}^{n_0} \lambda_{i0} Y(s_i) \tag{6.3.1}$$

of $Y(s_0)$. To facilitate the interpretation of this predictor, it is convenient to proceed in two steps. First we develop a *BLU estimator* of μ, and then use this result to simplify the form of the *BLU predictor* obtained for $Y(s_0)$.

6.3.1 Best Linear Unbiased Estimation of the Mean

Since the mean, μ, is assumed to be constant throughout region *R*, it is natural to use the *entire set* of sample observations, $\{Y(s_i) = \mu + \varepsilon(s_i) : i = 1,..,n\}$, to estimate μ. To do so, we again we start with the linear hypothesis that the desired estimate, $\hat{\mu}_n$, can be written as a linear combination of these observations, say

$$\boxed{\hat{\mu}_n = \sum\nolimits_{i=1}^{n} a_i Y(s_i) = a' Y_n} \tag{6.3.2}$$

where $Y_n = [Y(s_1),..,Y(s_n)]'$ denotes the *full sample* vector of *Y*-variates, and where $a' = (a_1,..,a_n)$ denotes the vector of unknown coefficients. To ensure that this linear estimator is *unbiased*, we then require that

$$\mu = E(\hat{\mu}) = E(a'Y_n) = a'E(Y_n) = a'(\mu 1_n) = \mu(a'1_n) = \mu(1_n'a) \tag{6.3.3}$$

where $1_n = (1,..,1)'$ is the unit vector of length *n*. Hence unbiasedness for all values of μ will be guaranteed if and only if these unknown coefficients sum to one, i.e.,

$$1_n'a = 1 \tag{6.3.4}$$

Among all such linear unbiased estimators, we seek that one with minimum variance. To calculate the variance of linear estimators, we start by letting

$$(6.3.5) \qquad V = \text{cov}(Y_n) = \begin{pmatrix} \sigma^2 & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma^2 \end{pmatrix}$$

denote the *full sample* covariance matrix (in contrast to the smaller covariance matrices, V_0, for each predictor set, $S(s_0) \subseteq \{s_1,..,s_n\}$). With this definition, it follows at once from (3.2.21) that

$$(6.3.6) \qquad \text{var}(a'Y_n) = a'\text{cov}(Y_n)a = a'Va$$

Hence to determine the linear unbiased estimator of μ with smallest variance, we seek to find that coefficient vector, $\hat{a}$, that yields a minimum value of (6.3.6) subject to the unit-sum condition in (6.3.4), i.e., which solves the following *constrained minimization problem* in a:

$$(6.3.7) \qquad \boxed{\textit{minimize}: \ a'Va \qquad \textit{subject to}: \ 1_n'a = 1}$$

In expression (A2.8.23) of the Appendix it is shown that the unique solution of this problem is given by the coefficient vector:

$$(6.3.8) \qquad \hat{a} = \left(\frac{1}{1_n'V^{-1}1_n}\right)V^{-1}1_n$$

Hence for each possible vector of sample variates, $Y_n = [Y(s_1),..,Y(s_n)]'$, the unique *BLU estimator* for μ is given by:

$$(6.3.9) \qquad \boxed{\hat{\mu}_n = \hat{a}'Y_n = \left(\frac{1}{1_n'V^{-1}1_n}\right)1_n'V^{-1}Y_n = \frac{1_n'V^{-1}Y_n}{1_n'V^{-1}1_n}}$$

To gain some feeling for this estimator, consider the classical case of uncorrelated samples, namely where the covariance matrix in (6.3.5) reduces to

$$(6.3.10) \qquad V = \text{cov}(Y_n) = \begin{pmatrix} \sigma^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{pmatrix} = \sigma^2 I_n$$

with I_n denoting the *n*-square identity matrix. In this case we see that

(6.3.11) $$\hat{\mu}_n = \frac{1_n'(I_n)Y_n}{1_n'(I_n)1_n} = \frac{1_n'Y_n}{1_n'1_n}$$

But since $1_n'1_n = \sum_{i=1}^n (1) = n$ and $1_n'Y_n = \sum_{i=1}^n Y(s_i)$, it follows that

(6.3.12) $$\boxed{\hat{\mu}_n = \tfrac{1}{n}\sum_{i=1}^n Y(s_i) = \overline{Y}_n}$$

Thus $\hat{\mu}_n$ reduces to the *sample mean*, $\overline{Y}_n$, which is of course the unique BLU estimator of μ for uncorrelated samples. Hence in the presence of spatial correlation, the optimal weights in the coefficient vector, $\hat{a}$, reflect the covariances among these correlated samples. In the case of Simple Kriging, the use of $\overline{Y}_n$ to estimate μ necessarily results in a linear unbiased estimator with *higher variance* than $\hat{\mu}_n$.

6.3.2 Best Linear Unbiased Predictor of *Y*(s_0)

Given this intermediate result, we now formulate the Best Linear Unbiased prediction problem for $Y(s_0)$. Here we again stress that the prediction set, $S(s_0) = \{s_1,..,s_{n_0}\}$, for s_0 is generally smaller than the full sample of size *n*. So here we focus on the smaller vector of sample variates, $Y = [Y(s_1),..,Y(s_{n_0})]'$ used for predicting $Y(s_0)$ in (6.3.1) above. As in the case of Simple Kriging, if we again denote the desired vector of prediction weights by $\lambda_0 = (\lambda_{01},..,\lambda_{0n_0})'$, then the desired *linear predictor* of $Y_0 = Y(s_0)$ can be written in vector form as

(6.3.13) $$\hat{Y}_0 = \lambda_0' Y$$

For purposes of prediction, recall from (6.1.4) that the desired *unbiasedness criterion* for $\hat{Y}_0$ is that expected *prediction error* be zero, i.e., that

(6.3.14) $$\begin{aligned} 0 = E(e_0) = E(Y_0 - \hat{Y}_0) &= E(Y_0) - E(\lambda_0' Y) \\ &= \mu - \lambda_0' E(Y) = \mu - \lambda_0'(\mu 1_{n_0}) \\ &= \mu(1 - \lambda_0' 1_{n_0}) \end{aligned}$$

So, as a parallel to (6.3.4) above, it follows that λ_0 will yield an unbiased predictor for all possible values of μ if and only if the bracketed expression is zero, i.e.,

(6.3.15) $$1_{n_0}' \lambda_0 = 1$$

Moreover, to satisfy the *efficiency criterion* it is required that among all linear unbiased predictors, $\hat{Y}_0$ should yield the smallest prediction error variance, which in view of (6.3.15) together with (6.2.12) is again seen to be precisely residual *mean squared error*,

$$(6.3.16)\qquad \text{var}(e_0) = E(e_0^2) = E[(Y_0 - \hat{Y}_0)^2] = E[(Y_0 - \lambda_0'Y)^2] = E\left([(\mu + \varepsilon_0) - \lambda'(\mu 1_{n_0} + \varepsilon)]^2\right)$$

$$= E\left([\mu(1 - \lambda_0' 1_{n_0}) + (\varepsilon_0 - \lambda_0'\varepsilon)]^2\right) = E[(\varepsilon_0 - \lambda_0'\varepsilon)^2] = MSE(\lambda_0)$$

But since all covariances in (6.2.26) continue to be given (i.e., are assumed to be *known*) for the case of Ordinary Kriging, the argument leading to (6.2.27) for Simple Kriging still holds. Hence we again seek to minimize

$$(6.3.17)\qquad MSE(\lambda_0) = \sigma^2 - 2c_0'\lambda_0 + \lambda_0'V_0\lambda_0 \ ,$$

but now subject to the unit sum condition in (6.3.15). Hence the desired weights, $\hat{\lambda}_0$, for Ordinary Kriging are given by the solution of the *constrained minimization problem*:

$$(6.3.18)\qquad \boxed{\textit{minimize}\text{:}\ \ \sigma^2 - 2c_0'\lambda_0 + \lambda_0'V_0\lambda_0 \qquad \textit{subject to}\text{:}\ \ 1_{n_0}'\lambda_0 = 1}$$

The solution to this problem is shown in the Appendix [expression (A2.8.26)] to be given by

$$(6.3.19)\qquad \boxed{\hat{\lambda}_0 = \left(\frac{1 - 1_{n_0}'V_0^{-1}c_0}{1_{n_0}'V_0^{-1}1_{n_0}}\right)V_0^{-1}1_{n_0} + V_0^{-1}c_0}$$

By substituting this solution into (6.3.13), one then obtains the following *BLU predictor* of Y_0 [see also expression (A2.8.28) in the Appendix]:

$$(6.3.20)\qquad \boxed{\hat{Y}_0 = \left(\frac{1_{n_0}'V_0^{-1}Y}{1_{n_0}'V_0^{-1}1_{n_0}}\right) + c_0'V_0^{-1}Y - c_0'V_0^{-1}1_{n_0}\left(\frac{1_{n_0}'V_0^{-1}Y}{1_{n_0}'V_0^{-1}1_{n_0}}\right)}$$

At first glance, this expression appears rather formidable. But by using the results of Section 6.3.1 above, it can be made quite transparent. In particular, suppose that the samples available for mean estimation are taken to be given by the *prediction sample*, Y, at s_0 rather than the full sample, Y_n. Then it follows at once from (6.3.9) that this *BLU estimator* must be of the form

$$(6.3.21)\qquad \hat{\mu}_{n_0} = \frac{1_{n_0}'V_0^{-1}Y}{1_{n_0}'V_0^{-1}1_{n_0}}$$

where n is now replaced by n_0, and where V is replaced by $V_0 = \text{cov}(Y)$. So by substituting (6.3.21) into (6.3.20) , we see that this optimal predictor reduces to

$$(6.3.22) \qquad \hat{Y}_0 = \hat{\mu}_{n_0} + c_0' V_0^{-1} Y - c_0' V_0^{-1}(\hat{\mu}_{n_0} 1_{n_0})$$

$$= \hat{\mu}_{n_0} + c_0' V_0^{-1}(Y - \hat{\mu}_{n_0} 1_{n_0})$$

Finally, if we treat $\hat{\mu}_{n_0}$ as a prior estimate of μ, and [as in (6.2.2)] take the corresponding sample residuals based on this prior estimate, to be

$$(6.3.23) \qquad \hat{\varepsilon}_i = Y(s_i) - \hat{\mu}_{n_0} \,, \quad i = 1,..,n_0$$

then the vector of these residuals is given by

$$(6.3.24) \qquad \hat{\varepsilon} = \begin{pmatrix} \hat{\varepsilon}_1 \\ \vdots \\ \hat{\varepsilon}_{n_0} \end{pmatrix} = \begin{pmatrix} Y(s_1) - \hat{\mu}_{n_0} \\ \vdots \\ Y(s_{n_0}) - \hat{\mu}_{n_0} \end{pmatrix} = Y - \hat{\mu}_{n_0} 1_n$$

Similarly, if we let $\hat{\varepsilon}_0 = \hat{Y}_0 - \hat{\mu}_{n_0}$ denote the residual predictor corresponding to $\hat{Y}_0$, then (6.3.22) is further reduced to

$$(6.3.25) \qquad \boxed{\hat{\varepsilon}_0 = c_0' V_0^{-1} \hat{\varepsilon}}$$

But by (6.2.29) this is seen to be precisely the *Simple Kriging predictor* of $\varepsilon_0 = \varepsilon(s_0)$ based on the vector of residual data, $\hat{\varepsilon}$.

In short, the *BLU predictor* of $Y_0 = Y(s_0)$ in (6.3.20) can be obtained by the following two-part procedure:

(***i***). Construct the BLU estimator, $\hat{\mu}_{n_0}$, of μ based on the prediction sample data, Y, as in (6.3.21).

(***ii***). Use the sample residuals, $\hat{\varepsilon}$, in (6.3.24) to obtain the Simple Kriging predictor, $\hat{\varepsilon}_0$, of ε_0 as in (6.3.25), and set $\hat{Y}_0 = \hat{\mu}_{n_0} + \hat{\varepsilon}_0$.

In retrospect, this procedure seems quite natural. Since all covariance information is assumed to be *given* (as in Simple Kriging) the first step simply uses this information to obtain a BLU estimator for μ. The second step then uses Simple Kriging to construct the predictor. What is remarkable here is that this *ad hoc* procedure actually yields the *Best Linear Unbiased predictor* for $Y(s_0)$ based solely on the prediction sample Y.

The only shortcoming of this procedure is that it does not use all sample information available for estimating μ. For since this mean is assumed to be constant over the entire region *R*, it should be clear that a better estimate can be obtained by using the BLU estimator, $\hat{\mu}_n$, based on the *full sample*, Y_n. It is this modified procedure that constitutes the most commonly used form of *Ordinary Kriging*.[25] To formalize this procedure, it thus suffices to modify the two steps above as follows:

(1). Construct the BLU estimator, $\hat{\mu}_n$, of μ based on the full sample data, Y_n, as in (6.3.9).

(2). Use the sample residuals, $\hat{\varepsilon} = Y - \hat{\mu}_n 1_{n_0}$ to obtain the Simple Kriging predictor, $\hat{\varepsilon}_0$, of ε_0 as in (6.3.25), and set $\hat{Y}_0 = \hat{\mu}_n + \hat{\varepsilon}_0$.

6.3.3 Standard Error of Prediction

Recall that to obtain prediction intervals, one requires an estimate of the *standard error of prediction*, σ_0, as well as $\hat{Y}_0$. To do so, recall from the argument in (6.3.16) and (6.3.17) that prediction error variance for any weight vector, λ_0, has the same form as for Simple Kriging, i.e.,

$$(6.3.26) \qquad \sigma_0^2 = \mathrm{var}(e_0) = \sigma^2 - 2c_0'\lambda_0 + \lambda_0' V_0 \lambda_0$$

So all that is required to obtain the desired prediction error variance is to substitute the optimal weight vector, $\hat{\lambda}_0$, into this expression. After some manipulation, it can be shown [see expression (A2.8.69) in the Appendix] that desired value, $\hat{\sigma}_0^2$, is given by:

$$(6.3.27) \qquad \hat{\sigma}_0^2 = \sigma^2 - 2c_0'\hat{\lambda}_0 + \hat{\lambda}_0' V_0 \hat{\lambda}_0$$

$$= (\sigma^2 - c_0' V_0^{-1} c_0) + \frac{(1 - 1_{n_0}' V_0^{-1} c_0)^2}{1_{n_0}' V_0^{-1} 1_{n_0}}$$

The key point to notice is that the first bracketed expression is precisely the prediction error variance for Simple Kriging in expression (6.2.53). But since the second term is

[25] It should noted however that one may consider "local" versions of ordinary kriging in which the mean is *re-estimated* at each prediction site, s_0. This yields a set of local mean estimates, $\hat{\mu}(s_0)$, which can be regarded as local estimates of a possibly non-constant trend surface. See for example [BG], pp.195-196. This idea is also implicit in Section 5.4.2 of Schabenberger and Gotway (2005).

always *positive*,[26] it follows that prediction error variance for Ordinary Kriging is always *larger* than for Simple Kriging. The additional positive term turns out to be precisely the addition to prediction error variance created by the internal estimation of the mean, μ.

Finally, given this expression for prediction error variance, the desired *standard error of prediction* is simply the square root of this expression, namely,

$$\boxed{\hat{\sigma}_0 = \sqrt{(\sigma^2 - c_0'V_0^{-1}c_0) + \frac{(1 - 1_{n_0}'V_0^{-1}c_0)^2}{1_{n_0}'V_0^{-1}1_{n_0}}}} \tag{6.3.28}$$

6.3.4 Implementation of Ordinary Kriging

From the development above, it should be evident how to implement Ordinary Kriging by a direct modification of the three-step procedure for Simple Kriging in Section 6.2.5. To do so, we again start by assuming the existence of a given set of *n* observations (data points), $\{y_i = y(s_i) : i = 1,..,n\}$ in *R*, where each y_i is a realization of the corresponding random variable, $Y(s_i)$, in the full sample vector, $Y_n = [Y(s_i) : i = 1,..,n]'$, in (6.3.2) above. In this context, we again consider the prediction of $Y_0 = Y(s_0)$, at a single given location, $s_0 \in R$, with respect to a given prediction set, $S(s_0) = \{s_1,..,s_{n_0}\} \subseteq \{s_1,..,s_n\}$. Within this framework, we can operationalize the *Ordinary Kriging model* by re-ordering the three steps of the Simple Kriging implementation in Section 6.2.5 as follows:

Step 1. Estimation of Covariances

This step amounts essentially to a reinterpretation of Step 2 for Simple Kriging, where here we focus on Y-process rather than the ε-process. To do so, simply recall from Section 4.8 that (as with Simple Kriging) Ordinary Kriging assumes a *constant-mean* model $[Y(s) = \mu + \varepsilon(s) :\ s \in R]$, so that the variograms for the Y-process and ε-process are *identical*. Hence we can again use the sample data $(y_1,..,y_n)$ in **var_spher_plot.m** to obtain a spherical variogram estimate, $\gamma(h;\hat{r},\hat{s},\hat{a})$, and derived covariogram estimate as in (6.2.65), i.e.,

$$\hat{C}(h) = \hat{s} - \gamma(h;\hat{r},\hat{s},\hat{a}) \tag{6.3.29}$$

The only difference in the present setting is that we treat the covariances between Y values rather than ε values. In particular, we now require estimates of the covariances, $\sigma_{ij} = \text{cov}[Y(s_i),Y(s_j)]$, for all sample pairs, $Y(s_i)$ and $Y(s_j)$, in Y_n. Using (6.3.29), these can be estimated precisely as (6.2.66) by setting,

[26] Positivity of the denominator follows from the fact that it is the *variance* of the linear compound, $1_{n_0}'V_0^{-1}Y$, since $\text{var}(1_{n_0}'V_0^{-1}Y) = 1_{n_0}'V_0^{-1}\,\text{cov}(Y)V_0^{-1}1_{n_0} \ = 1_{n_0}'V_0^{-1}(V_0)V_0^{-1}1_{n_0} \ = 1_{n_0}'V_0^{-1}1_{n_0}$.

$$(6.3.30)\qquad \hat{\sigma}_{ij} = \widehat{\text{cov}}[Y(s_i),Y(s_j)] = \hat{C}(\| s_i - s_j \|)$$

These in turn provide an estimate of the full-sample covariance matrix, $V = \text{cov}(Y_n)$, in (6.3.5) as follows:

$$(6.3.31)\qquad \hat{V} = \begin{pmatrix} \hat{\sigma}^2 & \cdots & \hat{\sigma}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{\sigma}_{n1} & \cdots & \hat{\sigma}^2 \end{pmatrix}$$

By the same procedure, we can obtain estimates for all covariances between the variable, $Y_0 = Y(s_0)$, to be predicted and the given set of prediction variates, $Y = [Y(s_1),..,Y(s_{n_0})]'$, namely,

$$(6.3.32)\qquad \hat{\sigma}_{0j} = \widehat{\text{cov}}[Y(s_0),Y(s_j)] = \hat{C}(\| s_0 - s_j \|) \ , \quad j = 1,..,n_0$$

By again letting $\hat{c}_0 = (\hat{\sigma}_{0i} : i = 1,..,n_0)'$, we can use these together with the appropriate sub-matrix, $\hat{V}_0$, of covariance estimates in (6.3.27) to obtain an estimate,

$$(6.3.33)\qquad \boxed{\hat{C}_0 = \begin{pmatrix} \hat{\sigma}^2 & \hat{c}_0{}' \\ \hat{c}_0 & \hat{V}_0 \end{pmatrix}}$$

of the full covariance matrix, C_0, relevant for prediction at s_0. From a computational viewpoint, this matrix is numerically identical to the matrix in (6.2.23), with elements now interpreted as covariances directly between Y values rather than ε values.

Step 2. Estimation of the Mean

This step involves the main departure from Simple Kriging. Here we replace the sample-mean estimator ($\overline{Y}_n$) of μ with the BLU estimator, $\hat{\mu}_n$, in expression (6.3.9) above. By using the covariance estimates in (6.3.27) together with the *sample data vector*, $y = (y_1,..,y_n)'$, this estimate can be calculated as

$$(6.3.34)\qquad \boxed{\hat{\mu}_n = \frac{1_n' \hat{V}^{-1} y}{1_n' \hat{V}^{-1} 1_n}}$$

Step 3. Estimation of Kriging Predictions

As emphasized in the final two-step procedure of Section 6.3.2 above, this step is identical to that in the Simple Kriging procedure. All that it required at this point is to replace the sample-mean estimate, $\hat{\mu}$, with the BLU estimate, $\hat{\mu}_n$, and redefined the appropriate residual estimates in (6.2.70) by

(6.3.35) $$\hat{\varepsilon}_i = y_i - \hat{\mu}_n \quad , \; i = 1,..,n_0$$

and again use (6.2.71) and (6.2.72) to construct the desired prediction, $\hat{Y}_0$, by

(6.3.36) $$\boxed{\hat{Y}_0 = \hat{Y}(s_0) = \hat{\mu}_n + \hat{c}_0' \hat{V}_0^{-1} \hat{\varepsilon}}$$

Finally, the estimated standard error of prediction, $\hat{\sigma}_0$, is given by substituting the covariance estimates into (6.3.28) above to obtain:

(6.3.37) $$\boxed{\hat{\sigma}_0 = \sqrt{(\hat{\sigma}^2 - \hat{c}_0' \hat{V}_0^{-1} \hat{c}_0) + \frac{(1 - 1_{n_0}' \hat{V}_0^{-1} \hat{c}_0)^2}{1_{n_0}' \hat{V}_0^{-1} 1_{n_0}}}}$$

The pair $(\hat{Y}_0, \hat{\sigma}_0)$ can then be used to construct prediction intervals for $Y_0 = Y(s_0)$ precisely as in (6.2.62) and (6.2.63) above.

6.3.5 An Example of Ordinary Kriging

This implementation of Ordinary Kriging can be illustrated in terms of the Log-Nickel example developed for Simple Kriging in Section 6.2.6. As emphasized in the above implementation, all covariogram estimates are identical. Hence from a practical viewpoint, the only numerical differences between these prediction procedures will result from the replacement of the sample-mean estimator, $\hat{\mu} = \bar{y}_n$, in (6.2.62) with the BLU estimator, $\hat{\mu}_n$, in (6.3.30). Recall that in the present case, $\bar{y}_n = 3.252$. A computation of $\hat{\mu}_n$ using the same data turns out to yield a value $\hat{\mu}_n = 3.329$, which is quite similar to that of $\bar{y}_n$. Hence in the present example, one can expect to find very similar predictions and standard errors. However, it should be stressed that this by no means true in general. Indeed when substantial spatial dependencies are present, the sample mean $\bar{y}_n$ can yield a very poor estimate of μ relative to $\hat{\mu}_n$.

With these general observations, we can now sketch how Ordinary Kriging is done in both MATLAB and ARCMAP. Starting with MATLAB, Ordinary Kriging is operationalized in the program, **o_krige.m**. The inputs are essentially the same as **simple_krige.m**, except that *values* are made distinct from *locations*. So here, values are given by **y = log_nickel(:,3)** and locations by **L0 = log_nickel(:,1:2).** To obtain a prediction at the given location, **s0 = (659000,586000)**, in Figure 6.11 above, one now uses the command:

```
>> OUT = o_krige(y,L0,s0,h0,p_log);
```

Here the prediction and standard error are the last two cells of the output structure, which can be obtained as:

```
>>  [OUT{3} OUT{4}]  =  3.0461     0.76771
```

A comparison with the results on p.5-30 above show that (as expected) the Ordinary Kriging results are virtually the same.

Finally, to carry out an Ordinary Kriging prediction at **s0** in ARCMAP, the procedure described for Simple Kriging is again the same, except that at "Kriging Step 2 of 5" one now selects **Kriging Type = Ordinary** (which is the default choice). By employing all the same settings as in the Simple Kriging example (pp.5-31 to 5-32 above), the Ordinary Kriging prediction, $\hat{Y}_0$, and standard error of prediction, $\hat{\sigma}_0$, at **s0** turn out to be

(6.3.38) $$\hat{Y}_0 = 3.0477 \quad \hat{\sigma}_0 = 0.7683$$

Hence, as expected, these are again seen to be virtually the same as those for MATLAB.

6.4 Selection of Prediction Sets by Cross Validation

Before proceeding to more general kriging models, it is important to consider the question of choosing "best" prediction sets, $S(s_0)$, for each prediction site, $s_0 \in R$. At first glance, it would appear that if the range, r, of the covariogram has been correctly estimated by $\hat{r}$, then the most natural choice for predictions sets is to include all points in closer to s_0 than $\hat{r}$. If the set of all n sample point locations is denoted by

(6.3.39) $$S_n = \{s_1,..,s_n\}$$

then this amounts formally to setting

(6.3.40) $$S(s_0) = \{s_i \in S_n : \| s_0 - s_i \| < \hat{r}\}$$

[In fact, this option for defining search neighborhoods is available in "Kriging step 4 of 5" in ARCMAP, as denoted by "Copy from Variogram".] However, in spite of its apparent theoretical appeal, this option generally tends to include "too much". This will become evident in the simulation analysis below.

To determine a "best" size for prediction sets, one first defines a set of candidate sizes. In the present case, we shall focus on circular prediction sets of the form (6.3.40) for a selection of *bandwidths*, $H = \{h_1,..,h_k\}$, and let

$$S_h(s_0) = \{s_i \in S_n : \| s_0 - s_i \| < h\} \; , \quad h \in H \tag{6.3.41}$$

While it is in principle possible to consider different bandwidths at each prediction site, we follow the standard convention of considering only uniform bandwidths (as reflected by the bandwidth parameter, **h0**, used in **o_krige.m**). The task is then to find a "best" bandwidth.

The standard procedures for doing so, known as *cross validation* procedures, leave out part of the data and attempt to predict these values with the rest of the data. By calculating the average prediction error for this data, one can then find the bandwidth that minimizes this value. The most commonly used procedure, known as *leave-one-out cross validation*, is to systematically omit single data points one at a time, and predict these using the rest of the data. Hence, given a candidate bandwidth, $h \in H$, one will obtain for each data point, $y_i = y(s_i)$, a predicted value, say $\hat{y}_i(h)$, by using all other sample data in $y = (y_1,..,y_n)$ together with the prediction set $S_h(s_i)$. By squaring these *prediction errors*, $y_i - \hat{y}_i(h),\ i = 1,..,n$, and taking the average, one obtains a summary measure that can be viewed as a sample version of *mean squared error*. But in order to preserve units (so that values, for example, are in terms of Nickel rather Nickel-squared) the most commonly used measure of performance is *root mean squared error*, as defined for each candidate bandwidth, $h \in H$, by:

$$\boxed{RMSE(h) = \sqrt{\tfrac{1}{n}\sum\nolimits_{i=1}^{n} [y_i - \hat{y}_i(h)]^2}} \tag{6.3.42}$$

Hence by systematically calculating $RMSE(h)$ for all $h \in H$, one can define the best bandwidth, h^*, to be the one that minimizes (6.3.42), i.e.,

$$\boxed{RMSE(h^*) = \min\nolimits_{h \in H} RMSE(h)} \tag{6.3.43}$$

6.4.1 Log-Nickel Example

For the case of Ordinary Kriging, this leave-one-out cross validation procedure is operationalized in the MATLAB program, **o_krige_cross_val.m**. To apply this program to the log-nickel example, recall that the estimated range was $\hat{r} = 21{,}631$ meters, and that the bandwidth chosen for kriging at **s0** was **h0** $= 5000$ meters. Hence we choose H to be the set of bandwidths increasing from 1000 to 25,000 in increments of 1000, i.e.,

```
>> H = [1000:1000:25000];
```

If the $n = 436$ locations and log_nickel values are denoted respectively by

```
>> L = log_nickel(:,1:2);

>> y = log_nickel(:,3);
```

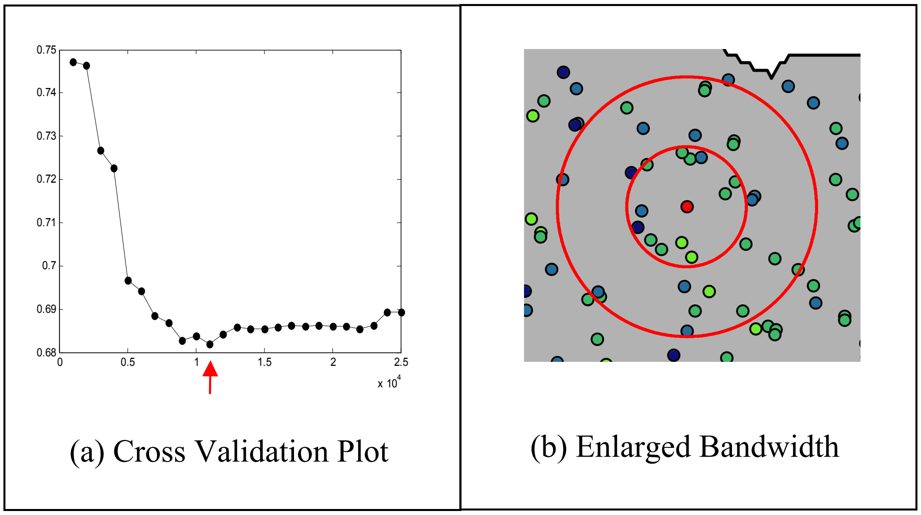
then the above program can be run for this example using the command

```
>> o_krige_cross_val(y,L,H);
```

The output is a graph that plots the values of $RMSE(h)$ against bandwidths, *h*, as shown in Figure 6.16(a) below. Here the *best bandwidth* (shown by the red arrow in the figure) is here seen to be 11,000 meters, which is roughly twice the value chosen for kriging at point **s0** in the examples above. This larger bandwidth is shown by the larger circle in Figure 6.1(b), with the smaller circle denoting the original choice of 5000 meters. Notice that many more data points are now included (33 versus 12 in the original analysis). The predictions obtained by **o_krige.m** using this larger bandwidth are shown below:

```
>> [OUT{3} OUT{4}]  =   3.1219     0.76365
```

So the predicted value is seen to be somewhat higher, and the standard error of prediction is slightly smaller.[27] Since the latter implies a slight tighter prediction interval, this larger bandwidth may indeed be preferable.



(a) Cross Validation Plot

(b) Enlarged Bandwidth

Figure 6.16 Log Nickel Example

[27] The values obtained in ARCMAP are 3.1237 and 0.7643, respectively, and are again seen to be in close agreement.

But the most important point to note here is that this best bandwidth is much *smaller* than the estimate range ($\hat{r} = 21,631$). It can of course be argued that in this particular example, the estimated range may not be very accurate. Indeed, it is well known that estimates of the range tend to be the least stable (most variable) of the three parameter estimates $(\hat{r}, \hat{s}, \hat{a})$. Hence it is useful to consider this question in simulated data sets where the true range is *known*.

6.4.2 A Simulated Example

To construct a simulated example, we start by generating $n = 500$ random points in a 100×100 kilometer square with locations denoted by $L = (s_1,..,s_n)$. Next we simulate $K = 20$ realizations, $Y = (y_1,..,y_K)$, of a covariance-stationary process on these points, where each column, $y_k = (y_{k1},..,y_{kn})'$ is a realization on the locations in L. Here we use a constant mean of $\mu = 10$ and covariogram with parameters, $p = (r,s,a) = (25,5,0)$. The simulation was carried out using the MATLAB program, **cov_stat.m**, with the command:

>> Y = cov_stat(p,L,20);

A typical realization of this process is shown in Figure 6.17 below.

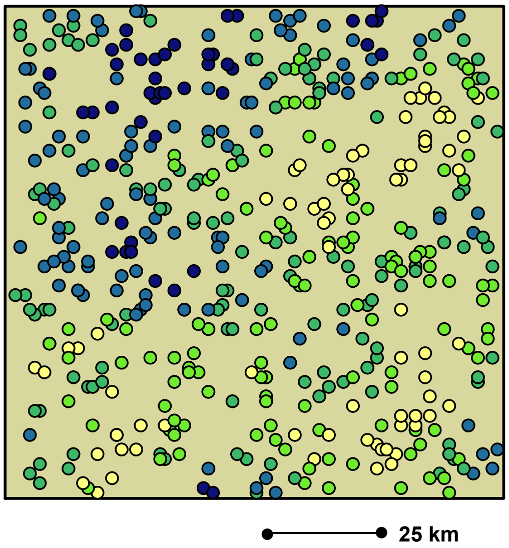


Figure 6.17. Simulated Realization

Notice that spatial correlation is indeed evident at scales smaller than the 25 *km* range shown. Hence the question of interest is whether bandwidths less than this range value do a better job of prediction. The above program, **o_krige_cross_val.m**, was run for each of these 20 simulations. Based on this limited sample, the answer is definitely yes. The cross-validation plot for the realization in Figure 6.17 is shown in Figure 6.18 below:

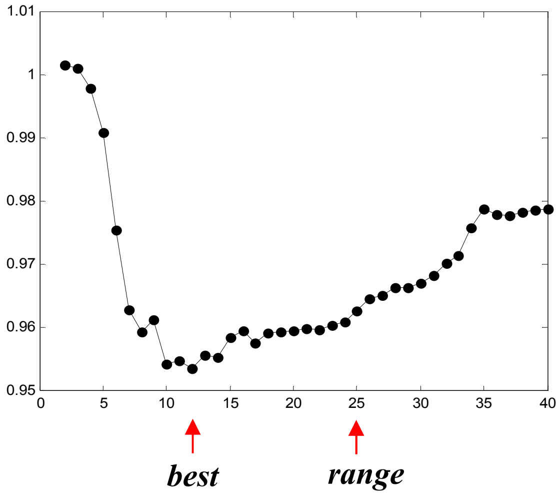


Figure 6.18 Cross Validation Plot

So again the best bandwidth is seen to be about half the true range value. It is also important to note that the estimates of the constant mean and covariogram parameters are actually quite reasonable:

$$\hat{\mu}_n = 9.932\,,\quad (\hat{r},\hat{s},\hat{a}) = (31.638\ ,\ 3.502\ ,0.74557) \tag{6.3.44}$$

So it cannot be argued that this is a result of parameter-estimation error. Indeed, given the moderate overestimation of the true range in this case, one might have expected larger bandwidths to do quite well here.

Finally it should be added that these best bandwidths showed considerable variation over the 20 simulated realizations. The lowest was 5 *km*, and one was actually above the true range (27 *km*), even though the range estimate for this case was almost exactly 25 *km*. So a great deal seems to depend on the spatial structure of the particular pattern realized. But this limited investigation does support the commonly held belief that that best bandwidths for kriging predictions are generally *less* that the estimated range value.