7. General Spatial Prediction Models

Recall that within our general spatial modeling framework, $\{Y(s) = \mu(s) + \varepsilon(s),\ s \in R\}$, the global trend, $\mu(s)$, is assumed to be constant in both Simple and Ordinary Kriging. What this means in practice is that all spatial variations are assumed to be captured by the covariance structure among the residuals, $\varepsilon(s)$. However, the more general kriging models, described as *Universal Kriging* and *Geostatistical Kriging* in Section 6.1.2 above, allow non-constant spatial trend structures. Hence the central task of this section is to develop these more general models in detail.

We begin by developing the types of trend functions to be considered. Recall from the Sudan Rainfall example in Section 2.1 that a number of such trend functions were developed. Here the simplest of these postulated that there was some *linear trend* over space expressible entirely in terms of the spatial coordinates, $s = (s_1, s_2)$, i.e.,

$$(7.1) \qquad \mu(s) = \beta_0 + \beta_1 s_1 + \beta_2 s_2$$

An elaboration of this was given by the *quadratic trend* function,

$$(7.2) \qquad \mu(s) = \beta_0 + \beta_1 s_1 + \beta_2 s_2 + \beta_3 s_1^2 + \beta_4 s_1 s_2 + \beta_5 s_2^2$$

More generally, one may consider *polynomial trend* functions of the form,

$$(7.3) \qquad \mu(s) = \beta_0 + \sum\nolimits_{j=1}^{k} \beta_j s_1^{n_j} s_2^{m_j}$$

where n_j and m_j are nonnegative integer values. Spatial trends in phenomena that vary smoothly over space tend to be well approximated (locally) by such polynomial functions. A good example is *elevation* in hilly terrain. The advantage of these functions is that they require nothing more than the coordinate data in the map itself. Hence the data for constructing such functions is essentially always available. It is for this reason that ARCMAP uses polynomial functions as built-in options for modeling spatial trends (including all polynomials up to order three, i.e., with $n_j + m_j \le 3$, $j = 1,..,k$). A second advantage of these functions is that even though they may involve many spatially nonlinear terms, they are still *linear in parameters*. In other words, such functional forms are linear in all parameters to be *estimated*, namely $\beta_0, \beta_1,.., \beta_k$. So unlike the nonlinear least squares estimation procedure required for the standard variogram models in Section 4.7.2 above, these models can be estimated by *ordinary least squares* (OLS).

But while such functions are sufficiently general to fit many types of spatial trends, they offer little in the way of explanation regarding the nature of these trends. For example, we saw in the introductory California Rainfall example that variables such as "altitude" and "rain shadow" were useful predictors of average rainfall that are not captured by coordinate positions. Even in the case of Vancouver Nickel used for Simple and Ordinary

Kriging above, it may well be that local soil types as well as concentrations of other mineral types might yield better predictions of nickel deposits that simple location coordinates. So, in the spirit of the regression model used in the California Rainfall example, it is of interest to consider linear-in-parameter spatial trend functions involving many possible spatial attributes:

(7.4) $$\mu(s) = \beta_0 + \sum_{j=1}^{k} \beta_j x_j(s)$$

This is seen to include all examples above, where for example, one may have polynomial terms, $x_j(s) = s_1^{n_j} s_2^{m_j}$, or more general spatial attributes such as $x_j(s) =$ "altitude at *s*", or $x_j(s) =$ "copper concentration at *s*" . Moreover, it should be clear that all such trend functions yield spatial models

(7.5) $$\boxed{Y(s) = \beta_0 + \sum_{j=1}^{k} \beta_j x_j(s) + \varepsilon(s) \ , \quad s \in R}$$

which appear to be simply instances of classical linear regression models like the California Rain example. However there is one important difference, namely that the spatial random effects, $\varepsilon(s)$, are allowed to exhibit nonzero covariances. So the key difference here is the *covariance structure* of the residuals. More formally, such models are instances of the *general linear regression model* that allows for nonzero covariances between residuals. Hence to develop spatial prediction models with non-constant trends, we turn first to a consideration of the general linear regression model itself.

7.1 The General Linear Regression Model

To formalize such models in the simplest way, it is essential to use vector representations. We start with a given finite sample,[1] $Y = [Y(s_i) : i = 1,..,n]' = (Y_i : i = 1,..,n)'$ from a spatial stochastic process with global trend of the form (7.5). Let

(7.1.1) $$x(s_i) = [x_0(s_i), x_1(s_i),..., x_k(s_i)] = (x_{i0}, x_{i1},.., x_{ik}) = (1, x_{i1},.., x_{ik})$$

denote the vector of relevant attributes at each sample location, $i = 1,..,n$, where by convention the "attribute", $x_{i0} \equiv 1$, corresponds to the *intercept* term (β_0) in (7.5). With this convention, the integer k denotes the actual number of *spatial attributes* used in the model. If $\beta = (\beta_0, \beta_1,.., \beta_k)$ denotes the corresponding vector of coefficients, then (7.6) can be rewritten as

(7.1.2) $$Y(s_i) = x(s_i)'\beta + \varepsilon(s_i), \quad i = 1,..,n$$

[1] Notice that we now drop the notation, Y_n, used for this sample in Section 6 [in order to avoid confusion with the data point, $Y_n = Y(s_n)$].

This can be further reduced by letting

(7.1.3)
$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{nk:} \end{pmatrix} = \begin{pmatrix} x(s_1)' \\ \vdots \\ x(s_n)' \end{pmatrix} \quad , \quad \varepsilon = \begin{pmatrix} \varepsilon(s_1) \\ \vdots \\ \varepsilon(s_n) \end{pmatrix} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

so that (7.8) can be written in compact matrix form as

(7.1.4)
$$Y = X\beta + \varepsilon$$

Our primary interest for the moment focuses on the *residual vector*, ε. Recall from Section 3 that ε is assumed to be *multi-normally distributed* with mean zero. Moreover, the usual multiple regression model (as for example in the California Rain case), assumes that the individual components of ε are *statistically independent*, and hence have zero covariance. Thus [as in (6.3.10) above] this covariance matrix has the form:

(7.1.5)
$$\text{cov}(\varepsilon) = \begin{pmatrix} \sigma^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{pmatrix} = \sigma^2 I_n$$

In this spatial context, the *classical regression model* can be formally summarized as follows:

(7.1.6)
$$\boxed{Y = X\beta + \varepsilon \quad , \quad \varepsilon \sim N(0, \sigma^2 I_n)}$$

But as in Section 3.3 above, we wish to extend this model by allowing *covariance-stationary* spatial dependencies between the individual components of ε. Hence, while all diagonal elements will continue to have the constant value, σ^2, many of the off-diagonal elements will now be nonzero. If we now let σ_{ij} and ρ_{ij} denote, respectively, the *covariance* and *correlation* between residuals ε_i and ε_j, and recall that [as in expression (3.3.13)], $\rho_{ij} = \sigma_{ij} / \sigma^2$, then the general *covariance matrix, V,* for ε can be written as follows:

(7.1.7)
$$V = \text{cov}(\varepsilon) = \begin{pmatrix} \sigma^2 & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma^2 \end{pmatrix} = \sigma^2 \begin{pmatrix} 1 & \cdots & \rho_{1n} \\ \vdots & \ddots & \vdots \\ \rho_{n1} & \cdots & 1 \end{pmatrix} = \sigma^2 C$$

where C is the corresponding *correlation matrix* for ε. The advantage of this particular representation is that the important variance parameter, σ^2, is made explicit. Moreover, (7.1.7) is now more easily related to the classical case in (7.1.5) where C reduces to the identity matrix, I_n. In this setting, the *general linear regression model* can be formally summarized for our purposes by simply replacing I_n with C in (7.1.6), i.e.,[2]

$$\boxed{Y = X\beta + \varepsilon \quad , \quad \varepsilon \sim N(0,V) = N(0,\sigma^2 C)} \tag{7.1.8}$$

7.1.1 Generalized Least Squares Estimation

Recall that the classical linear regression model is estimated by the method of ordinary least squares. As shown below, this method is directly extendable to the generalized linear regression model. In particular, since the correlation matrix, *C*, is assumed to be *given* (as for example in the Universal Kriging model to be developed below), this model can be reduced to an equivalent classical linear regression model. To develop these results, we begin with the classical linear regression case, and then proceed to generalized linear regression.

OLS Estimators

Given a sample realization, $y = (y_1,..,y_n)'$, of *Y* in model (7.1.6), the method of *ordinary least squares* (OLS) seeks to determine an estimate of the unknown coefficient vector, β, that minimizes the sum of squared deviations between the y_i values and their estimated mean values, $x(s_i)'\beta$. More formally, if the *sum-of-squares function* (*S*) is defined for all possible β values by:

$$S(\beta) = \sum\nolimits_{i=1}^{n}[y_i - x(s_i)'\beta]^2 \;=\; \sum\nolimits_{i=1}^{n}[y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik})]^2 \tag{7.1.9}$$

then the *OLS estimator*, $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1,..,\hat{\beta}_k)'$, is taken to be the minimizer of (7.1.9), i.e.,

$$S(\hat{\beta}) \;=\; \min\nolimits_{\beta} S(\beta) \tag{7.1.10}$$

To determine this estimator, we begin by using (7.1.3) to rewrite this function in matrix terms as,

$$S(\beta) = \;(y - X\beta)'(y - X\beta) \;=\; y'y - 2y'X\beta + \beta'X'X\beta \tag{7.1.11}$$

Notice that this is again a quadratic form in the unknown value, β, that is similar to the mean squared error function, $MSE(\lambda_0)$, in expression (6.2.27) above. So the solutions for these two problems are also similar. In the present case, it is shown in Section A2.7.3 of the Appendix that the solution to (7.1.10) is given by

[2] In Part III of this NOTEBOOK we shall return to this general linear regression model in a somewhat different context. So both covariance representations, V and $\sigma^2 C$, will be useful. For similar treatments see expression (9.11) in Gotway and Waller and section 10.1 in Green (2003).

(7.1.12) $\hat{\beta} = (X'X)^{-1}X'Y$

Notice that we have used the random vector, Y, rather than the realized sample data, *y*, in (7.1.12) in order to view $\hat{\beta}$ as a *random vector* defined for all realizations. [In statistical terms, the distinction here is between $\hat{\beta}$ as an *estimate* of β for a given data set, y, and its role as an *estimator* of β for all sample realizations of *Y*.] Notice also that for this OLS estimator to be well defined, it is necessary that the matrix $X'X$ be *nonsingular*. This will almost surely be guaranteed whenever the number of samples is substantially greater than the number of parameters to be estimated, i.e., whenever $n \gg k+1$.[3] More generally, statistical estimation of any set of parameters can only be reliable when the number of data points well exceeds the number of parameters. In the case of classical linear regression, a common *rule of thumb* is that there be at least 10 samples for every parameter, i.e., $n \geq 10(k+1)$.

Before proceeding to the more general case, it is important to point out that $\hat{\beta}$ is an *unbiased* estimator, since under model (7.1.6), $E(Y) = X\beta$ implies that

(7.1.13) $E(\hat{\beta}) = E[(X'X)^{-1}X'Y] = (X'X)^{-1}X'E(Y) = (X'X)^{-1}X'X\beta = \beta$

What is equally important is the fact that (like the sample mean used in Simple Kriging predictions) this unbiasedness property is *independent of* $\text{cov}(\varepsilon)$. All that is required is that the linear trend specification, $X\beta$, is correct [i.e., that $E(\varepsilon) = 0$]. So in the case of California Rainfall, for example, if the four final variables used were a correct specification of the model, then regardless of possible spatial dependencies among residuals ignored in this model, the estimated beta coefficients would still be unbiased.

GLS Estimators

To extend these results to *generalized linear regression*, we employ the fact that every (nonsingular) covariance matrix can be decomposed in a very simple way. For the covariance matrix, C, in (7.1.7) it is shown in the Appendix [by combining the *Positive Definiteness Property* above expression (A2.7.67) with the *Cholesky Theorem* following expression (A2.7.45)] that there exists a *Cholesky decomposition* of *V*, i.e., there exists a *lower triangular* matrix,

(7.1.14) $$T = \begin{pmatrix} t_{11} & 0 & \cdots & 0 \\ t_{21} & t_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{pmatrix}$$

such that

[3] The symbol "$\gg$" is conventionally used to mean "substantially greater then"

(7.1.15) $C = TT'$

The matrix, *T*, is usually called the *Cholesky matrix* for *C*. While we require no detailed knowledge of such matrices here, it is of interest to point out that the desired Cholesky matrix is easily obtained in MATLAB by the command,[4]

```
>> T = chol(C)′;
```

Perhaps the most attractive feature of the lower triangular matrices is that they are extremely easy to *invert* (and indeed first appeared as part of the classical "Gaussian elimination" method for solving systems of linear equations). Moreover, it is this inverse, T^{-1}, which is directly useful for out purposes. In particular, since *C* is given, we can compute T^{-1} prior to any analysis of model (7.1.8). But if we then premultiply both sides of the equation in (7.1.8) to obtain,

(7.1.16) $$T^{-1}Y = T^{-1}X\beta + T^{-1}\varepsilon$$

and define the following *transformed* quantities,

(7.1.17) $$\tilde{Y} = T^{-1}Y \quad , \quad \tilde{X} = T^{-1}X \quad , \quad \tilde{\varepsilon} = T^{-1}\varepsilon$$

then by (7.1.16) we obtain the following *transformed model*:

(7.1.18) $$\boxed{\tilde{Y} = \tilde{X}\beta + \tilde{\varepsilon}}$$

Moreover, since $\tilde{\varepsilon}$ is a linear transformation of ε, it follows from the *Linear Invariance Theorem* for multi-normal random vectors [in (3.2.22) above] that $\tilde{\varepsilon}$ is also multi-normally distributed with mean zero. But by using (7.1.15) and (3.2.21) [together with the matrix identity $(T^{-1})' = (T')^{-1}$] we can determine the *covariance matrix* for $\tilde{\varepsilon}$ as follows:

(7.1.19) $$\begin{aligned}\text{cov}(\tilde{\varepsilon}) = \text{cov}(T^{-1}\varepsilon) &= T^{-1}\text{cov}(\varepsilon)(T^{-1})' \\ &= T^{-1}(\sigma^2 C)(T')^{-1} \\ &= \sigma^2\, T^{-1}(TT')(T')^{-1} \\ &= \sigma^2\, (T^{-1}T)[(T')(T')^{-1}] \\ &= \sigma^2\, (I_n)\end{aligned}$$

[4] Note the *transpose* operation here. MATLAB for some reason has chosen to produce T' rather than *T*.

So this transformed model is seen to take the form:

(7.1.20) $$\boxed{\tilde{Y} = \tilde{X}\beta + \tilde{\varepsilon} \quad , \quad \tilde{\varepsilon} \sim N(0,\sigma^2 I_n)}$$

Finally, by comparing this with (7.1.6) we see that the generalized linear regression model in (7.1.8) has been transformed into a *classical linear regression model*. This may seem a bit like "magic". But it is simply one of the many consequences of the Linear Invariance Theorem for multi-normal random vectors, and serves to underscore the power of this theorem.

Given this equivalence, we may again use OLS to estimate β. In particular, by using the transformed data, $(\tilde{X},\tilde{Y})$ in (7.1.17), it follows at once from (7.1.12) the desired OLS estimator is given by

(7.1.21) $$\hat{\beta} = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{Y}$$

To distinguish this from the classical linear regression model, it is customary to transform this estimator back into the form of the generalized linear regression model. This amounts simply to substituting the above relations into (7.1.21) [and using the matrix identity $(T\,T')^{-1} = (T')^{-1}(T^{-1}) = (T^{-1})'(T^{-1})$] to obtain

(7.1.22) $$\hat{\beta} = [(T^{-1}X)'(T^{-1}X)]^{-1}(T^{-1}X)'(T^{-1}Y)$$

$$= [X'(T^{-1})'(T^{-1})X]^{-1}X'(T^{-1})'(T^{-1})Y$$

$$= [X'(T\,T')^{-1}X]^{-1}X'(T\,T')^{-1}Y$$

Finally, recalling from (7.1.15) that $C = T\,T'$, it follows that

(7.1.23) $$\hat{\beta} = (X'C^{-1}X)^{-1}X'C^{-1}Y$$

which is entirely independent of Cholesky matrices or transformed models. For our later purposes, it is convenient to rewrite (7.1.23) using the *full covariance matrix*, V, for ε in (7.1.7), i.e.,

(7.1.24) $$\boxed{\hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}Y}$$

The latter version is typically designated as the *generalized least squares* (GLS) estimator of β. However, these two versions are in fact equivalent representations of the GLS estimator since the substitution, $V = \sigma^2 C$, in (7.1.24) shows that

(7.1.25) $$\hat{\beta} = (X'[\sigma^2 C]^{-1} X)^{-1} X'[\sigma^2 C]^{-1} Y = (X' C^{-1} X)^{-1} X' C^{-1} Y$$

Note that this identity also shows that the GLS estimator, $\hat{\beta}$, is *functionally independent* of σ^2. This independence will prove to be enormously useful in later applications.

Note also that even though $\hat{\beta}$ is still dependent on the covariance matrix, *V*, this dependence has no effect on the unbiasedness of $\hat{\beta}$. This should be obvious from its equivalence to an OLS estimator. But in any case, by taking expectations in (7.1.24) we see that

(7.1.26) $$E(\hat{\beta}) = E[(X' C^{-1} X)^{-1} X' C^{-1} Y] = (X' C^{-1} X)^{-1} X' C^{-1} E(Y)$$

$$= (X' C^{-1} X)^{-1} X' C^{-1} (X\beta) = (X' C^{-1} X)^{-1} (X' C^{-1} X)\beta = \beta$$

So regardless of how badly this covariance matrix is misspecified (including $C = I_n$), this by itself creates no biasedness. (Rather it creates *inefficiency* of the estimator, $\hat{\beta}$)

Finally it should be noted that by letting $\tilde{y} = T^{-1} y$, one can also transform the sum-of-squares function, *S*, (by using the same matrix identities above) to obtain:

(7.1.27) $$S(\beta) = (\tilde{y} - \tilde{X}\beta)'(\tilde{y} - \tilde{X}\beta) = (T^{-1}y - T^{-1}X\beta)'(T^{-1}y - T^{-1}X\beta)$$

$$= [T^{-1}(y - X\beta)]'[T^{-1}(y - X\beta)] = (y - X\beta)'(T^{-1})'(T^{-1})(y - X\beta)$$

$$= (y - X\beta)'(TT')^{-1}(y - X\beta) = (y - X\beta)' C^{-1} (y - X\beta)$$

Note again that since C differs from $V = \sigma^2 C$ by a positive scalar, it can be replaced by *V* in (7.1.27) without altering the solution. Both forms are seen to be *weighted* versions of (7.1.11). For this reason, GLS estimation is often referred to as *weighted least squares*. In any case, it should be clear that by minimizing (7.1.27) to obtain (7.1.23) [or (7.1.24)], one need never mention Cholesky matrices or transformed models. But this underlying equivalence between OLS and GLS has many consequences that are not readily perceived otherwise (as will be seen, for example, in Section 7.3.4 below).

7.1.2 Best Linear Unbiasedness Property

Having derived these estimators in terms of standard least squares procedures, it is important to consider their *optimality properties* as estimators. Our objective is to show that these estimators have the same BLU properties of Simple and Ordinary Kriging above. But to do so, it is necessary to extend the notion of Best Linear Unbiased estimation to *vectors* of parameters such as β. Here one might simply argue that we should consider the estimation of each component, β_j, $j = 0,1,..,k$, separately. But it turns out that one can do much better than this. In particular if we were trying to estimate the expected value of a particular component of Y, say $Y_i = Y(s_i)$, then by (7.1.2) this takes the form of a condition mean

$$E[Y_i \mid x(s_i)] = x(s_i)'\beta \tag{7.1.28}$$

Here the standard regression procedure is simply to plug in the beta estimators, $\hat{\beta}$, and use the derived "Yhat" estimator,

$$\hat{Y}_i = x(s_i)'\hat{\beta} = \hat{\beta}_0 + \sum\nolimits_{j=1}^{k} x_j(s_i)\hat{\beta}_j \tag{7.1.29}$$

Hence, even if one were able to establish optimality properties for individual estimators, $\hat{\beta}_j$, there would remain the question as to whether linear combinations of estimators such as in (7.1.28) were still optimal in any sense.

It is for this reason that a much more powerful way to characterize optimality properties of vector estimators is in terms of all possible linear combinations of these estimators. In the present case, observe that if we now focus on GLS estimators and consider any linear combination of the unknown β vector, say $a'\beta$, then by (7.1.24) the corresponding estimator, $a'\hat{\beta}$, takes the form

$$a'\hat{\beta} = a'(X'V^{-1}X)^{-1}X'V^{-1}Y = [a'(X'V^{-1}X)^{-1}X'C^{-1}]Y = \theta_a'Y \tag{7.1.30}$$

where $\theta_a' = a'(X'V^{-1}X)^{-1}X'V^{-1}$. But since (a, X, V) are all *known* values, this estimator is indeed seen to be a *linear estimator* of $a'\beta$, i.e., a linear function of the Y vector (in a manner completely analogous to Simple and Ordinary Kriging weights). Moreover, by the argument in (7.1.26) it follows at once that

$$E(a'\hat{\beta}) = a'E(\hat{\beta}) = a'\beta \tag{7.1.31}$$

Hence the "plug-in" estimator, $a'\hat{\beta}$, is seen to be a *linear unbiased estimator* of $a'\beta$, for *all possible choices* of a. But the real power of this "linear compound" approach is that it provides natural definition of *best* linear unbiased estimators in this vector setting. In particular, we now say that $\hat{\beta}$ is a *Best Linear Unbiased* (BLU) estimator of β, if and only if in addition to (7.1.30) and (7.1.31) is also true that the variance of $a'\hat{\beta}$ is *smallest* among all such linear unbiased estimators. More formally, if we now denote the class of all *linear unbiased estimators*, $\tilde{\beta}$, of β by

$$LU(\beta) = \left\{\tilde{\beta} = \tilde{\beta}(X, V, Y) : [a'\tilde{\beta} = \tilde{\theta}_a'Y] \& [E(a'\tilde{\beta}) = a'\beta] \ , a \in \mathbb{R}^{k+1}\right\} \tag{7.1.32}$$

then $\hat{\beta}$ is said to be a *Best Linear Unbiased* (BLU) estimator of β if and only if for all linear compounds, $a \in \mathbb{R}^{k+1}$,

$$\boxed{\operatorname{var}(a'\hat{\beta}) = \min\{\operatorname{var}(a'\tilde{\beta}) : \tilde{\beta} \in LU(\beta)\}} \tag{7.1.33}$$

While this definition looks rather ambitious, it is shown in the Appendix (see the first subsection of Section A2.8.3) that the unique estimator in $LU(\beta)$ satisfying this minimum variance condition for all $a \in \mathbb{R}^{k+1}$ is precisely the *GLS estimator* in (7.1.24).

7.1.3 Regression Consequences of Spatially Dependent Random Effects

As discussed in detail in Section 3 above, our primary interest in *GLS* models is to allow covariance structures to reflect *spatially dependent* random effects. We are now in a position to see the consequences of such effects in more detail. To do so, we begin with the simplest possible spatial regression model, where such effects can be seen explicitly. We then examine these effects in a more complex setting by means of simulation.

Simple Constant-Mean Example.

Here we start with the simplest possible spatial regression model with a constant mean, i.e., with no "explanatory variables" at all:

$$\boxed{Y(s) = \mu + \varepsilon(s) \ , \ s \in \{s_1,..,s_n\} \subset R} \tag{7.1.34}$$

In this context, suppose we ignore possible spatial dependencies among residuals, and assume simply that the residuals in (7.1.34) are independent, say $\varepsilon(s) \underset{iid}{\sim} N(0,\sigma^2)$. Then in matrix form, we have the regression model:

$$Y = \mu \cdot 1_n + \varepsilon \ , \quad \varepsilon \sim N(0, \sigma^2 I_n) \tag{7.1.35}$$

where in this case, $X = 1_n = (1,..,1)'$ and $\beta = \mu$. Hence for this case it follows from (7.1.24) that the *BLU* estimator of μ is given by:

$$\boxed{\hat{\mu} = (1_n' 1_n)^{-1} 1_n' Y = (n)^{-1}\left(\sum\nolimits_{i=1}^n Y_i\right) = \tfrac{1}{n}\sum\nolimits_{i=1}^n Y_i = \overline{Y}} \tag{7.1.36}$$

which is of course simply the *sample mean* . [Recall also expressions (6.3.11) and (6.3.12)]. Moreover, recall from (3.1.19) that the variance of this estimator must be given by

$$\operatorname{var}(\overline{Y}) = \frac{\sigma^2}{n} \tag{7.1.37}$$

So all inferences about the true value of μ will be based on the estimator, $\bar{Y}$, and its variance in (7.1.37).

But suppose that in reality there are *positive spatial dependencies* among the residuals in (7.1.35) so that in fact the covariance of ε has the form,

$$(7.1.38) \qquad \mathrm{cov}(\varepsilon) = \begin{bmatrix} \mathrm{cov}(\varepsilon_1,\varepsilon_1) \cdots \mathrm{cov}(\varepsilon_1,\varepsilon_n) \\ \vdots \quad \ddots \quad \vdots \\ \mathrm{cov}(\varepsilon_n,\varepsilon_1) \cdots \mathrm{cov}(\varepsilon_n,\varepsilon_n) \end{bmatrix} = \begin{pmatrix} \sigma^2 \cdots \sigma_{1n} \\ \vdots \ \ddots \ \vdots \\ \sigma_{n1} \cdots \sigma^2 \end{pmatrix} \geq \sigma^2 I_n$$

with $\sigma_{ij} > 0$ for many distinct (i,j) pairs. Then, in a manner similar to expression (4.10.3) above, it follows that since $\mathrm{cov}(Y) = \mathrm{cov}(\varepsilon)$, the true variance of $\bar{Y}$ is given by

$$(7.1.38) \qquad \mathrm{var}(\bar{Y}) = \mathrm{var}\left(\tfrac{1}{n}\sum\nolimits_{i=1}^{n} Y_i\right) = \tfrac{1}{n^2}\left(\sum\nolimits_{i=1}^{n}\mathrm{var}(Y_i) + \sum\nolimits_i \sum\nolimits_{j\neq i}\mathrm{cov}(Y_i,Y_j)\right)$$

$$= \tfrac{1}{n^2}\sum\nolimits_{i=1}^{n}\mathrm{var}(Y_i) + \tfrac{1}{n^2}\sum\nolimits_i\sum\nolimits_{j\neq i}\mathrm{cov}(Y_i,Y_j) \ = \tfrac{1}{n^2}\sum\nolimits_{i=1}^{n}\sigma^2 + \tfrac{1}{n^2}\sum\nolimits_i\sum\nolimits_{j\neq i}\sigma_{ij}$$

$$= \tfrac{1}{n^2}(n\sigma^2) + \tfrac{1}{n^2}\sum\nolimits_i\sum\nolimits_{j\neq i}\sigma_{ij} \ = \boxed{\frac{\sigma^2}{n} + \tfrac{1}{n^2}\sum\nolimits_i\sum\nolimits_{j\neq i}\sigma_{ij}}$$

which, in the presence of many *positive* spatial dependencies, implies that,

$$(7.1.39) \qquad \mathrm{var}(\bar{Y}) \ >> \ \frac{\sigma^2}{n}$$

and hence that *standard deviation*, $\sigma(\bar{Y})$, is much larger than assumed, i.e.,

$$(7.1.40) \qquad \boxed{\sigma(\bar{Y}) \ >> \ \frac{\sigma}{\sqrt{n}}}$$

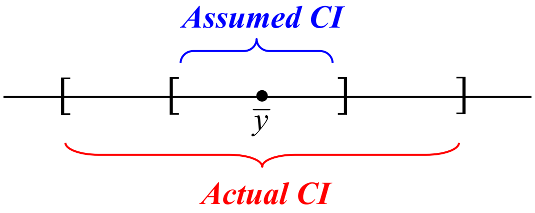
This means, for example, that if we consider a 95% confidence interval for the true mean, μ, then the *actual* interval is given by

$$(7.1.41) \qquad CI_{actual} = [\bar{Y} \pm (1.96)\sigma(\bar{Y})]$$

rather than the *assumed* interval

$$(7.1.42) \qquad CI_{assumed} = \left[\bar{Y} \pm (1.96)\tfrac{\sigma}{\sqrt{n}}\right]$$

So for any given estimate, $\bar{y}$, this implies from (7.1.40) that the actual confidence intervals for μ are much *larger* than those calculated, as depicted schematically below:



Thus if such spatial dependencies are not accounted for, then the results obtained will tend to look "too good". It is this type of *false significance* that motivates the need to remove the effects of spatial dependencies in residuals before attempting to draw statistical inferences.

More Complex Example.

As one illustration of a more complex spatial example, consider a spatial regression model with data points, $\{s_i = (x_{1i}, x_{2i}) : i = 1,..,100\}$, forming a 10×10 unit grid on the plane, and with Y_i defined by a linear function of these grid points,

$$(7.1.43) \qquad \boxed{Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + u_i \quad , \quad i = 1,..,100}$$

with specific parameter values, $\beta_0 = 1$, $\beta_1 = .04$, and $\beta_2 = .08$. Suppose moreover that the residuals $\{u_i : i = 1,..,100\}$ are part of an underlying covariance-stationary spatial stochastic process with covariogram, $C(h)$, parameterized by $[r = 5, s = 1, a = 0]$, as shown in Figure 7.1 below.

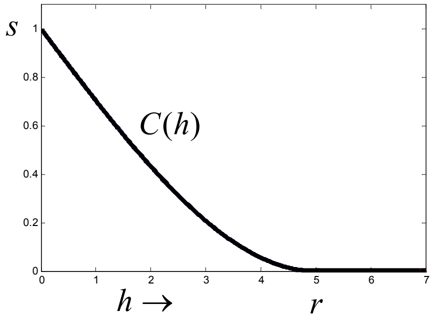


Figure 7.1. Example Covariogram

Given this model, one can in principle calculate the theoretical estimates and standard errors for any given set of data $\{y_i : i = 1,..,100\}$ under the (OLS) assumption of independent errors, and under the true (GLS) model itself. But it is more instructive to simulate this model many times and compare the OLS and GLS estimates of beta parameters. In Table 7.1 below, the average results of 100 simulations are shown, where the "GLS Est" column shows the average GLS estimates of each beta parameter, the "GLS Std Err" column shows the corresponding average standard errors of these estimates, and similarly for the OLS columns.

	GLS Est	GLS Std Err	OLS Est	OLS Std Err
const	**0.9284**	**0.4802**	**0.9156**	**0.2396**
***X*1**	**0.0564**	**0.0565**	**0.0568**	**0.0289**
***X*2**	**0.0897**	**0.0565**	**0.0934**	**0.0289**

Table 7.1. Average Values for 100 Simulations

Notice first that while the *GLS* estimates are on average slightly better than the *OLS* estimates, both sets of estimates are unbiased (regardless of the true covariance) and should tend to be roughly the same. The real difference is in the estimated *standard errors* for each of these models. Here it is clear that the *GLS* estimates are about twice as large as the *OLS* estimates. So as a direct parallel to expression (7.1.40) above, it is now clear that by ignoring the true spatial dependencies, *OLS* is severely underestimating the true standard deviations. So the confidence intervals on true beta values are again much tighter than they should be.

To illustrate the consequences of such underestimation, we consider one specific instance of the simulations above (number 47 in the set of 100 simulations). Here the specific estimates and standard errors are shown in Table 7.2 below:

	GLS Est	GLS Std Err	OLS Est	OLS Std Err
const	**1.5197**	**0.6754**	**2.0143**	**0.2669**
***X*1**	**-0.0062**	**0.0789**	**-0.1228**	**0.0352**
***X*2**	**0.0913**	**0.0789**	**0.0981**	**0.0352**

Table 7.2. Specific Values for a "Bad Case"

This example illustrates a particularly bad case in which the estimates of β_1 actually have the wrong sign in both *OLS* and *GLS*. But if we display the 95% confidence intervals for each case, we can see a substantial difference in the conclusions reached. First for the *GLS* case we have:

(7.1.44) $$\beta_1 = -.0062 \pm (1.96)(.0789) \;\Rightarrow\; \boxed{\beta_1 \in [-.1607,.1485]}$$

In particular, since the true value, .04, is contained it this interval, this value cannot be ruled out by these results. More generally, since zero is also contained in this interval, it can certainly not be concluded that x_1 is negatively related to y. On the other hand, since the corresponding *OLS* confidence interval is given by:

(7.1.45) $$\beta_1 = -.1228 \pm (1.96)(.0352) \;\Rightarrow\; \boxed{\beta_1 \in [-.1917,-.0537]}$$

it must here be concluded that $\beta_1 \le -.0537$, and thus that x_1 is *significantly negatively related* to *y*. This is precisely the type of *false significance* that one seeks to avoid by allowing for the possibility of spatially-dependent errors in estimation procedures.

Given this general linear regression framework, together with our present emphasis on modeling spatially-dependent errors, the task remaining is to develop specific methods for spatial prediction within this setting. Recall from our general classification of Kriging models in Section 6.1.2 that the method for doing so is known as *Universal Kriging*. Hence we now develop this spatial prediction model in more detail.

7.2 The Universal Kriging Model

Recall from (6.1.10) and (6.1.11) that the basic probability model underlying Universal Kriging is essentially the general linear regression model in (7.1.2) above. Within this probabilistic framework, the task of spatial prediction (as in both Simple and Ordinary Kriging) is to determine a *BLU predictor* for values, $Y(s_0)$, at locations s_0 not in the given sample data set, $Y_n = [Y(s_i) : i = 1,..,n]'$. As we shall see in the next section, this essentially amounts to an appropriate extension of the analysis for Ordinary Kriging. Following this development we derive the appropriate standard error of prediction for Universal Kriging. As with Simple Kriging, our main interest in Universal Kriging is that it provides the simplest setting within which one can include the types of spatial trend models developed above. Because this model is included as part of ARCMAP, we also outline the procedure for implementing this model. However, the main role of this model for our present purposes is to serve as an introduction to *Geostatistical Regression and Kriging*, as developed in Section 7.3 below.

7.2.1 Best Linear Unbiased Prediction

Here we again start with a given prediction set, $S(s_0) = \{s_i : i = 1,..,n_0\} \subseteq \{s_1,..,s_n\}$, for s_0 together with corresponding *prediction samples*, $Y = [Y(s_i) : i = 1,..,n_0]'$.[5] Moreover, by

[5] Note that we have now returned to the convention that Y_n denotes the *full sample vector* and *Y* is the *prediction sample vector* for s_0. As with Ordinary Kriging, *both* random vectors will be used here.

again appealing to the *linear prediction hypothesis*, it is assumed that the desired predictor, $\hat{Y}_0 = \hat{Y}(s_0)$, is of the form:

$$(7.2.1) \qquad \hat{Y}_0 = \lambda_0' Y$$

for some appropriate weight vector, $\lambda_0 = (\lambda_1,..,\lambda_{n_0})'$. Turning next to the unbiasedness condition, it follows from condition (6.3.14) for Ordinary Kriging, that this unbiasedness condition again takes the basic form:

$$(7.2.2) \qquad 0 = E(e_0) = E(Y_0 - \hat{Y}_0) = E[Y_0 - \lambda_0' Y] = E[Y_0] - \lambda_0' E(Y)$$

But now these expectations are more complex. By (7.1.2) we see that

$$(7.2.3) \qquad E(Y_0) = E[Y(s_0)] = x(s_0)'\beta$$

and similarly that

$$(7.2.4) \qquad \lambda_0' E(Y) = \sum\nolimits_{i=1}^{n_0} \lambda_{0i} E[Y(s_i)] = \sum\nolimits_{i=1}^{n_0} \lambda_{0i} x(s_i)'\beta$$

So to write (7.2.2) more explicitly, it is convenient to introduce the following notational conventions. First let the vector of attributes at the prediction location, s_0, be denoted by

$$(7.2.5) \qquad x(s_0) = x_0 = \begin{pmatrix} 1 \\ x_{01} \\ \vdots \\ x_{0k} \end{pmatrix}$$

and similarly, let the matrix of attributes for locations in $S(s_0)$ be denoted by

$$(7.2.6) \qquad X_0 = \begin{pmatrix} x(s_1)' \\ \vdots \\ x(s_{n_0})' \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n_0 1} & \cdots & x_{n_0 k} \end{pmatrix}$$

Then since

$$(7.2.7) \qquad \sum\nolimits_{i=1}^{n_0} \lambda_{0i} x(s_i)'\beta = (\lambda_{01},..,\lambda_{0n_0}) \begin{pmatrix} x(s_1)' \\ \vdots \\ x(s_{n_0})' \end{pmatrix} \beta = \lambda_0' X_0 \beta$$

it follows that (7.2.2) can be written in an explicit compact form as

(7.2.8) $$0 = x_0'\beta - \lambda_0' X_0 \beta = (x_0 - X_0'\lambda_0)'\beta$$

But since this *unbiasedness condition* is required to hold for all β, it should be clear that this is only possible if $x_0 - X_0'\lambda_0 = 0$, or equivalently, if and only if

(7.2.9) $$\boxed{X_0'\lambda_0 = x_0}$$

Turning finally to the *efficiency condition*, the argument in (6.3.17) for Ordinary Kriging can now be extended by using (7.2.8) to show that prediction error variance continues to be the same as residual *mean squared error*:

(7.2.10) $$\begin{aligned} \text{var}(e_0) = E(e_0^2) = E[(Y_0 - \hat{Y}_0)^2] &= E[(Y_0 - \lambda_0' Y)^2] \\ &= E\left([(x_0'\beta + \varepsilon_0) - \lambda_0'(X_0\beta + \varepsilon)]^2\right) \\ &= E\left([(x_0' - \lambda_0' X_0)\beta + (\varepsilon_0 - \lambda_0'\varepsilon)]^2\right) \\ &= E[(\varepsilon_0 - \lambda_0'\varepsilon)^2] = \boxed{MSE(\lambda_0)} \end{aligned}$$

But since all covariances are given, it follows by setting $V_0 = \text{cov}(Y)$ that (as with both Simple and Ordinary Kriging) prediction error variance must again be given by,

(7.2.11) $$\text{var}(e_0) = \sigma^2 - 2c_0'\lambda_0 + \lambda_0' V_0 \lambda_0$$

Hence the *optimal weight vector*, $\hat{\lambda}_0$, for the case of Universal Kriging must be the solution to the following *constrained minimization problem*:

(7.2.12) $$\boxed{\textit{minimize}:\ \sigma^2 - 2c_0'\lambda_0 + \lambda_0' V_0 \lambda_0 \qquad \textit{subject to}:\ X_0'\lambda_0 = x_0}$$

At this point, it should be clear that Ordinary Kriging is simply a special case of Universal Kriging. Indeed, if one eliminates all explanatory variables and keeps only the intercept term in (7.1.2), then by (7.2.5) and (7.2.6), x_0 reduces to 1 and X_0' reduces to $1_{n_0}'$, so that the constraint in (7.2.12) reduces to $1_{n_0}'\lambda_0 = 1$,which is precisely (6.3.18). This is a consequence of the fact that under the assumptions of Ordinary Kriging, this reduced model implies that $\beta_0 = \mu$, i.e.,

(7.2.13) $$Y(s) = \beta_0 + \varepsilon(s) \Rightarrow E[Y(s)] = \beta_0 = \mu$$

Turning now to the solution, $\hat{\lambda}_0$, of (7.2.12), it is shown in the Appendix [expression (A2.8.58)] that

$$\text{(7.2.14)} \qquad \boxed{\hat{\lambda}_0 \;=\; V_0^{-1}X_0(X_0'V_0^{-1}X_0)^{-1}(x_0 - X_0'V_0^{-1}c_0) + V_0^{-1}c_0}$$

By substituting $\hat{\lambda}_0$ into (7.2.1) we then obtain the following *BLU predictor* of $Y_0 = Y(s_0)$ for Universal Kriging [see also expression (A2.8.59) in the Appendix]:

$$\text{(7.2.15)} \qquad \hat{Y}_0 = x_0'(X_0'V_0^{-1}X_0)^{-1}X_0'V_0^{-1}Y + c_0'V_0^{-1}[Y - X_0(X_0'V_0^{-1}X_0)^{-1}X_0'V_0^{-1}Y]$$

While this solution appears to be even more complex than expression (6.3.20) for the Ordinary Kriging case, it turns out to have an equally simple interpretation. To show this, we start by noting that as a parallel to (6.2.21), if we now estimate β based solely on the prediction sample, $Y = [Y(s_i) : i = 1,..,n_0]$, for Y_0 (with attribute data, X_0, and covariance matrix, V_0) then it follows from (7.1.24) that the resulting *GLS estimator* of β, say $\hat{\beta}_{n_0}$, must be given by,

$$\text{(7.2.16)} \qquad \boxed{\hat{\beta}_{n_0} = (X_0'V_0^{-1}X_0)^{-1}X_0'V_0^{-1}Y}$$

Moreover, by the results of Section 7.1.2 above, this must be the *BLU estimator* of β based on this sample data. But by substituting (7.2.16) into (7.2.15), we then see that $\hat{Y}_0$ has the simpler form,

$$\text{(7.2.17)} \qquad \hat{Y}_0 = x_0'\hat{\beta}_{n_0} + c_0'V_0^{-1}(Y - X_0\hat{\beta}_{n_0})$$

Finally, since the last expression in brackets is simply the vector of *estimated residuals,*

$$\text{(7.2.18)} \qquad \hat{\varepsilon} = Y - X_0\hat{\beta}_{n_0}$$

generated by $\hat{\beta}$, it follows that $\hat{Y}_0$ takes the following form:

$$\text{(7.2.19)} \qquad \boxed{\hat{Y}_0 \;=\; x_0'\hat{\beta}_{n_0} \;+\; c_0'V_0^{-1}\hat{\varepsilon}}$$

So as with Ordinary Kriging, the construction of Universal Kriging predictors is seen to have a appealing *two-step interpretation*:

(***i***). Construct the BLU estimator, $\hat{\beta}_{n_0}$, of β based on the prediction sample data, Y, as in (7.2.16).

(***ii***). Use the sample residuals, $\hat{\varepsilon}$, in (7.2.18) to obtain the Universal Kriging predictor, $\hat{\varepsilon}_0 = c_0'V_0^{-1}\hat{\varepsilon}$, of ε_0 and set $\hat{Y}_0 = x_0'\hat{\beta}_{n_0} + \hat{\varepsilon}_0$.

But as with Ordinary Kriging, it can also be argued that if β characterizes the *global* trend over the entire region, *R*, then a better estimate can be obtained by using the GLS estimator,

(7.2.20) $$\hat{\beta}_n = (X'V^{-1}X)^{-1}X'V^{-1}Y_n$$

based on the *full* set of samples, Y_n, with attribute data, X. It is this modified procedure that constitutes the most commonly used form of *Universal Kriging*.[6] To formalize this procedure, it thus suffices to modify the two steps above as follows:

(1). Construct the BLU estimator, $\hat{\beta}_n$, of β based on the full sample data, Y_n, as in (7.2.20).

(2). Use the sample residuals, $\hat{\varepsilon} = Y - X_0\hat{\beta}_n$, to obtain the Universal Kriging predictor, $\hat{\varepsilon}_0 = c_0'V_0^{-1}\hat{\varepsilon}$, of ε_0 and set $\hat{Y}_0 = x_0'\hat{\beta}_n + \hat{\varepsilon}_0$.

7.2.2 Standard Error of Prediction

As with Ordinary Kriging, one can obtain prediction error variance for the optimal weight vector, $\hat{\lambda}_0$, by substituting (7.2.14) into (7.2.11). As is shown in the Appendix [see expression (A2.8.69)] this yields the follow explicit expression for *prediction error variance* in the general case of Universal Kriging:

(7.2.21) $$\hat{\sigma}_0^2 = (\sigma^2 - c_0'V_0^{-1}c_0) + (x_0 - X_0'V_0^{-1}c_0)'(X_0'V_0^{-1}X_0)^{-1}(x_0 - X_0'V_0^{-1}c_0)$$

Paralleling the interpretation $\hat{\sigma}_0^2$ for Ordinary Kriging, the first bracketed expression in (7.2.21) is again prediction error variance for Simple Kriging, and the second expression is again positive. This second term now accounts for the additional variance created by estimating β internally. Finally, the resulting *standard error of prediction* for Universal Kriging is by definition the square root of (7.2.21), i.e.,

(7.2.22) $$\boxed{\hat{\sigma}_0 = \sqrt{(\sigma^2 - c_0'V_0^{-1}c_0) + (x_0 - X_0'V_0^{-1}c_0)'(X_0'V_0^{-1}X_0)^{-1}(x_0 - X_0'V_0^{-1}c_0)}}$$

[6] As with Ordinary Kriging, there are again arguments for using the *local* version in [(*i*),(*ii*)] above. In fact, many treatments of Universal Kriging implicitly use this local version, as for example in Section 5.3.3 of Schabenberger and Gotway (2005).

7.2.3 Implementation of Universal Kriging

In many respects, the implementation of Universal Kriging closely parallels that of Ordinary Kriging. But the key difference is that when global trends are *not constant*, the fundamental identity between differences of *Y*-values and ε-values in (4.8.4) breaks down. So prior estimation of the variogram becomes quite problematic in this more general setting. Indeed, this is the primary motivation for the method of *Geostatistical Kriging* to be developed in Section 7.3 below. The most common procedure here is to start with *OLS estimation* of β, which assumes all covariances are zero. This will yield a set of OLS residuals that can then be used to estimate a spherical variogram. Given this estimate, the procedure closely follows that of Ordinary Kriging.

With these preliminary observations, the implementation procedure for Universal Kriging can be specified as follows. We again start with a given set of *sample data*, $y_n = (y(s_i){:}\, i = 1,..,n)'$ in *R*, where each y_i is taken to be a realization of the corresponding random variable, $Y(s_i)$, in a *sample vector*, $Y_n = [Y(s_i) : i = 1,..,n]'$. This sample vector, Y_n, is now hypothesized to satisfy the generalized linear regression model in (7.1.8) with *attribute data*, *X*, and *covariance matrix*, V. In this context, we again consider the prediction of $Y_0 = Y(s_0)$, at a given location, $s_0 \in R$. This prediction is carried out through the following series of steps:

Step 1. OLS Estimation

Construct an OLS estimate,

$$\hat{\beta}_{OLS} = (X'X)^{-1}X'y_n \qquad (7.2.23)$$

of β and form the corresponding residuals

$$\hat{\varepsilon}_{OLS} = y - X\hat{\beta}_{OLS} \qquad (7.2.24)$$

Step 2. Covariance Estimation

Using these residuals, $\varepsilon_{OLS} = [\varepsilon_i = \varepsilon(s_i) : i = 1,..,n]'$, proceed as in Step 2 for Simple Kriging by estimating a spherical variogram, $\gamma(h;\hat{r},\hat{s},\hat{a})$, and associated covariogram,

$$\hat{C}(h) = \hat{s} - \gamma(h;\hat{r},\hat{s},\hat{a}) \qquad (7.2.25)$$

as in (6.2.65). Then using the identity

$$\hat{\sigma}_{ij} \;=\; \widehat{\mathrm{cov}}[\varepsilon(s_i),\varepsilon(s_j)] \;=\; \hat{C}(\| s_i - s_j \|) \qquad (7.2.26)$$

as in (6.2.66), construct an estimate:

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

of the *full-sample covariance matrix*, V .

Step 3. GLS Estimation

Now use (7.2.26) to construct a final *GLS estimate* of β as in (7.2.20),

(7.2.28) $$\hat{\beta}_n = (X'\hat{V}^{-1}X)^{-1}X'\hat{V}^{-1}Y_n$$

with $\hat{V}$ replacing V in (7.2.20).

Step 4. Selection of a Prediction Set for *Y*(*s*0)

Given the development of prediction set selection in Section 6.4 above, we can now consider this selection problem more explicitly for Universal Kriging. In particular, we now assume that the appropriate prediction set, $S(s_0)$, is defined by an appropriate *bandwidth*, h_0, as follows,

(7.2.29) $$S(s_0) = \{s_i \in S_n : \| s_0 - s_i \| < h_0\}$$

where $S_n = \{s_1,..,s_n\}$ is again the full sample set of locations. Ideally this bandwidth should be selected by a *cross-validation procedure* such as in Section 6.4. But given the computation intensity of such procedures, we here assume that h_0 is selected simply by a visual inspection of the mapped data surrounding site, s_0.

However, there is one additional requirement that must be met by prediction sets, $S(s_0) = \{s_1,..,s_{n_0}\}$, in the case of Universal Kriging. Recall that if the *attribute vector* at s_0 is denoted by x_0 as in (7.2.5), then the *unbiasedness condition* for Universal Kriging in (7.2.9) requires that

(7.2.30) $$X_0'\lambda_0 = x_0$$

where the transpose, X_0', of *prediction attribute matrix*, X_0, in (7.2.6) has $k+1$ rows (one for each attribute) and n_0 columns (one for each prediction point). But (7.2.30) formally requires that the given ($k+1$)-vector, x_0, of attributes at s_0 be a linear combination of the columns of X_0'. This can only be guaranteed in general if $n_0 \geq k+1$. Moreover, to avoid trivial solutions, we require that $n_0 \geq k+2$. [7]

Step 5. Construction of Estimated Prediction Covariances

Given a prediction set, $S(s_0) = \{s_1,..,s_{n_0}\}$, one can then use (7.2.26) above to construct estimates of the set of covariances,

$$(7.2.31) \qquad \hat{C}_0 = \begin{pmatrix} \hat{\sigma}^2 & \hat{c}_0{}' \\ \hat{c}_0 & \hat{V}_0 \end{pmatrix}$$

relevant for prediction of $Y(s_0)$ [as in (6.3.33)]. These can in turn be to krige residuals as in the second step of the basic two-step procedure for Universal Kriging above. Here the procedure is as follows:

Step 6. Kriging Prediction Residuals at *s*₀

If the *prediction sample data* relevant for s_0 is denoted by $y = (y_1,..,y_{n_0})'$, and if the corresponding *prediction residuals* are estimated by,

$$(7.2.32) \qquad \hat{\varepsilon} = y - X_0\hat{\beta}_n$$

then the residual, $\hat{\varepsilon}_0$, predicted at s_0 can be constructed by Simple Kriging of $\hat{\varepsilon}$ as follows:

$$(7.2.33) \qquad \hat{\varepsilon}_0 = c_0'V_0^{-1}\hat{\varepsilon}$$

Step 7. Constructing the Prediction of *Y*(*s*₀)

Finally, (7.2.33) can be combined with (7.2.28) to obtain the desired prediction of the unobserved value, $Y(s_0) = x_0'\beta + \varepsilon_0$, at s_0, namely

$$(7.2.34) \qquad \hat{Y}_0 = x_0'\hat{\beta}_n + \hat{\varepsilon}_0$$

[7] More precisely, x_0 is required to lie in the *span* of theses column vectors. Hence there must be at least $k+1$ linearly independent columns of X_0' to insure this condition. But if this number were exactly $n_0 = k+1$ then λ_0 would be *uniquely* determined by $\lambda_0 = (X_0')^{-1}x_0$. So for nontrivial solutions one must require that $n_0 \geq k+2$.

Step 8. Prediction Intervals

By combining this with the corresponding estimate of *prediction standard error*,

$$\text{(7.2.35)} \qquad \hat{\sigma}_0 = \sqrt{(\hat{\sigma}^2 - \hat{c}_0' \hat{V}_0^{-1}\hat{c}_0) + (x_0 - X_0'\hat{V}_0^{-1}\hat{c}_0)'(X_0'\hat{V}_0^{-1}X_0)^{-1}(x_0 - X_0'\hat{V}_0^{-1}\hat{c}_0)}$$

one can use the pair $(\hat{Y}_0, \hat{\sigma}_0)$ to construct prediction intervals for $Y(s_0)$. As in (6.2.63), the *default interval* takes the form:

$$\text{(7.2.36)} \qquad \boxed{[\hat{Y}_0 - (1.96)\hat{\sigma}_0,\ \hat{Y}_0 + (1.96)\hat{\sigma}_0]}$$

7.3 Geostatistical Regression and Kriging

As mentioned at the beginning of Section 7.2.3 above, the estimation of variograms for Universal Kriging is somewhat problematic. In particular, observe that the OLS residuals in (7.2.24) used for estimation of variograms are generally *not consistent* with the final GLS residuals in (7.2.32). So if the variogram were re-estimated on the basis of these residuals, then generally this would not agree with the variogram used. This inconsistency is simply ignored in the implementation of Universal Kriging outlined above, and hence renders this procedure somewhat *ad hoc*. To be more precise, if now denote the *parameter vector* for the spherical variogram by

$$\text{(7.3.1)} \qquad \theta = (r, s, a) \quad ,$$

then one the one hand, if θ were *known* (as is implicit in the "known covariance" assumption of Universal Kriging) one could employ GLS estimation to determine $\hat{\beta}$. On the other hand, if β were *known*, then the residual "data", $\varepsilon = Y - X\beta$, could be used to construct a consistent estimate, $\hat{\theta}$, of the variogram parameters, θ. Hence the real difficulty here is trying to obtain *simultaneous* estimates, $(\hat{\beta}, \hat{\theta})$, of these two sets of parameters. In Schabenberger and Gotway (2005, p.257)) this circular argument is aptly described as the "cat and mouse game of Universal Kriging". While it is possible to reformulate this entire estimation problem in terms of more general maximum-likelihood methods,[8] a more practical approach is simply to construct an *iterative estimation procedure* in which each parameter vector is estimated given some current value of the other. It is this procedure that we now develop in more detail.[9]

[8] For further discussion of such methods, see Section 9.2.1 in Waller and Gotway (2004). Here it should also be noted that a maximum-likelihood estimation approach of this type will be developed to estimate *spatial autoregressive models* in Part III of this NOTEBOOK.

[9] This procedure is also developed in Section 9.2.11 in Waller and Gotway (2004), where it is designated as the *Iteratively Re-Weighted Generalized Least Squares* (IRWGLS) procedure. A less formal presentation of the same idea is given in [BG], p.189.

Before doing so, it is important to emphasize that the type of spatial model developed here has uses other than simply predicting values of Y at unobserved locations. A good example is the California Rainfall study, already used to motivate the present class of more general spatial trend functions. In this study, the main focus was on identifying spatial attributes that are significant predictors of rainfall at each data location. While one could also attempt to predict rainfall levels at locations not in the data set, this was not the main objective. Hence it is useful to distinguish between two types of spatial applications here. We begin with a general linear regression model as in (7.1.8), where it is now assumed that the covariance matrix, V, is generated by an underlying covariogram with parameter vector, θ, in (7.3.1), which we now write explicitly as,

$$\boxed{Y = X\beta + \varepsilon \quad , \qquad \varepsilon \sim N[0, V(\theta)]} \tag{7.3.2}$$

This is of course precisely the type of model postulated for Universal Kriging above. However, since the iterative estimation procedure developed below differs from the *implementation* of Universal Kriging as developed in Section 7.2.3, it is convenient to distinguish between these two models. Hence we now designate model (7.3.2) [together with its iterative implementation developed below] as a *Geostatistical Regression model*. In the California Rainfall example, such a model might well be used to incorporate possible spatial dependencies between rainfall in cities close to one another. The emphasis here is on estimating β in a manner that will allow proper statistical inferences to be drawn about each of its components. On the other hand, such a model might also be used for prediction purposes. Hence when such geostatistical regression models are used for spatial prediction, they will be designated as *Geostatistical Kriging models*.[10]

With these preliminary observations, we can now develop an implementation of both these models. As in Section 7.2.3, we start with a given set of *sample data*, $y = (y(s_i){:}\, i = 1,..,n)'$ in R, where each y_i is taken to be a realization of the corresponding random variable, $Y(s_i)$, in a *sample vector*, $Y = [Y(s_i) : i = 1,..,n]'$.[11] This sample vector, Y, is now hypothesized to satisfy the generalized linear regression model in (7.3.2) with *attribute data*, X, and *covariance matrix*, $V(\theta)$.

7.3.1 Iterative Estimation of β and θ

We first give an overview of the estimation procedure and then formalize its individual steps. Every iterative estimation procedure must start with some *initial value*. Here, as with Universal Kriging, the initialization used (step [1] below) is to estimate β by OLS, which we designate as $\hat{\beta}_0$. The residuals $\hat{\varepsilon}_0$ generated by $\hat{\beta}_0$ are then used to obtain an

[10] It should be noted that in other treatments, such as Schabenberger and Gotway (2005), all such implementations are regarded simply as different ways of estimating the same "Universal Kriging model". However, for our purposes it seems best to avoid confusion by reserving the term "Universal Kriging" for the implementation adopted in ARCMAP, as outlined in Section 7.2.3 above.

[11] Note again that we here use Y for the *full sample* rather than Y_n. The latter is only required when we need to distinguish between the full sample and subsamples used for prediction at each location.

estimate, $\hat{\theta}_0$, of the spherical variogram parameters in (7.3.1). These are in turn used (in steps [2] to [6] below) to obtain a GLS estimate, $\hat{\beta}_1$ of β using the covariance matrix, $V(\hat{\theta}_0)$. Up to this point, the implementation is identical with that in Section 7.2.3. But the purpose of the present numbering of these estimators is to formalize a continuation of this procedure. Here the residuals, $\hat{\varepsilon}_1$, generated by $\hat{\beta}_1$ are next used (in step [7]) to obtain a new estimate, $\hat{\theta}_1$, of the spherical variogram parameter. If the estimates $(\hat{\beta}_1,\hat{\theta}_1)$ are deemed (as in steps [8] to [9] below) to be "sufficiently similar" to $(\hat{\beta}_0,\hat{\theta}_0)$, then the estimation procedure terminates with these as final values. Otherwise it continues until such values are found. With this overview, we now formalize these steps as follows:

[1] First construct an *OLS estimate*,

$$\hat{\beta}_0 = (X'X)^{-1}X'y \tag{7.3.3}$$

of β with corresponding *residuals*,

$$\hat{\varepsilon}_0 = y - X\hat{\beta}_0 . \tag{7.3.4}$$

[2] Use these residuals to estimate an *empirical variogram*, $\hat{\gamma}_0(h)$, at some set of selected distance values, $(h_i : i = 1,..,q)$.

[3] Next use this empirical variogram data $(\hat{\gamma}_{0i},h_i), i = 1,..,q$ to fit (by nonlinear least squares) a *spherical variogram*, $\gamma(h\,;\,\hat{\theta}_0)$, with *parameter vector*,

$$\hat{\theta}_0 \;=\; (\hat{r}_0,\hat{s}_0,\hat{a}_0) \;\; . \tag{7.3.4}$$

[4] Then use the identity, $C(h) = \sigma^2 - \gamma(h)$, to construct the corresponding *spherical covariogram*,

$$\hat{C}_0(h) = \hat{s}_0 - \gamma(h\,;\,\hat{r}_0,\hat{s}_0,\hat{a}_0) \tag{7.3.5}$$

for all distances h.

[5] If the distance between each pair of data points, s_i and s_j is denoted by h_{ij}, then the *covariance*, $\sigma_{ij} = \text{cov}(\varepsilon_i,\varepsilon_j)$, between the residuals at s_i and s_j is estimated by

$\hat{\sigma}_{0ij} = \hat{C}_0(h_{ij})$ [where by definition, $\sigma_{ii} \equiv \sigma^2 \Rightarrow \hat{\sigma}_{0ii} \equiv \hat{\sigma}_0^2 \equiv \hat{s}_0$], and the resulting estimate of the *covariance matrix*, $V(\theta) = \text{cov}(\varepsilon)$, between residuals at all data points $i = 1,..,n$ is given by[12]

(7.3.6) $$\hat{V}_0 = V(\hat{\theta}_0) = \begin{pmatrix} \hat{\sigma}_0^2 & \cdots & \hat{\sigma}_{01n} \\ \vdots & \ddots & \vdots \\ \hat{\sigma}_{0n1} & \cdots & \hat{\sigma}_0^2 \end{pmatrix}$$

[6] Using this covariance matrix, now apply *GLS* to obtain a *new* estimate of β:

(7.3.7) $$\hat{\beta}_1 = (X'\hat{V}_0^{-1}X)^{-1}X'\hat{V}_0^{-1}y \quad .$$

with corresponding *residuals*,

(7.3.8) $$\hat{\varepsilon}_1 = y - X\hat{\beta}_1$$

[7] Then replace $\hat{\varepsilon}_0$ by $\hat{\varepsilon}_1$ and apply steps [2] and [3] to obtain a *new* spherical variogram, $\gamma(h;\hat{\theta}_1)$, with parameter vector,

(7.3.9) $$\hat{\theta}_1 = (\hat{r}_1, \hat{s}_1, \hat{a}_1)$$

[8] At this point, one can check to see if there are any "significant" differences between the initial parameter estimates, $(\hat{\beta}_0, \hat{\theta}_0)$, and the new estimates, $(\hat{\beta}_1, \hat{\theta}_1)$. Here there are many criteria to check for differences. If one is primarily interested in the β parameters (as is typical in regression), the simplest approach is to focus on *fractional changes* in these estimates by letting[13]

(7.3.10) $$\Delta_1 = \max\left\{ \left| \frac{\hat{\beta}_{1j} - \hat{\beta}_{0j}}{\hat{\beta}_{0j}} \right| : j = 0,1,..,k \right\}$$

One may then choose an appropriate *threshold value*, $\overline{\Delta}$ (say $\overline{\Delta} = .001$) and define a *significant change* to be $\Delta_1 > \overline{\Delta}$. If one is also interested in the variogram parameters, $\theta = (r,s,a)$, then one may replace (7.3.10) by the broader set of *fractional changes*

[12] Be careful not to confuse this *initial estimate*, $\hat{V}_0$, with the estimated sub-matrix of covariances, $\hat{V}_0$, used to predict $Y(s_0)$ in previous sections.

[13] For a possible modification of this simple criterion, see Schabenberger and Gotway (2005, p.259).

$$\tilde{\Delta}_1 = \max\left\{\Delta_1, \left|\frac{\hat{r}_1-\hat{r}_0}{\hat{r}_0}\right|, \left|\frac{\hat{s}_1-\hat{s}_0}{\hat{s}_0}\right|, \left|\frac{\hat{a}_1-\hat{a}_0}{\hat{a}_0}\right|\right\} \tag{7.3.11}$$

[9] If there is *no significant change*, i.e., if $\Delta_1 \le \overline{\Delta}$ (or $\tilde{\Delta}_1 \le \overline{\Delta}$), then *stop* the iterative estimation procedure and set the *final parameter estimates* to be

$$(\hat{\beta},\hat{\theta}) = (\hat{\beta}_1,\hat{\theta}_1) \ . \tag{7.3.12}$$

[10] On the other hand, if $\Delta_1 > \overline{\Delta}$ (or $\tilde{\Delta}_1 > \overline{\Delta}$), then *continue* the iterative estimation procedure by replacing $\hat{\theta}_0$ with $\hat{\theta}_1$ in steps [4] through [7] to obtain a *new* β estimate,

$$\hat{\beta}_2 = (X'\hat{V}_1^{-1}X)^{-1}X'\hat{V}_1^{-1}y \tag{7.2.13}$$

[based on the new covariance matrix, $\hat{V}_1 = V(\hat{\theta}_1)$], and *new* variogram parameter estimates

$$\hat{\theta}_2 = (\hat{r}_2,\hat{s}_2,\hat{a}_2) \tag{7.2.14}$$

[based on the new residuals, $\hat{\varepsilon}_2 = y - X\hat{\beta}_2$].

[11] With these new parameters, define Δ_2 (or $\tilde{\Delta}_2$) as in step [8]. If $\Delta_2 \le \overline{\Delta}$ (or $\tilde{\Delta}_2 \le \overline{\Delta}$) then *stop* the procedure and set the final parameter estimates to

$$(\hat{\beta},\hat{\theta}) = (\hat{\beta}_2,\hat{\theta}_2) \ . \tag{7.2.15}$$

[12] On the other hand, if $\Delta_2 > \overline{\Delta}$ (or $\tilde{\Delta}_2 > \overline{\Delta}$), then *continue* the iterative estimation procedure by replacing $(\hat{\beta}_1,\hat{\theta}_1)$ with $(\hat{\beta}_2,\hat{\theta}_2)$ in steps [4] through [7].

[13] Continue in the same way until a set of parameters $(\hat{\beta}_m,\hat{\theta}_m)$ is found for which $\Delta_m \le \overline{\Delta}$ (or $\tilde{\Delta}_m \le \overline{\Delta}$). Then *stop* the procedure and set the *final estimates* to

$$(\hat{\beta},\hat{\theta}) = (\hat{\beta}_m,\hat{\theta}_m) \ . \tag{7.3.16}$$

These final parameter estimates are said to be *mutually consistent* in the sense that the covariance matrix, $\hat{V} = V(\hat{\theta})$, will (approximately) reproduce $\hat{\beta}$ as,

$$\hat{\beta} = (X'\hat{V}^{-1}X)^{-1}X'\hat{V}^{-1}y \tag{7.3.17}$$

and similarly, that the residuals, $\hat{\varepsilon} = y - X\hat{\beta}$, yield an empirical variogram, $\hat{\gamma}(h)$, that will (approximately) reproduce the parameter estimates, $\hat{\theta} = (\hat{r}, \hat{s}, \hat{a})$, of the spherical variogram yielding $\hat{C}$.

Here it should be emphasized that while this mutual consistency property is certainly desirable from a conceptual viewpoint, there is no guarantee that any of the Best Linear Unbiased estimation properties for GLS estimators will continue to hold for $\hat{\beta}$. Hence, as discussed at the end of the implementation for Simple Kriging in Section 6.2.5 above, these are often designated as *Empirical GLS estimators*.[14]

7.3.2. Implementation of Geostatistical Regression (Geo-Regression)

Given the regression estimates, $\hat{\beta}$, one can use the parameter estimates, $\hat{\theta} = (\hat{r}, \hat{s}, \hat{a})$, to construct the final *covariogram* as follows:

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

This covariogram is in turn used to obtain a final estimate,

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

of the *residual covariance matrix*, $V = V(\theta) = \text{cov}(\varepsilon)$ [mentioned above (7.3.17)].

To employ these estimates for inference about the components of β in geo-regression applications, one must estimate the *covariance matrix* of the estimator, $\hat{\beta}$, say $\Sigma = \text{cov}(\hat{\beta})$. Following standard GLS procedures, one can determine Σ as follows. By definition,

(7.3.20) $$\hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}Y = (X'V^{-1}X)^{-1}X'V^{-1}(X\beta + \varepsilon)$$

$$= (X'V^{-1}X)^{-1}(X'V^{-1}X)\beta + (X'V^{-1}X)^{-1}X'V^{-1}\varepsilon$$

$$= \;\beta + (X'V^{-1}X)^{-1}X'V^{-1}\varepsilon$$

[14] See for example the discussion in Waller and Gotway (2004, p.337).

But by the Linear Invariance Theorem for multi-normal random vectors [in (3.2.22)], it then follows that $\hat{\beta}$ is *multi-normally distributed* with mean

$$E(\hat{\beta}) = \beta + (X'V^{-1}X)^{-1}X'V^{-1}E(\varepsilon) = \beta + (0) = \beta \tag{7.3.21}$$

and covariance,

$$\begin{aligned} \Sigma &= \mathrm{cov}(\hat{\beta}) = \mathrm{cov}\left[(X'V^{-1}X)^{-1}X'V^{-1}\varepsilon\right] \\ &= (X'V^{-1}X)^{-1}X'V^{-1}\mathrm{cov}(\varepsilon)V^{-1}X\,(X'V^{-1}X)^{-1} \\ &= (X'V^{-1}X)^{-1}X'V^{-1}V\,V^{-1}X\,(X'V^{-1}X)^{-1} \\ &= (X'V^{-1}X)^{-1}(X'V^{-1}X)(X'V^{-1}X)^{-1} \\ &= \boxed{(X'V^{-1}X)^{-1}} \end{aligned} \tag{7.3.22}$$

Hence (7.3.19) yields the following estimate of Σ,

$$\hat{\Sigma} = (X'\hat{V}^{-1}X)^{-1} = \begin{pmatrix} \hat{v}_{11} & \cdots & \hat{v}_{1k} \\ \vdots & \ddots & \vdots \\ \hat{v}_{k1} & \cdots & \hat{v}_{kk} \end{pmatrix} \tag{7.3.24}$$

which in turn yields *standard error estimates*

$$s_j = \sqrt{\hat{v}_{jj}} \tag{7.3.25}$$

for each beta parameter estimate, $\hat{\beta}_j$, $j = 0,1,..,k$. These standard errors can then be used to construct *p-values* for significance tests of these coefficients based on the *t-ratios*:

$$\boxed{t_j = \hat{\beta}_j / s_j \ \ , \ j = 0,1,..,k} \tag{7.3.26}$$

Hence, standard tests of significance can be carried out in terms of these estimates.[15] This procedure is implemented in the MATLAB program, **geo_regr.m**, and will be illustrated in Section 7.3.4 below.

7.3.3. Implementation of Geostatistical Kriging (Geo-Kriging)

Recall that Universal Kriging used a prior estimate of the variogram parameters based on OLS residuals. But one can now improve this procedure by using the mutually consistent estimates obtained above. In doing so, we must again distinguish between the *full sample*

[15] As with OLS, t_j is *t*-distributed with $n-(k+1)$ degrees of freedom under the null hypothesis, $\beta_j = 0$. See also expressions (9.16) through (9.18) in Waller and Gotway (2004).

vector, Y_n, and the prediction sample vector, Y, used for predicting $Y_0 = Y(s_0)$ at a selected site, $s_0 \in R$. So for convenience we now rewrite model (7.3.2) as:

$$(7.3.27) \qquad Y_n = X\beta + \varepsilon \quad , \quad \varepsilon \sim N(0,V) = N[0,V(\theta)]$$

to emphasize that this model refers to the *full sample*. Hence for the mutually consistent estimates, $(\hat{\beta},\hat{\theta}) = [\hat{\beta},(\hat{r},\hat{s},\hat{a})]$, obtained from the iterative procedure above, the estimate, $\hat{\beta}$, now yields the (full sample) GLS estimate, $\hat{\beta}_n$, in (7.2.28), and the estimated covariance matrix, $V(\hat{\theta})$, yields the appropriate $\hat{V}$ matrix. So by mutual consistency, we may write[16]

$$(7.3.28) \qquad \hat{\beta}_n = (X'\hat{V}^{-1}X)^{-1}X'\hat{V}^{-1}Y_n$$

At this point, Steps 4 through 8 in the implementation of Universal Kriging can now be carried out in tact [where the prediction covariance estimates, $\hat{C}_0$, in (7.2.31) are again assumed to be constructed using the variogram parameters, $\hat{\theta} = (\hat{r},\hat{s},\hat{a})$, from the iterative estimation procedure].

In summary, while the iterative estimation procedure in *Geo-Kriging* is computationally more intensive than that of Universal Kriging, the mutual consistency of all estimated parameters should in principle yield more satisfactory spatial predictions. This procedure is implemented in the MATLAB program, **geo_krige.m**, and will be illustrated briefly at the end of Section 7.3.5 below.

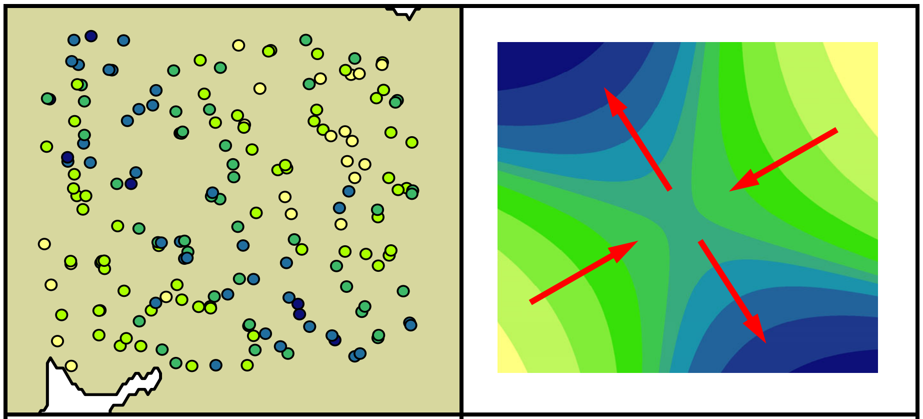
7.3.4 Cobalt Example of Geo-Regression

As an illustration of geo-regression, a small rectangular region of Vancouver Island has been selected in which Cobalt (*Co*) values appear to exhibit a interesting spatial trend, as shown in Figure 7.2(a) below. Notice in particular that the highest values tend to be in the northwest and southeast corners of this rectangle, while the lowest values tend to be in the southwest and northeast corners. This suggests a "saddle" shape, as depicted in Figure 7.2(b) below. Such saddle shapes, known technically as *hyperbolic paraboloids*, are instances of quadratic functions in the underlying coordinate variables, $s = (x, y)$. This suggests that spatial trends in this data might be well fitted by a *geo-regression* with a *quadratic spatial trend function* of the form,

$$(7.3.29) \qquad Co = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x y + \beta_4 x^2 + \beta_5 y^2 + \varepsilon$$

The Cobalt data for this example is in the JMP file, **Cobalt_1.JMP**. Before proceeding, it is worthwhile noticing from this data that the coordinates locations are in feet, so that

[16] Here the equality in (7.3.28) is implicitly taken to be "approximately equal" in the sense defined by the mutual consistency condition in the iterative estimation procedure above.

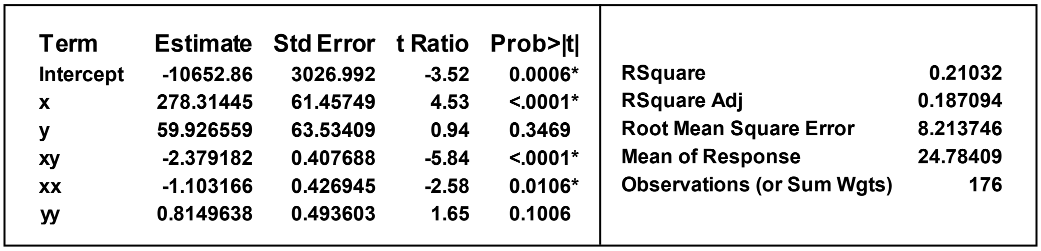


(a) Cobalt Data Map

(b) Spatial Trend

Figure 7.2. Cobalt Data Example

their values are quite large. For example, the first point is $(x_1 = 651612, y_1 = 566520)$. More importantly, when one forms a quadratic function, these values are squared in order of magnitude. So for example the cross product term in (7.3.29) is $x_1 y_1 = 3.69\times 10^{11}$. Since the cobalt magnitudes are drastically smaller (in this case, $Co_1 = 36$), it should be clear that some of the beta slope coefficients in (7.3.29) will be vanishingly small (roughly of order 10^{-8}). Such values are so close to zero that they are awkward to analyze. More importantly, since the intercept is by definition a data vector of ones, $1_n = (1,..,1)'$, this column in the data matrix, X, is vanishing small compared to other data columns like, xy. This can create *numerical instabilities* in the regression itself.[17] So before beginning the present analysis, it is advisable to rescale the coordinate data to a more reasonable range. In the present case, we have divided all coordinate values by 10,000, so that terms like the cross product above now have more tractable values ($x_1 y_1 = 3691.5$). With these values, the OLS regression in (7.3.29) yields the following results (where **xx** denotes x^2, and so on):



Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	-10652.86	3026.992	-3.52	0.0006*
x	278.31445	61.45749	4.53	<.0001*
y	59.926559	63.53409	0.94	0.3469
xy	-2.379182	0.407688	-5.84	<.0001*
xx	-1.103166	0.426945	-2.58	0.0106*
yy	0.8149638	0.493603	1.65	0.1006

RSquare	0.21032
RSquare Adj	0.187094
Root Mean Square Error	8.213746
Mean of Response	24.78409
Observations (or Sum Wgts)	176

Table 7.3. Initial OLS Regression

[17] Software such as JMP is usually sophisticated enough to employ internal rescaling procedures to avoid such obvious instabilities. But this is not true of all regression software.

Notice that y is not significant, and that y^2 is only weakly significant. But since there are clear nonlinearities in the y direction, this suggests that the collinearity between y and y^2 in this region are masking the effect of y^2. If the insignificant y variable is removed, then one obtains the new regression shown below.

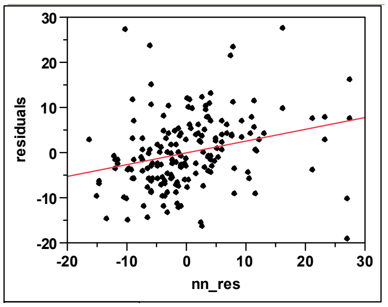
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	-8439.792	1911.869	-4.41	<.0001*
x	262.9905	59.25208	4.44	<.0001*
xy	-2.204434	0.363043	-6.07	<.0001*
xx	-1.062146	0.424588	-2.50	0.0133*
yy	1.2389119	0.203946	6.07	<.0001*

RSquare	0.206187
RSquare Adj	0.187618
Root Mean Square Error	8.211095
Mean of Response	24.78409
Observations (or Sum Wgts)	176

Table 7.4. Final OLS Regression

Notice that y^2 is now very significant, and moreover, that the adjusted R^2 value has *increased* by removing y. This is a clear indication that the present model is capturing this spatial trend more accurately. Note finally that the coefficients on x^2 and y^2 have opposite signs. This is a characteristic of hyperbolic paraboloids.[18]

However, there still remains the question of possible spatial dependencies among the unobserved residuals, ε, in (7.3.29). We can check this in the usual way by regressing these residuals on their nearest-neighbor residuals. The result of this regression are shown below:



Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	-0.128734	0.59103	-0.22	0.8278
nn_res	0.2628387	0.069852	3.76	0.0002*

Figure 7.3. OLS Residual Analysis

Here it is clear that there does indeed exist significant spatial dependency among these residuals. As discussed in Section 7.1.3, this can in turn inflate the significance levels

[18] See for example *http://mathworld.wolfram.com/HyperbolicParaboloid.html*.

obtained in Table 7.4. So this motivates an extended analysis using geo-regression to account for these dependencies.

To do so, this cobalt data has been transported to MATLAB, and is found in the workspace, **Cobalt_1.mat**. Here the 176 locations are stored in the matrix, **L0**, with corresponding cobalt values in **y0** and data [**x**, **xy**, **xx**, **yy**] in the matrix, **X0**. The geo-regression is run with the command,

```
>> OUT = geo_regr(y0,X0,L0,vnames);
```

where **vnames** contains the variable names, and is constructed by the command:

```
>> vnames = strvcat('X','XY','XX','YY');
```

The actual regression portion of the screen output for this iterative estimation procedure is as follows:

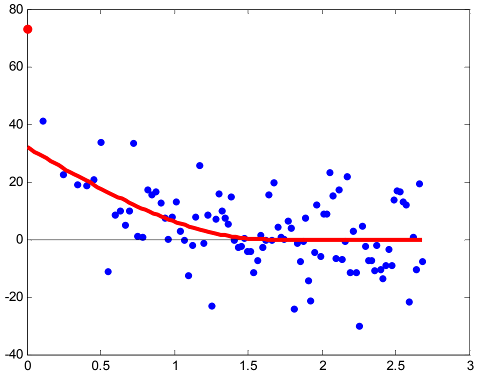
```
FINAL REGRESSION RESULTS:

VAR        COEFF        T-VAL       PROB
const  -6848.808565   -1.877835   0.062106
X        212.895520    1.881204   0.061644
XY        -1.813464   -2.509593   0.013017
XX        -0.843514   -1.028343   0.305241
YY         1.021249    2.519087   0.012683
```

Table 7.5. Regression Output of Geo_Regr

Notice first that the basic signs of all beta coefficient are the same, so that this new spatial trend is again a “saddle” shape. In fact this is precisely the saddle shape plotted in Figure 7.2(b) above. But the main thing to notice is that all variables are now *less significant* than they were under OLS. In particular, x^2 is no longer even weakly significant. However, the *relative ordering* among the *p*-values (as seen more clearly from the absolute *t*-values) is essentially the same. So there appears to have been a fairly uniform deflation of all significance levels under OLS. While this will certainly not always be true, in the present case it suggests that spatial dependencies in these OLS residuals are relatively isotropic (i.e., the same in the *x* and *y* directions), and hence are consistent with the *covariance stationarity* assumption underlying geo-regression.

Before interpreting these results, it is important to check to see whether this geo-regression has in fact removed the spatial dependencies among residuals. Here it is important to stress that this *cannot* be done by simply examining the residuals of the geo-regression. Indeed these residuals exhibit precisely the spatial covariance structure estimated by the geo-regression as displayed in Figure 7.4 below:



SPHERICAL VARIOGRAM-
COVARIOGRAM PARAMETERS:

Range 16265.939790
Sill 73.253163
Nugget 41.206102

Figure 7.4. Covariogram Estimate

So the task remaining is to *remove* this estimated spatial covariance structure and determine whether any spatial dependencies remain. This can be accomplished by recalling that every GLS model can be reduced to an equivalent OLS model by the Cholesky procedure in (7.1.15) through (7.1.20) above. By way of review, let us now write the appropriate GLS model as

$$(7.3.30) \qquad Y = X\beta + \varepsilon \ , \ \ \varepsilon \sim N(0,V)$$

where in this case, Y is the random vector of $n = 176$ cobalt levels, X is the $(n \times 4)$ matrix of coordinate variables (labeled as **X0** above), and ε is the spatially dependent residual vector with unknown covariance matrix, V. As in (7.1.15), if T denotes the Cholesky matrix for V, so that $V = TT'$, then as in (7.1.16) and (7.1.17), if we multiply both sides of (7.3.30) by T^{-1}, and let $Y_T = T^{-1}Y$, $X_T = T^{-1}X$, and $\varepsilon_T = T^{-1}\varepsilon$, then we obtain a new linear model,

$$(7.3.31) \qquad Y_T \ = \ X_T\beta + \varepsilon_T \ , \ \varepsilon_T \sim N(0, V_T)$$

where β is exactly the same as in (7.3.30), but where the argument in (7.1.19) now shows that the covariance matrix, V_T, is simply the identity matrix, i.e.,

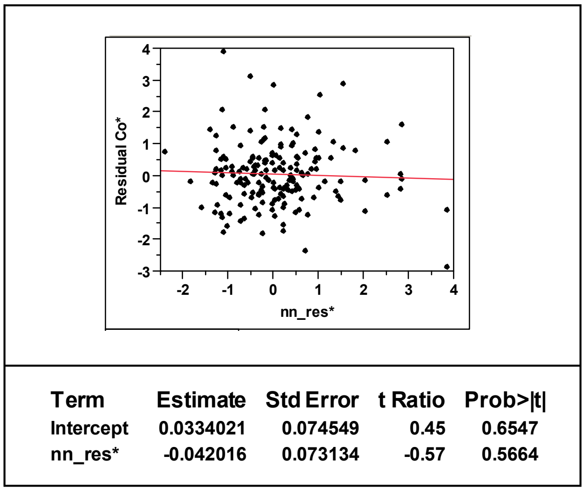
$$(7.3.32) \qquad V_T = T^{-1}V(T^{-1})' = T^{-1}(TT')(T')^{-1} = I_n$$

In particular, this implies that the components of the transformed residual vector, ε_T, are *independent*. Of course, the true covariance matrix, V, and its Cholesky matrix, T, are unknown. But if the geo-regression above was successful, then the covariogram estimate in Figure 7.4 should generate a reasonably good estimate, $\hat{V}$, of this covariance matrix [by the same procedure as in (7.2.25) through (7.2.27) above]. If so, then by letting $\hat{T}$

denote the Cholesky matrix for $\hat{V}$, we can use this to transform the given data into an OLS regression problem. In particular, if $[y, X]$ denotes the given cobalt and coordinate data (represented by [**y0**,**X0**] above), then the transformed data for the present case is given by,

$$(7.3.33) \qquad \hat{y}_T = \hat{T}^{-1} y \quad , \quad \hat{X}_T = \hat{T}^{-1} X$$

Hence if the geo-regression above was successful, then this data should yield an OLS regression with approximately *independent* residuals. This can be checked by the nearest-neighbor regression procedure above, and provides a useful diagnostic for geo-regression. To do so, the transformed data in (7.3.33) is saved as part of the output of geo-regression. By examining the program description of **geo_regr.m**, it can be seen that the fifth component, **OUT{5}**, of the output cell structure, **OUT**, contains precisely the matrix $[\hat{y}_T, \hat{X}_T]$. This can be imported to JMP and run as a regression. In doing so, it is important to note that the first column of the data matrix, X , in (7.3.30) is necessarily the unit vector, 1_n , corresponding to the intercept coefficient, β_0 . But in $\hat{X}_T$ this is transformed to the vector, $\hat{T}^{-1} 1_n$, which is *not* a unit vector. So if this regression were run in JMP without modification, then JMP would *add* a unit vector which is not present in (7.3.30). This means that JMP must be run using the "No Intercept" option (at the bottom of the Fit Model window).[19] The results of this no-intercept regression must produce exactly the same beta estimates as the geo-regression output above (except for possible rounding errors in transporting the data from MATLAB). So this in itself is a good check to be sure that the data has been transported properly. The results of this nearest-neighbor residual regression are shown below:



Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.0334021	0.074549	0.45	0.6547
nn_res*	-0.042016	0.073134	-0.57	0.5664

Figure 7.5. Transformed Residual Analysis

[19] We shall see this option used again in Section 4.1.1 of Part III.

Here it should be clear that the geo-regression above has indeed been successful in removing any trace of spatial dependencies among residuals. However, there is one additional check that is worth mentioning. Notice in (7.3.32) that these transformed residuals are not only independent, but in fact all have unit variance ($\sigma^2 = 1$) so that the associated standard deviation is also one ($\sigma = 1$). This means that the estimated standard deviation, $\hat{\sigma}$, known as "Root Mean Squared Error" should be close to one. This value is reported in the regression output right under Adjusted R^2. In the present case, $\hat{\sigma} = 0.995$, which provides additional support for the success of this geo-regression.

By way of summary, this cobalt example provides a simple illustration of the use of geo-regression. Here the objective has been simply to capture the overall shape of spatial trends in this data. (A more substantive example will be given in the next section.) But aside from the geo-regression procedure itself, this example serves to illustrate a number of more general issues that are common to *all* spatial regressions. First notice from the initial OLS regression itself that this spatial trend captures less than 20% of the overall variation in this cobalt data (with an adjusted R^2 of 0.188). So even though a visual inspection of Figure 7.2(a) suggests an overall "saddle" shape for these trends, the present quadratic specification is at best only a rough approximation. Thus for purposes of spatial prediction, it is vital that the residual structure be modeled in a careful way. This is a further motivation for techniques like geo-regression.

From an even more general perspective, this example illustrates the fundamental problem of separating "trends" from "residuals". To what extent is the spatial pattern of cobalt values in Figure 7.4(a) the result of some underlying trend, or simply the result of correlations between cobalt values at nearby locations? If one were able to examine many "replications" of the underlying spatial process, then such separation would be a relatively simple matter. Indeed, if most replications produced similar "saddle-like" patterns, then this would suggest the presence of a dominant spatial trend along the lines that we have modeled. On the other hand, if such replications produced a wide variety of similarly correlated patterns (including "mountains" and "valleys" as well as "saddles'), then this would suggest the presence of a dominant covariance stationary process, possibly even with a constant mean (as postulated in Ordinary Kriging for example). But since direct replications are not possible, the best that one can do is to be aware of these problems, and to treat all model specifications with some degree of suspicion. To paraphrase the famous remark of George Box,[20] "all models are wrong, but some are more useful than others".

7.3.5 Venice Example of Geo-Regression and Geo-Kriging

The following example of geo-regression is more substantive in nature, and is based on the "Ground Water in Venice" data from [BG, pp.147-148]. This data set originally appeared in the two-part article by Gambolati and Volpi (1979) [which is included as

[20] See for example http://en.wikipedia.org/wiki/George_E._P._Box.

References 7 and 8 in the class reference material].[21] The area around Venice Island in Italy is shown (schematically) in Figure 7.6 below.

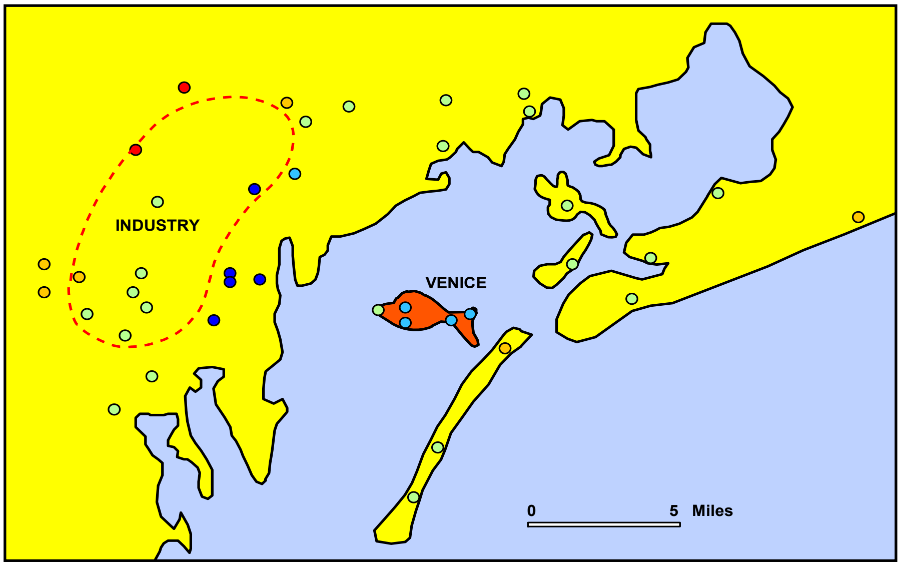


Figure 7.5. Venice Island and Lagoon

Venice Island (shown in red) lies in a shallow lagoon, and has been slowly sinking for many decades. In 1973 there was a suspicion that the Puerto Marghera industrial area to the west of Venice was contributing to this rate of sinking. The reason for this suspicion can be seen from the schematic depiction of the groundwater structure underlying the Venice Lagoon shown in Figure 7.6 below.

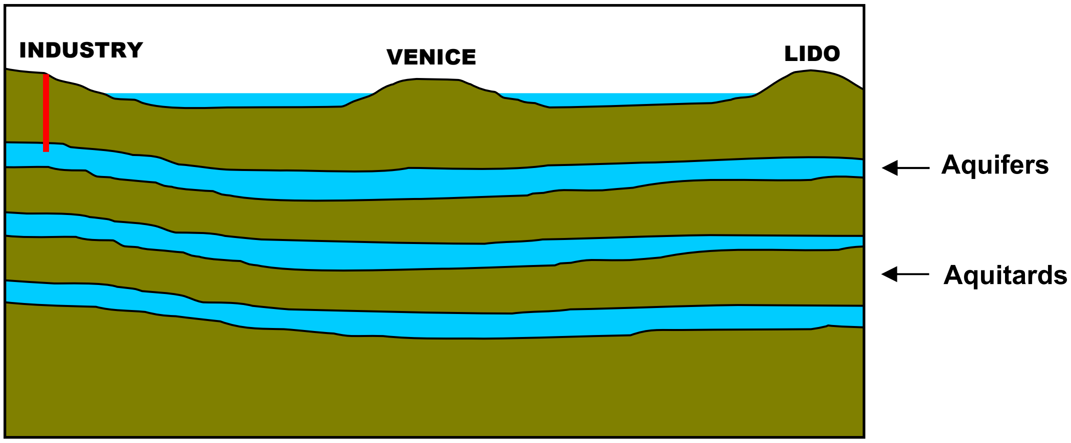


Figure 7.6. Venice Aquifer System

[21] This paper also contains an excellent overview of Kriging methods, as well as the groundwater problem in Venice.

Here the blue bands denote porous water-filled layers of soil called *aquifers* that are separated by denser layers called *aquitards*. Industry consumes water by drilling wells into the aquifer layers (as depicted by the red shaft in the figure). This lowered the level of the water table, potentially contributing to the sinking of Venice. Thus the question in 1973 was whether or not this industrial *draw-down* of water was a significant factor in the sinking of Venice.

Geo-Regression Model

To study this question, data was gathered on *water table levels*, L_i , from 40 bore hole sites, $i = 1,..,40$, in existing wells throughout the Venice Lagoon area (shown by the dots in Figure 7.5 above with colors ranging from red to blue denoting higher to lower levels). [This data, along with the coordinate locations of well sites, can be found in the (40×3) matrix, **venice**, in the workspace, **venice.mat**.] The objective of this study was to identify the key factors influencing these water table levels by applying geo-regression methods. Here it was hypothesized that the key factors influencing the water table level, $L(s)$, at any location, $s = (s_1, s_2)$, were the elevation, $Ev(s)$, above sea level at *s*, together with local draw-down effects both from industry, $D_I(s)$, and from local water consumption, $D_V(s)$, in Venice itself. To model D_I a convenient coordinate system was chosen, with origin centered in the Industrial Area as shown in Figure 7.7 below.

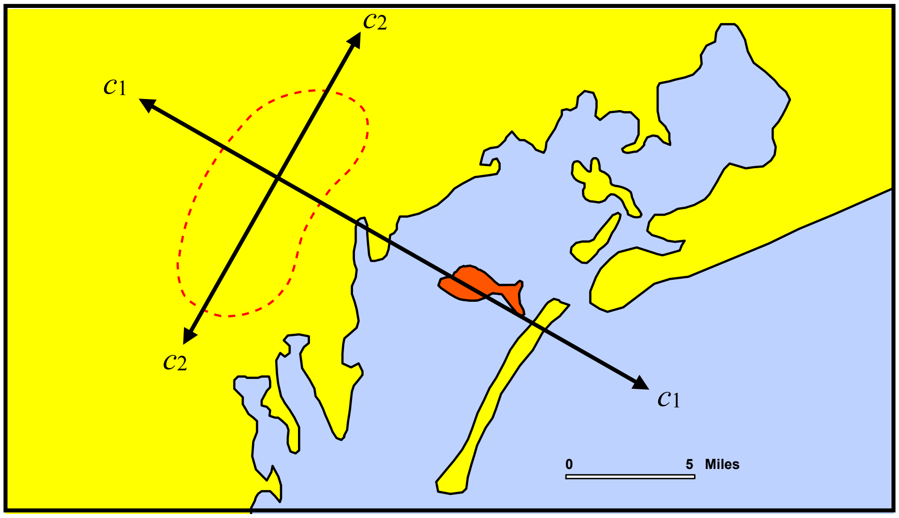


Figure 7.7. Spatial Coordinates for Analysis

For later use, we now record this coordinate transformation as follows:

$$c_1 = c_1(s) = (0.01)\,[0.873(s_1 - 418) - 0.488(s_2 - 458)]$$

(7.3.34)

$$c_2 = c_2(s) = (0.01)[0.488(s_1 - 418) + 0.873(s_2 - 458)]$$

The orientation of these axes is designed to simplify the model representation of both elevation and industrial draw-down effects. Starting with the *Industrial draw-down function*, D_I, this can be essentially approximated by a decreasing function with elliptical contours centered on the axes. The present equation used is the following:[22]

$$(7.3.35)\qquad D_I(s) = D_I[c_1(s), c_2(s)] = \exp\left\{-[(1.5)c_1^2 + c_2^2]\right\}$$

A similar draw-down function, D_V, was constructed for Venice Island and has the following form:

$$(7.3.36)\qquad D_V(s) = D_V(s_1, s_2) = \exp\left\{-\left(\sqrt{(s_1 - 560)^2 + (s_2 - 390)^2}\Big/35\right)^8\right\}$$

Here the large exponent, $(\cdot)^8$, is designed to drive this function to zero outside of Venice Island, where local water consumption has little effect. The procedure for calculating these functions (as well as the elevation function below) can be found in the MATLAB script, **venice_funcs.m**. The resulting contours of these two functions are shown in Figures 7.8 and 7.9 below.

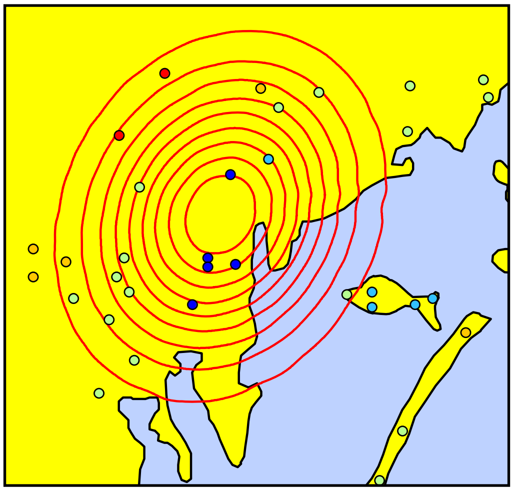


Figure 7.8. Industry Draw-Down

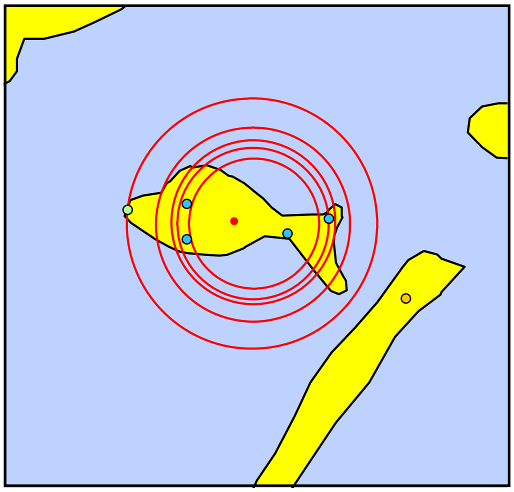


Figure 7.9. Venice Draw-Down

As mentioned above, there is a third effect that cannot be overlooked, namely *elevation*. Though detailed data on elevation was not available in this data set, the elevation contours are roughly parallel to the c_2 axis in Figure 7.7, and increase in elevation more

[22] The actual functions used in Gambolati and Volpi (1979) are based on more complex hydrological models. So the present simplified functions are for illustrative purposes only.

rapidly to the west. So the following simple (local) approximation to *elevation*, $Ev(s)$, at locations, s, was adopted,[23]

$$(7.3.37) \qquad Ev(s) = Ev[c_1(s)] = 10\exp(-c_1)$$

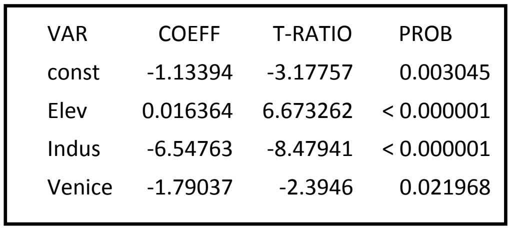
If the data sites (well locations) are denoted by $s_i = (s_{i1}, s_{i2})$, $i = 1,..,40$, and if the computed values of the above functions at these locations are denoted by $(D_{Ii}, D_{Vi}, Ev_i) = [D_I(s_i), D_V(s_i), Ev(s_i)]$, $i = 1,..,40$, then these values can now serve as the explanatory variables in a *linear regression model* of this water table data as follows:

$$(7.3.38) \qquad L_i = \beta_0 + \beta_I D_{Ii} + \beta_V D_{Vi} + \beta_{Ev} Ev_i + \varepsilon_i \ , \ i = 1,..,40.$$

As with the Cobalt example above, this model was run using both OLS and the iterative Geostatistical Regression Procedure implemented in **geo_regr.m**, with the command

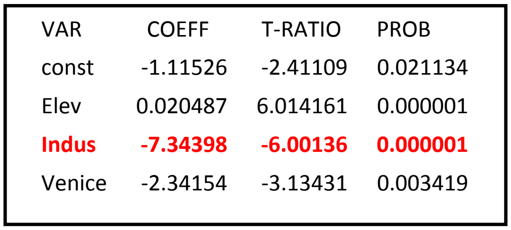
>> geo_regr(y0,X0,L0,vnames);

where **y0** is the L data, **X0** the computed (D_I, D_V, Ev) data, and **L0** the coordinate data at each of the 40 well sites. A comparison of the parameter estimates and significance levels in shown in Tables 7.6 and 7.7 below;



VAR	COEFF	T-RATIO	PROB
const	-1.13394	-3.17757	0.003045
Elev	0.016364	6.673262	< 0.000001
Indus	-6.54763	-8.47941	< 0.000001
Venice	-1.79037	-2.3946	0.021968

Table 7.6. OLS Estimates



VAR	COEFF	T-RATIO	PROB
const	-1.11526	-2.41109	0.021134
Elev	0.020487	6.014161	0.000001
Indus	**-7.34398**	**-6.00136**	**0.000001**
Venice	-2.34154	-3.13431	0.003419

Table 7.7. Geo Regression Estimates

Note that as in the Cobalt case above, the signs of all coefficients are consistent in both procedures, but the *t*-ratios are generally lower (in absolute magnitude) for GLS. Notice however that the Venice drawdown effect provides an exception to this rule, and shows that significance levels *need not always be higher for OLS*. As a final consistency check, note that the signs of these coefficients are as expected, namely that mean water table levels rise with higher elevations and that greater levels of water drawdown lower the mean water table level.

Before analyzing the consequences of these results, it is important to determine whether spatial correlation effects have been removed by this geo-regression procedure. Rather

[23] This approximation produces a maximum elevation of about 30 meters at the western edge of the Industrial Area, where the water table level is about 7 meters.

than repeat the nearest-neighbor residual analysis done for the Cobalt case, it is of interest to consider a different approach here. In particular, one can compare the (spherical) covariogram for the original OLS residuals with that of the residuals from the final transformed model in expressions (7.3.31) and (7.3.32) above. If the procedure has been successful, then the final covariogram should be much closer to pure independence. But it is important to note here that since the transformed data is quite different from that of the original model, there is a problem in comparing these residual covariograms directly. In fact, this provides us with an important case where it is more appropriate to compare the *correlograms* derived from these covariograms, as defined in expression (3.3.13) above. These correlograms are free from any dimensional restrictions, and hence are directly comparable. In particular, since $\rho(0)=1$ for *all* correlograms, their scales must be identical. This allows one to focus entirely on their relative *shapes*. In the present case, the original correlograms and final correlograms of the transformed data are shown in Figures 7.10 and 7.11, respectively. Notice first that in the original correlogram the relative nugget effect (defined in Section 4.5 above) is *zero*, indicating that this process exhibits *no* spatial independence whatsoever. In contrast, the relative nugget effect in the final correlogram is *close to one*, indicating that the process is now almost completely spatially independent. In other words, very little spatial correlation remains in this transformed data. Notice also that the fluctuation of nonzero correlation values is much smaller, indicating that spatial correlations are uniformly closer to zero at all scales.[24] These two observations provide convincing evidence that this geo-regression has indeed been successful in accounting for almost all spatial correlation in the original OLS model.

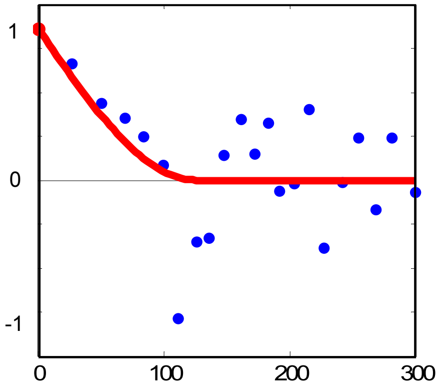


Figure 7.10. Original Correlogram

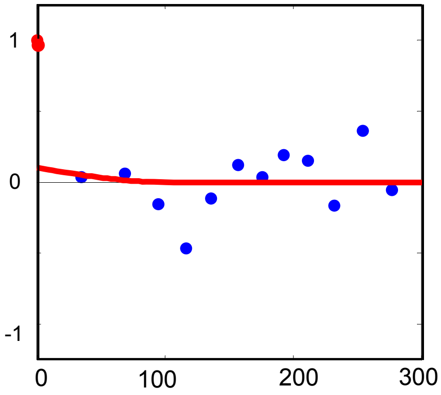


Figure 7.11. Final Correlogram

Impact Analysis of Industrial Water Drawdown

Given these preliminary findings, the main purpose of this model is to analyze the impacts of industrial water drawdown effects on the water table level in Venice. To estimate this impact, observe first from the geo-regression results above, that we can

[24] This is due in part to the larger bin sizes used in this figure (50 rather than 30 points per bin).

obtain a *upper 95% confidence bound* on the beta coefficient, β_I, for D_I in model (7.3.38) as follows. First note that if the *standard error* of $\widehat{\beta}_I$ is denoted by s_I, then for any level of significance, α, the $100(1-\alpha)\%$ upper confidence bound for β_I can be obtained from the probability identity,

$$(7.3.39) \qquad \Pr(\beta_I \le \widehat{\beta}_I + t_{\alpha,n-(k+1)} s_I) = 1-\alpha$$

where $t_{\alpha,n-(k+1)}$ is the *t*-critical value at level α for degrees of freedom, $n-(k+1)$ [where $n =$ sample size and $k =$ number of explanatory variables]. To obtain the desired standard error, recall that by definition the *t*-ratio, t_I, for β_I in Table 7.7 is given by $t_I = \widehat{\beta}_I / s_I$, so that by Table 7.7,

$$(7.3.40) \qquad s_I = \widehat{\beta}_I / t_I = (-7.34398)/(-6.00136) = 1.2237$$

Hence noting that in our case, $n = 40$ and $k = 3$ [so that $n-(k+1) = 40-4 = 36$], the desired upper 95% confidence bound is given by

$$(7.3.41) \qquad \beta_I \le \widehat{\beta}_I + t_{.05,36}\, s_I = -7.34398 + (1.6883)(1.2237) = -5.278$$

Next observe that for the representative location, $s = (s_1, s_2) = (555{,}390)$, in the middle of Venice Island (shown by the red dot in Figure 7.9 above), the transformed coordinates in (7.3.34) are seen to be $(c_1, c_2) = (\ 1.572, 0.075)$, so that the value of the Industrial drawdown in (7.3.35) is given by:

$$(7.3.42) \qquad D_I(s) = \exp\left\{-[(1.5)c_1^2 + c_2^2]\right\} = 0.02442$$

Thus, for each additional meter of Industrial water drawdown, one can be 95% confident that the *expected decrease*, Δ, in the water table level at location *s* will be bounded below by

$$(7.3.43) \qquad \boxed{\Delta \ge D_I(s)|-5.278| = (0.02442)(5.278) = 0.1289\ \textit{meters}}$$

Thus, based on the above model, one can be 95% confident that *the mean industrial drawdown effect on Venice Island is at least* 12%.

While this model is only a rough approximation to the analysis of Gambolati and Volpi (1979),[25] it serves to illustrate how geo-regression can actually be used to address substantive spatial issues. According to these authors, water pumping in Puerto Marghera

[25] Aside from their more elaborate drawdown functions, Gambolati and Volpi also used a universal kriging approach rather than our present application of geo-regression.

was in fact reduced by 60% after 1973, and their subsequent analysis of 1977 data showed that the "subsurface flow field had substantially recovered, and the land settlement had been arrested". So their post-analysis confirmed that this industrial water drawdown was indeed a major contributing factor to the sinking of Venice. Of course, in more recent times, Venice has once again started to sink from more natural causes. But this is another story.

An Application of Geo-Kriging

Finally it is of interest to apply geo-kriging to the Venice data as an illustration of this technique. To do so, a grid was constructed using **grid_form.m** in MATLAB with specified values

$$\begin{aligned} s_1 &= [150:25:900] \\ s_2 &= [200:25:650] \end{aligned} \tag{7.3.44}$$

(where the cell size, 25, is roughly a third of a mile in terms of Figure 7.5). This grid was then used as input to the program, **geo_krige.m**, with the command

```
>>  OUT = geo_krige(y0,X0,L0,X1,L1,h);
```

where (**y0,X0,L0**) is the same as for **geo_regr** above, and where (**X1,L1**) are the computed values of (D_I, D_V, Ev) and coordinate values at each of the 589 grid points from (7.3.44). Finally, the *bandwidth* used was **h = 50** (around two thirds of a mile.)

To visualize these results, it is convenient to compare the geo-kriging output values, $\hat{Y}(s)$, with the geo-regression estimates, $\hat{L}(s)$, of *expected water table levels* based on the results in Table 7.7, where by definition,

$$\hat{L}(s) = \hat{\beta}_0 + \hat{\beta}_I D_I(s) + \hat{\beta}_V D_V(s) + \hat{\beta}_{Ev} Ev(s) \tag{7.3.45}$$

for all locations, s. These results were constructed from the above output as follows:

```
>> b = OUT{1}(:,1);           • Extract beta estimates from output

>> X = [ones(589,1),X1];      • Construct regression matrix (with intercept)

>> L_hat = X*b;               • Compute estimates (L̂) of expected L values

>> Y_hat = OUT{3};            • Extract kriged values from output

>> StdErr = OUT{4};           • Extract standard error values from output
```

This data was then collected into a single data matrix:

>> DAT = [L1, L_hat, Y_hat, StdErr];

and exported from MATLAB to ARCMAP. These values were then interpolated using the Spline option in **ArcToolbox**:

Spatial Analyst Tools > Interpolation > Spline

and finally converted to contour form by applying

Spatial Analyst Tools > Surface > Contour

to the spline rasters. A comparison of the fitted value, $\hat{L}$, and kriging values, $\hat{Y}$, is shown in Figures 7.12 and 7.13 below:

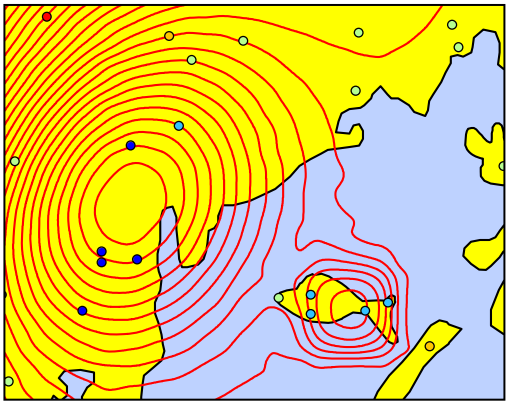


Figure 7.12. Geo-Regression $\hat{L}$ Values

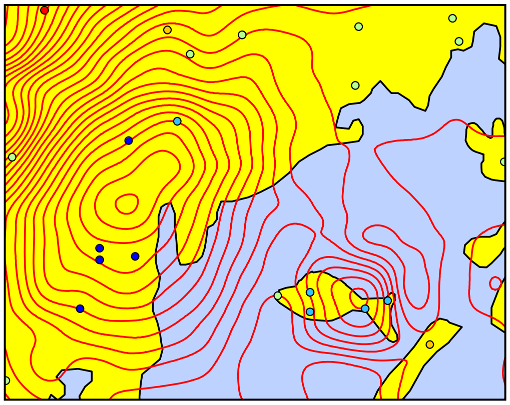


Figure 7.13. Geo-Kriging $\hat{Y}$ Values

Notice that the $\hat{L}$ values are essentially a weighted combination of the drawdown effects, D_I and D_V, in Figures 7.8 and 7.9 respectively (as captured by their values at the 40 well-site data points). The kriged values, $\hat{Y}$, also reflect these underlying drawdown effects, but to a lesser extent. By construction, these values also include stochastic interpolations of the regression residuals, and thus should reflect water table levels more accurately than the simpler regression predictions. Note however that alternative models of drawdown functions and fitting procedures will of course produce somewhat different results, as can be seen by comparing Figure 7.13 with Figure 5.21(a) in [BG, p.199] and Figure 2(a) in Part 2 of Gambolati and Volpi (1979, p.292).

Finally, the main advantage of this stochastic interpolation procedure is that it allows prediction intervals to be constructed for actual water table levels in terms of estimated standard errors of prediction. A plot of these standard errors around Venice Island is shown in Figure 7.14 below (with the 0.4 and 0.7 contours labeled to indicate representative values). Here a much finer grid of kriging locations was used here (with increments of about a tenth of a mile) in order to show the details of these standard error contours.

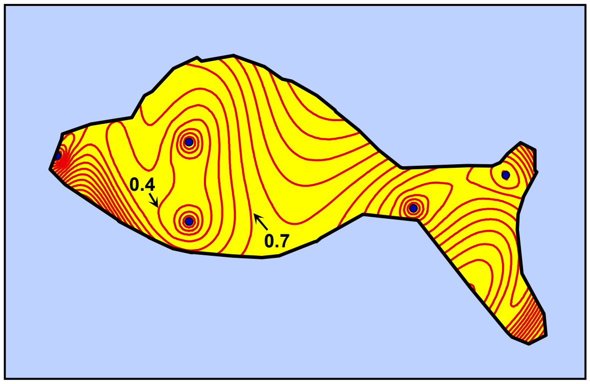


Figure 7.14. Kriging Standard Errors

Notice in particular that these standard errors *fall to zero* at each of the five data points (well sites) on Venice Island [in a manner similar to Figure 2(b) in Part 2 of Gambolati and Vopi (1979), though Venice Island itself is rather difficult to see in their figure]. This reflects the fact that geo-kriging (along with simple and ordinary kriging) is an *exact interpolator* that goes through every data point. This can be seen most easily from expression (7.2.17) above, together with the fact that if point s_0 is actually a data point, then it must always be a member if its own predition set, $S(s_0)$, and hence must correspond to one of the elements of the covariance matrix, V_0. But since $V_0 V_0^{-1} = I_{n_0} = (e_i : i = 1,..,n_0)$, it follows that if c_0 is the i^{th} column of V_0, then $c_0' V_0^{-1} = e_i'$, so that (7.2.17) becomes:

(7.3.46)
$$\hat{Y}(s_0) = x_0'\hat{\beta}_{n_0} + c_0' V_0^{-1}(Y - X_0\hat{\beta}_{n_0})$$

$$= x_0'\hat{\beta}_{n_0} + e_i'(Y - X_0\hat{\beta}_{n_0})$$

$$= x_0'\hat{\beta}_{n_0} + [Y(s_0) - x_0'\hat{\beta}_{n_0}] = Y(s_0)$$

This same argument also shows that the kriging standard error in (7.2.22) is identically zero.

Finally, it is of interest to consider the kriged values on Venice Island. Though the specific kriging contour values are not shown in Figure 7.13, these values yield water table predictions of around $\hat{Y}(s_0) \approx -3.0$ for points, s_0, on Venice Island (i.e., about 3 meters below sea level). Moreover, while not all standard error contours are shown in Figure 7.14, the 0.7 contour is roughly the average value, so that $\hat{\sigma}_0 = \hat{\sigma}(s_0) \approx 0.7$. Thus a typical prediction interval for points s_0 on Venice is about

$$\hat{Y}(s_0) \pm (1.96)\hat{\sigma}_0 \approx -3 \pm 1.4 \ \textit{meters} \tag{7.3.47}$$

While such intervals are not extremely sharp, one must take into account the fact that only 5 of the 40 data points are actually on Venice Island. So this is probably about the best that can be expected from such a small data set.