COKRIGING, KERNELS, AND THE SVD: TOWARD BETTER GEOSTATISTICAL ANALYSIS

by

Andrew Edmund Long

A Dissertation Submitted to the Faculty of the

Graduate Interdisciplinary Program
in Applied Mathematics

In Partial Fulfillment of the Requirements
For the Degree of

Doctor of Philosophy

In the Graduate College

The University of Arizona

1 9 9 4

Acknowledgments

A long and twisted path has brought me to this point: my journey has been (for the most part) a pleasant one, due primarily to some special travelling companions. It is a pleasure to thank them.

My mother and father kept the wolves from the door throughout graduate school; but, more importantly, they have always tried, and often managed, to keep the wolves from my mind. You have been my best teachers, using the best technique: education by example. Thanks for the support.

My ears and I will miss the whole KXCI family: Milo and Victor, for "the Bluegrass Show"; Carol Anderson; Peter Bourque; Eb Eberlein; Betsy Meisner; Kidd Squidd; Michael Hyatt; Jim Foley; Annie Barva; Mike Landwehr and George "It's Five" Ferris, my pals. Tucson is so lucky. Fish Karma, Rainer, Los Lasers: way cool! Thanks for teaching me, and keeping my spirits up!

Lois, Bob, and Kathleen: you've been the bosses, and you made it happen. Keep up the good deeds and work. I appreciate all your help and support.

Certain professors went above and beyond the call of duty, either in terms of making me feel comfortable here, or by trying harder to communicate difficult ideas than anyone could reasonably expect, or by spending time outside of class sharing of themselves. I thank particularly Bruce Bayly, Jim Cushing, Rick Michod, Hermann Flaschka, and Committee member Dr. A. Warrick in this regard.

Thanks, Anna and Tchapo, for putting up with the "Grumpy Old Man", especially at the last, when I was juggling a job search, a dissertation, and a family. I try to look cool but you saw me sweat. I regret especially the times when I foolishly felt that I couldn't afford even to read to you, Tchapo. I was wrong: I couldn't afford not to, buddy, and I'll try to do better from now on.

Friends Dan, Jim, Howard, Peter, David, Aric, George, Elizabeth and Warren: all of you got me through the transition period from ex-Peace Corps Volunteer to student (again! Good Lord, will it never end?! I guess it does...). Thanks guys, and especially Dan, who made the initial effort to bring me into the fold, and has proven himself a steadfast friend under all circumstances; Jim, who has bared his soul and been a searching and challenging friend; and my steady little buddy Howard: all three of you have helped make Tchapo a better fella, and he, Anna, and I salute you. I've had a blast, sometimes; fun, most of the time; and you were there when I wasn't having any fun at all.

Dr. Myers: you've been patient, interested, and interesting. I enjoy working with you, and look forward to working with you in the future. Thanks for being the lure that brought me here, and the prod that pushed me through. Your hard work in the interests of your students does not go unnoticed, although it may sometimes seem to go unrewarded. You'll probably say that the success of your charges is all the reward

you need; I contribute my heartfelt thanks, nonetheless.

Dedication

For my father

Thanks, Dad, for showing me how much fun this would be!

Table of Contents

TABLE OF CONTENTS—*Continued*

List of Figures

List of Figures—*Continued*

List of Figures—*Continued*

List of Figures—*Continued*

List of Figures—*Continued*

List of Tables

Abstract

Three forms of multivariate analysis, one very classical and the other two relatively new and little-known, are showcased and enhanced: the first is the Singular Value Decomposition (SVD), which is at the heart of many statistical, and now geostatistical, techniques; the second is the method of Variogram Analysis, which is one way of investigating spatial correlation in one or several variables; and the third is the process of interpolation known as cokriging, a method for optimizing the estimation of multivariate data based on the information provided through variogram analysis.

The SVD is described in detail, and it is shown that the SVD can be generalized from its familiar matrix (two-dimensional) case to three, and possibly n, dimensions. This generalization we call the "Tensor SVD" (or TSVD), and we demonstrate useful applications in the field of geostatistics (and indicate ways in which it will be useful in other areas).

Applications of the SVD to the tools of geostatistics are described: in particular, applications dependent on the TSVD, including variogram modelling in coregionalization. Variogram analysis in general is explored, and we propose broader use of an old tool (which we call the "corhogram", based on the variogram) which proves useful in helping one choose variables for multivariate interpolation.

The reasoning behind kriging and cokriging is discussed, and a better algorithm for solving the cokriging equations is developed, which results in simultaneous kriging estimates for comparison with those obtained from cokriging. Links from kriging systems to kernel systems are made; discovering kernels equivalent to kriging systems will be useful in the case where data are plentiful.

Finally, some results of the application of geostatistical techniques to a data set concerning nitrate pollution in the West Salt River Valley of Arizona are described.

Chapter 1

Introduction

The work which follows springs from two sources. The first was a research assistantship with Dr. Donald Myers, of the Department of Mathematics at the University of Arizona, which entailed the development of geostatistical software. In the course of developing and testing new procedures, ideas would crop up which demanded experimentation: this occasionally led to useful results, which could be incorporated into our thinking (usually, of course, it amounted to nothing!). So the implementation of ideas and a continual examination of the problems of spatial interpolation led us to the creation of new tools, which would aid us in our work.

The second source was a consulting position with the United States Geological Survey (USGS), which required the spatial analysis of a data set concerning water pollution (specifically nitrate pollution) in an area around Phoenix, Arizona. Hundreds of water analyses from wells in an area encompassing some 50 kilometers square served as the database. The problems posed included making the best maps of the nitrate concentrations for a series of periods, in order to provide a baseline or some point of reference for future work. In order to carry out our work, a consideration of various spatial analysis (in particular, geostatistical) techniques was begun, and from that consideration arose many results presented herein.

First of all, let us decide "what is spatial analysis?", and "why is it necessary?" Spatial analysis is essential when studying phenomena which have coordinates, usually spatial (but including temporal and more exotic coordinates), attached. Spatial analysis determines the level to which values of the variables of interest are related according to position from neighboring sites. One might argue that it is unnecessary if there is no spatial correlation in the data, which leads into the following proposition: unless one knows *a priori* that no spatial (temporal) dependence exists in a problem, then one should avail oneself of some techniques of spatial analysis to make a test. This explains the necessity of spatial analysis.

We present some results about several spatial analysis techniques found in that branch of spatial analysis called "Geostatistics", a word coined by Matheron in 1962 [59]. Journel [46] notes that "geostatistics has been defined commonly as the application of the 'Theory of Regionalized Variables' to the study of spatially distributed data." The theory of regionalized variables was developed by Matheron as early as 1962, in works such as [57] and [58]; he is responsible, more than any other single person, for the development of this field (and, whether he would accept the title or not, he undoubtably deserves to be called the "Father of Geostatistics").

Journel goes on to redefine geostatistics as "...a branch of statistics dealing with

spatial phenomena." We take this broader statement as the operational definition, with the caveat that we include temporal problems, as well as other sorts of coordinate systems.

We begin with a thorough discussion of a powerful mathematical tool: the Singular Value Decomposition (SVD). We describe the importance of the SVD to many of the techniques presently used in geostatistics, and in interpolation and estimation problems. Proceeding from characteristics of the SVD, we develop what is currently an essentially unknown technique, the Tensor SVD (TSVD), proving its existence in the three-dimensional case, and demonstrating its utility in problems of geostatistical importance. An algorithm is given for calculating the TSVD of an arbitrary tensor in three-dimensions.

Two of the most important techniques of geostatistics are the multivariate techniques of variogram matrix analysis and the interpolation method known as cokriging, which is based on the results of the variogram and cross-variogram modelling. We have made several improvements in both the understanding and the implementation of these.

We show that the sample variogram matrix is a spatial decomposition of the sample covariance matrix, which aids both in interpreting and modelling the variograms (spatial decompositions of variances) and cross-variograms (spatial decompositions of covariances). We promote the study of a spatial statistic which we call the "corhogram", also known as the codispersion coefficient, which is a spatial decomposition of the correlation between two variables. The corhogram is useful as a modelling tool, and as a means for helping decide when multivariate estimation will give results superior to univariate estimation.

The most important current application of the TSVD is as a tool in variogram modelling under the linear coregionalization model. We describe this application in detail, including links between the TSVD and the technique of near-simultaneous diagonalization of matrices, which has already been used to model variogram matrices in the same way.

On the cokriging side, a new way of writing the cokriging system leads to insight into both the solution of the system, and to links with the univariate Cauchy-Schwartz condition which show when the cokriging coefficient matrix may or may not be invertible. This new formulation is an improvement over the original formulation computationally in two ways: its solution requires that smaller matrices be inverted; and individual kriging results are obtained in the process of obtaining the cokriging results. Since kriging results are generally computed anyway, we show that the cokriging results require only the additional inversion of smaller matrices of low condition numbers.

In addition, the formal solution of the new formulation leads to a first-order approximation to the cokriging solution, which is useful if the cross-variogram terms are small. The first-order approximation involves no additional matrix inversions.

In Chapter Five we describe how we are proceeding toward the discovery of techniques which approximate the results of cokriging. Large linear systems occasionally arise in the process of obtaining cokriging estimates, especially in the case of global cokriging. Since the size of the matrices depends on the numbers of data locations and variables used, the addition of more data points or additional variables may prove detrimental in a cokriging scheme: the larger the coefficient matrix of a cokriging system, the more likely it is to become ill-conditioned, and so give results whose reliability may be questionable. This is quite obviously an undesirable feature of any interpolation or estimation process.

We seek kernel functions, determined in part by the variogram model used in the kriging system, which determine data weights in good agreement with those weights obtained by solving the kriging system. This substitution of kernels for kriging should be particularly useful in cases where data locations are many and well-dispersed.

We show that the weights obtained from kriging using a variety of models resemble the weight patterns obtained from kernels. We then show how one may use integral equations to obtain a kernel which successfully approximates the weights obtained using an exponential model. The technique of passing between the infinite and finite dimensional problems is explored.

Finally, we describe certain applications of geostatistical techniques, including some of those described here, to the Nitrate study mentioned above. In particular, we compare interpolation techniques using cross-validation statistics.

Chapter 2

The Singular Value Decomposition and a Tensor Generalization

In many areas of pure and applied mathematics use is made of a fundamental result known as the **Singular Value Decomposition**, or **SVD**. This decomposition is the result of a theorem, sometimes called the Singular Value Theorem [40], which says that a matrix can be "decomposed" into an optimal sum of outer-products of vectors: that is, that a high dimensional object can be represented as a sum of products of lower dimensional ones. This is useful because it is sometimes easier to analyze low dimensional objects than high dimensional ones, and some mathematical procedures apply to vectors, but not to matrices. Furthermore, it allows for information contained in a matrix, which is scattered about the matrix, to be "packed" into blocks which are more easily analyzed and studied. If there is redundant information in the matrix, the SVD indicates this and even provides a measure of just how much information is actually contained.

The SVD is much overlooked in mathematics, in spite of statements like this from a well-known mathematician of our time (Gilbert Strang, of MIT, author of a definitive undergraduate textbook on linear algebra [87]): "...[the Singular Value Decomposition] is not nearly as famous as it should be." [1] Hoechsmann [40] adds that "the Singular Value Theorem...is not only one of the nicest matrix theorems to state and to visualize but also one of the easiest to prove and to apply. In any introductory course on matrices it deserves a place near the center."

The history of the SVD is documented in a recent article by Stewart [85] in *SIAM Review*. Stewart dedicated it to Gene Golub, who developed the "workhorse" algorithm [32] most often used in its calculation. Applications of the SVD are found in many areas of mathematics: in statistics, where it is used to uncover relationships that exist between different variables, and where it is essential in least-squares problems; physics (where, for example, the PDE solution procedure known as "separation of variables" is actually an example of rank-one solutions of infinite dimensional problems); image processing (data compression and storage, noise removal, rapid image transmission); in cryptology [27]; and in many other areas, including estimation, or, as one could call it, "map-making" - where it is used as a tool for structure-recognition or as a tool for making quick "first-pass" maps of data defined on a grid. This last application will be described in detail.

Data is not restricted to the two dimensions of a sheet of paper or a sidewalk,

[1] Gilbert Strang, *Linear Algebra and Its Application*, Second edition, p 142.

however: there are times when it may be defined on a three-dimensional lattice (for example, the steel frame of a skyscraper); or one can think of the images on a film strip as a three-dimensional stack of matrices in time. Nor is this limited to three-dimensions: the space-time of our world is four-dimensional, and the data in a categorical statistical analysis can take an arbitrary number of dimensions. Can one carry out a sort of singular value decomposition of those structures, too?

It appears that one can indeed. There is an extension of the SVD, to tensors, which will be called the Tensor Singular Value Decomposition (TSVD). This extension should be a boon to those in many areas of research, and a number of useful applications will be described. Although we develop it fully only in the three-tensor case (of the skyscraper or filmstrip above), and do not prove its existence in general, we conjecture that it extends fully to tensors of all dimensions.

2.1 The Singular Value Decomposition of a Matrix

We begin with a review of the Singular Value Decomposition of matrices, proceeding in such a way that one will be able to understand both the motivation for the generalization and the route taken to it. (Good references in a similar vein, but without mention of the generalization, include [4] and [22].)

The place to start is with a statement of the Singular Value Theorem, and its proof.

2.1.1 Hoechsmann's Proof

Hoechsmann's simple statement and proof of the following "Singular Value Theorem" [40] bears repeating. His presentation will be expanded, however, to make it more complete. This proof and results to follow rely on two important theorems from analysis: 1) the Heine-Borel-Lesbesgue theorem, which states that a subset of Euclidean n-space is compact iff it is closed and bounded; and 2) that a real-valued function takes a maximum on a compact set [50].

Theorem 2.1.1. *Let $A \neq 0$ be an $m \times n$ real matrix. Then there exist orthogonal matrices U and V such that*

$$U^T A V = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \text{whereD} = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_r \end{bmatrix} \text{withd}_\text{i} \geq \text{d}_\text{i+1} > 0. \tag{2.1.1}$$

Proof: Let $M(k)$ be the set of all $k \times k$ matrices, and $O(k)$ be the set of all $k \times k$ orthogonal matrices. That is, the subset of $M(k)$ satisfying the k^2 algebraic equations given by the single matrix equation

$$f(Q) \equiv Q^T Q = I \ \ \forall Q \in O(k). \tag{2.1.2}$$

For $U \in O(m)$ and $V \in O(n)$, let $\alpha(U, V)$ stand for the entry in the upper left corner of $U^T AV$.

Now $O(k)$ is a compact subset of the metric space $\mathbb{R}^{k^2}$, with the Frobenius norm ($\mathbb{R}$ represents the real numbers): it is bounded, as every element has norm k, and it is closed, by the following argument: let Q_i be a sequence of orthogonal matrices. The limit point, Q, is in $M(k)$ since it is complete. The question is: is it in $O(k)$? It is, because the function f of equation (2.1.2) is a continuous function of its matrix argument, so pass the limit inside:

$$\lim_{i\to\infty} f(Q_i) = f(\lim_{i\to\infty} Q_i) = f(Q) = I$$

(as the limit of the right-hand side of equation (2.1.2) is always I).

Clearly α is a continuous function of the pair (U, V), and therefore attains a maximal value $d_1 > 0$ on the set $O(m) \times O(n)$ (which is compact as the tensor product of two compact sets).

Let $d_1 = U_1^T AV_1$; then

$$U_1^T AV_1 = \begin{bmatrix} d_1 & Y \\ X & A_1 \end{bmatrix},$$

where A_1 is an $(m-1) \times (n-1)$ matrix. Now X and Y are actually *zero* rows and columns: if X were non-trivial, then the first row (call it ρ_1, notational pun intended!) of $U_1^T AV_1$ would have length $d > d_1$. Then one could simply multiply on the right by the reflection matrix H which takes ρ_1 to $[d, 0, \ldots, 0]$. For analogous reasons, involving columns and left multiplication, it follows that $Y = 0$. So

$$U_1^T AV_1 = \begin{bmatrix} d_1 & 0 \\ 0 & A_1 \end{bmatrix},$$

No row or column of A_1 can have length $d > d_1$ either, for similar reasons: if it did, then a permutation matrix could move it into the position of X or Y, and then a reflection matrix could pile its weight up onto the upper left corner. An inductive argument finishes the proof. Suppose that the theorem is true up to order k:

$$U_k^T U_{k-1}^T \cdots U_1^T AV_1 \cdots V_{k-1} V_k = \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & d_k & 0 \\ 0 & 0 & 0 & A_k \end{bmatrix},$$

with $d_i \geq di + 1 > 0 \ \forall \ i < k-1$. Note that, in order to leave the previous k d_i unchanged from previous steps, successive matrices U_{k+1}, U_{k+2}, etc. will have the $k \times k$ identity in the upper left corner, and zeros in the rest of the first k rows and columns. Then, if A_k is zero, the proof is done. If not, one once again finds an

$(m-k)\times(m-k)$ matrix U'_{k+1} and an $(n-k)\times(n-k)$ matrix V'_{k+1} such that

$$U'^{T}_{k+1}A_kV'_{k+1}=\begin{bmatrix} d_{k+1} & 0 \\ 0 & A_{k+1}\end{bmatrix},$$

Embed these matrices U'_{k+1} and V'_{k+1} in the matrices U_{k+1} and V_{k+1} such that the $k\times k$ identity is in the upper left corner of each, with zeros in the other first k rows and columns: then

$$U^T_{k+1}U^T_kU^T_{k-1}\cdots U^T_1AV_1\cdots V_{k-1}V_kV_{k+1}=\begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & d_k & 0 & 0 \\ 0 & 0 & 0 & d_{k+1} & 0 \\ 0 & 0 & 0 & 0 & A_{k+1}\end{bmatrix},$$

which concludes the proof by induction.

2.1.2 The Operator Tao of SVD

(In this subsection, and in subsections to follow, we use the word "tao" to mean a certain "path" or "way of thinking" by which one arrives at the SVD of a matrix. We also often use the somewhat strange dimensions of $N\times p$ for matrices: this reflects the fact that we often think of the matrices as representing sites, of which there are N, and variables, of which there are p. *C'est la vie!*)

A matrix X can be thought of as a bounded linear operator on a vector space V: it takes vectors in the row space of X (which are elements of V) to vectors in its column space. The image vector is a continuous function of the argument vector, as is the norm of the image vector. The operator norm of X is given by

$$\|X\|=\max\frac{\|\mathrm{Xv}\|}{\|\mathrm{v}\|},\quad \mathrm{v}\neq 0,$$

or, better yet (from the geometrical perspective),

$$\|X\|=\max\|\mathrm{Xv}\|,\quad \|\mathrm{v}\|=1,$$

where the norm of the vector v is the usual L_2-norm. This matrix norm is understood, in a geometrical way, as the value of the greatest stretch of the unit sphere (the second definition above) under application of X (see [39] for this and other geometrical ideas related to concepts from linear algebra).

It is clear that the norm of X must attain a maximum value on the set of vectors described by the closed and bounded unit sphere, as a continuous real-valued function on any compact set has a maximum. A vector such that this maximum is obtained will be called a **Principal Singular Vector** of X (it is not necessarily unique). The

norm of the transformation will be called the **Principal Singular Value** of X (it is unique). This is perhaps the most intuitive way to begin to think of the SVD problem, illustrated by the following question:

Given a matrix X of dimensions $N \times p$, is there a unit vector $\underline{q} \in V$ such that the norm of the product $X\underline{q}$ is maximal: that is, such that

$$X\underline{q} = \lambda \underline{e},$$

with $\|\underline{e}\| = 1$ and λ maximal? Or rather, can one maximize $f(\underline{q}) = \underline{q}^T X^T X \underline{q}$ over all unit vectors $\underline{q} \in V$?

As f is continuous and bounded (by the boundedness of the components of X) on the (closed) unit sphere V, it must attain a maximum on V. This proves the existence of a PSVector and the PSValue, and everything else for that matter: for all the rest follows essentially from that first PSVector, as one can successively remove the outer-product of those vectors from X, and iterate on the reduced matrices (projections removed)

$$X^{(k)} = X - \sum_{i=1}^{k} \lambda_i \underline{e}_i^T \underline{q}_i.$$

This is the essence of Hoechsmann's proof. Removing a vector from the domain reduces the dimension of the sphere, but leaves the succeeding domain closed and bounded: e.g., $V_1 = \{\underline{q} \in V \ni \underline{q} \perp \underline{q}_1\}$.

Each of these reduced matrices also has a PSVector, and PSValue, which must live in the remainder of the space, and this continue inductively like so until one arrives at the zero matrix, and the decomposition is at an end. (According to Stewart [85], this "deflation" approach is due to Jordan.) In fact, this is a sort of restatement of the Eckart-Young theorem, which says that the matrix X can be best approximated by this sequence of sums of operators of increasing rank. This approximation property was due to Erhard Schmidt, of Gram-Schmidt fame, and was rediscovered by Eckart and Young in the context of matrices (Schmidt worked in the realm of continuous kernels, rather than matrices, and developed the continuous version of the SVD).

Thus the link between the singular values and the linear operator is laid bare, and, in particular, the norm of the operator (or the norms of a series of operators, obtained by iteration and deflation). These norms, and the corresponding progression of singular vectors, give rise to one useful representation of the SVD as a matrix sum of dyadic (rank-one outer-product) terms:

$$X = \sum_{i=1}^{p} \lambda_i \underline{e}_i^T \underline{q}_i. \tag{2.1.3}$$

This implies that the original operator is composed of a series of (bi-)orthogonal rank-one operators of decreasing importance (as measured by the decline in the singular

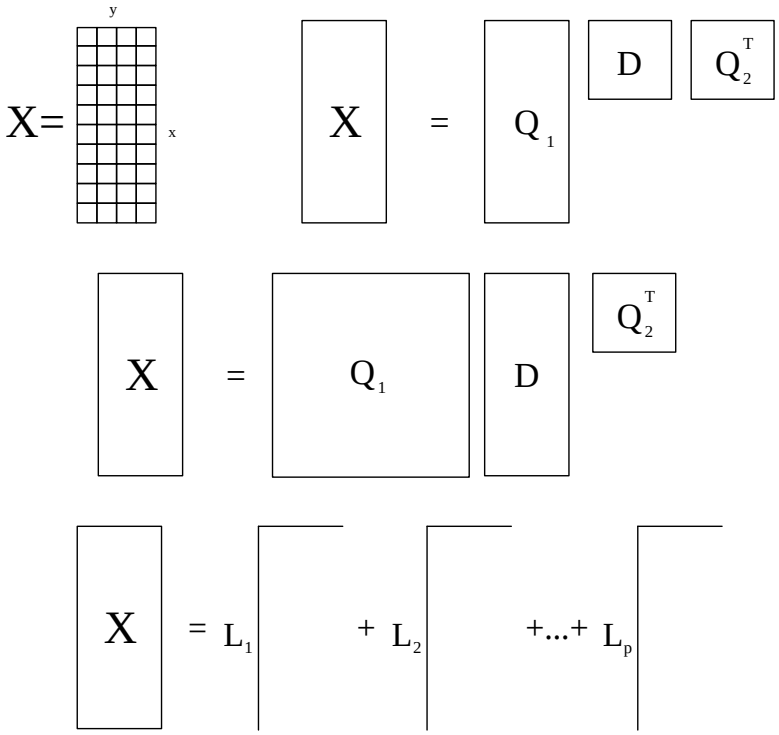


FIGURE 2.1. Three different operator decompositions of a matrix X.

values, or norms). The operators are "bi-orthogonal" since

$$\underline{e}_i \perp \underline{e}_j \text{and} \underline{q}_i \perp \underline{q}_j$$

for $i \neq j$. The pairs $\underline{e}_i$ and $\underline{q}_i$ are known as "Schmidt Pairs" [22], after Erhard Schmidt.

A second useful "operator-oriented" representation is as a matrix product,

$$X = Q_1 \Lambda Q_2^T,$$

which says that the matrix X can be decoupled as a product of orthogonal matrix Q_1, diagonal matrix Λ, and orthogonal matrix Q_2 (or its transpose, really). This is a good representation from the point of view of the geometrical action of the operator: that it rotates the whole $p-$space, then expands (or contracts) p-space (squashing some dimensions, if any singular values are zero), then rotates back (into $N-$space).

It has been our experience that some problems are more profitably pursued while thinking in terms of the former framework (outer-products), while others are better considered in terms of the latter (products of matrices). The particular generalization of the SVD, alluded to in the title of this chapter, is better imagined in the former sense - the outer-product sense (the last form in Figure (2.1)).

In fact, it is worth noting that one can represent X as a product of matrices in two fashions: if $N > p$, then X will not be of rank-N, and will not be square. Extending the matrix Q_1 of Figure (2.1) to be square entails adding $N-p$ rows of zeros to the D matrix (to get the dimensions right). It is obviously "wasteful" to do so from certain standpoints, but from the standpoint of proofs [40], say, or even from the operator standpoint of orthogonal matrices and rotations, it is sometimes better to think of X as the product of a square $N \times N$ matrix, the $N \times p$ matrix Λ, and the $p \times p$ matrix Q_2^T (the middle form of Figure (2.1)).

Thus there are at least three different ways of representing the same decomposition: the choice is related to the applications that one has in mind.

The Operator Tao tells us that one can represent $X_{N\times p}$ as $Q_1 \Lambda Q_2^T$: what that really means is that if one is willing to change to other bases, for both the row and column spaces, then one will have diagonalized the operator X (it will have the diagonal representation Λ); then the effect of taking an inner-product of X with some other matrix, or letting X operate on either $p-$space or $N-$space, will be easily understood in this coordinate system. Remember that vectors are not determined by their coordinates, but only by their coordinates relative to some basis: changing the basis does not change the vector, only its representation.

2.1.3 The Eigen Tao of SVD

In the previous section a relationship between the SVD and the quadratic form $\underline{q}^T X^T X \underline{q}$ was disclosed. In this section, the statement of the PSValue/PSVector problem above will be rephrased equivalently as an eigenvalue problem.

Given a matrix X of dimensions $N \times p$, is there a unit vector $\underline{q}$ which maximizes the quadratic form

$$\underline{q}^T X^T X \underline{q} = \lambda^2? \tag{2.1.4}$$

The requirement that $\underline{q}$ be a unit vector adds a Lagrange multiplier to the optimization problem: i.e., maximize

$$E(\underline{q}, \lambda) = \underline{q}^T X^T X \underline{q} - \mu(\sum_{i=1}^{p} q_i^2 - 1).$$

Differentiating with respect to the q_i and μ gives a system of $p+1$ equations: for the k^{th} component of $\underline{q}$

$$\sum_{i=1}^{N} \sum_{j=1}^{p} x_{ik} x_{ij} q_j - \mu q_k = 0, \tag{2.1.5}$$

with the constraint that

$$\sum_{k=1}^{p} q_k^2 = 1. \tag{2.1.6}$$

The first set of equations, (2.1.5), written in matrix form, is recognizable as the eigenvalue problem for the matrix $X^T X$:

$$X^T X \underline{q} = \lambda^2 \underline{q},$$

and (since it is linear) any (non-zero) solution can be scaled to meet the condition (2.1.6), which takes care of the constraint. In other words, the Singular Value problem leads directly to the eigenvalue problem for the matrix $X^T X$. This matrix is symmetric and nonnegative definite, which is guaranteed to have a full set of nonnegative eigenvalues λ_i with orthogonal eigenvectors. (For details on nonnegative definite matrices, and their equivalence with the class of covariance matrices, see [3].) The eigenvalues of $X^T X$ are thus the squares of the singular values of the matrix X, and the eigenvectors are the singular vectors of X. In sum: the singular value problem is equivalent to an eigenvalue problem.

Golub notes, however, that it is not necessarily wise to compute the SVD of a matrix via this eigenvalue problem, as it "does violence" to the small singular values in the process of "squaring" the matrix [31, 32].

2.1.4 The Inner Tao of SVD

The outer-product decomposition of X can be considered as an inner product decomposition, on the vector space of $N \times p$ matrices.

The inner-product is given by component-wise multiplication of matrix elements, so that the norm of X (the Frobenius norm) is given by

$$\|X\| \equiv \sqrt{\langle X, X \rangle} = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{p} x_{ij}^2} = \sqrt{\mathrm{Tr(X^T X)}}.$$

Note that

$$\|X\| = \sqrt{\sum_{i=1}^{p} \lambda_i^2},$$

and that the standard Euclidean basis of unit matrices (with one single non-zero entry whose value is "1") is the set of outer-products of the standard bases of p-space and N-space; any other set of all outer product of bases of the respective spaces is also a basis [84].

The outer-product representation of X from above (2.1.3) is also a representation of X in this vector space of matrices:

$$X = \sum_{i=1}^{p} \lambda_i \underline{e}_i^T \underline{q}_i \equiv \sum_{i=1}^{p} \lambda_i X_i,$$

so one can think of the set of X_i as an optimal rank-one subspace-basis with which to represent X ("subspace-basis" because it is not a basis of the whole space but only the span of the X_i, which clearly contains the matrix X). It is optimal in the sense that the projection of X onto the set of rank-one matrices is maximized for X_1, then for $X_2 \perp X_1$, etc.

This puts a new twist on the SVD of X, closer in spirit to the generalization of the SVD to tensors: the fundamental idea is that one can consider tensors as objects in a higher dimensional vector space, and look for best rank-one bases in which to express them.

This idea, of rank-one bases, may be somewhat foreign: in the "usual" vector space (that is, of arrows of variable length as in physics) one cannot represent vectors in anything but rank-one basis elements, as all vectors are rank-one! But in vector spaces of matrices, or tensors more generally, one can set up rank-one bases and argue that this one or that one is more appropriate for a given purpose (operator, etc.).

The question one might ask is: why rank-one bases? What is so special about these simple bases? And again, the answer is: they allow us to separate the linear space into nice little chunks, each of which probes into only one part of space, and which one hopes will have some meaning, or give some insight, or allow for dimension reduction, or provide for an analysis that one cannot achieve so well otherwise.

Let us now show explicitly how the matrix inner-product shows up as one changes bases while performing the SVD. At this point we consider it useful to introduce a notion for outer-product which can be generalized (as we will soon be doing three-dimensional outer-products). We use the symbol "$\otimes$" to denote an outer-product, so that for two dimensions, one can write

$$\underline{u} \otimes \underline{v} \equiv \underline{u}\underline{v}^T.$$

If one writes

$$X = \sum_{i=1}^{p} \sum_{j=1}^{q} \left(\underline{u}_i^1 \otimes \underline{u}_j^2 \right) x_{ij}$$

where $\underline{u}^1$ and $\underline{u}^2$ are the standard Euclidean unit vectors, then one can consider an arbitrary change of basis such that

$$I_p = EE^T \mathrm{and} \mathrm{I_q} = \mathrm{FF^T},$$

where the columns of E and F are the new basis vectors. Writing the unit vectors in these new coordinates leads to

$$\begin{aligned} X &= \sum_{i=1}^{p} \sum_{j=1}^{q} \left(\sum_{k=1}^{p} e_{ki} \underline{e}_k \right) \otimes \left(\sum_{l=1}^{q} f_{lj} \underline{f}_l \right) x_{ij} \\ &= \sum_{k=1}^{p} \sum_{l=1}^{q} \left(\underline{e}_k \otimes \underline{f}_l \right) \left(\sum_{i=1}^{p} \sum_{j=1}^{q} e_{ki} x_{ij} f_{lj} \right) \end{aligned}$$

$$= \sum_{k=1}^{p} \sum_{l=1}^{q} \left(\underline{e}_k \otimes \underline{f}_l \right) \underline{e}_k^T X \underline{f}_l$$

$$= \sum_{k=1}^{p} \sum_{l=1}^{q} \left(\underline{e}_k \otimes \underline{f}_l \right) \langle \underline{e}_k \otimes \underline{f}_l, X \rangle$$

The SVD suggests that one choose them in such a way that the inner-products $\langle \underline{e}_k \otimes \underline{f}_l, X \rangle = \alpha$ of X with rank-one matrices are successively maximized.

Having found the first vectors, $\underline{e}_1$ and $\underline{f}_1$, and hence the first matrix basis element, note that

$$\langle \underline{e}_1 \otimes \underline{f}_1^{\perp}, X \rangle = \langle \underline{e}_1^{\perp} \otimes \underline{f}_1, X \rangle = 0,$$

for, if not, then without loss of generality (WLOG) $\exists \underline{f} \in \underline{f}_1^{\perp}$ such that

$$\langle \underline{e}_1 \otimes \underline{f}, X \rangle = \beta,$$

which suggests that one then consider

$$\langle \underline{e}_1 \otimes \left(\frac{\alpha \underline{f}_1 + \beta \underline{f}}{\sqrt{\alpha^2 + \beta^2}} \right), X \rangle = \sqrt{\alpha^2 + \beta^2} > \alpha,$$

which is a contradiction of the claim that α was maximal, corresponding to $\underline{e}_1 \otimes \underline{f}_1$.

Thus, having chosen $\underline{e}_1 \otimes \underline{f}_1$,

$$X = \left(\underline{e}_1 \otimes \underline{f}_1 \right) \langle \underline{e}_1 \otimes \underline{f}_1, X \rangle + \sum_{k=2}^{p} \sum_{l=2}^{q} \left(\underline{e}_k \otimes \underline{f}_l \right) \langle \underline{e}_k \otimes \underline{f}_l, X \rangle.$$

The space is deflated like so, and the search for new singular vectors is reduced to the spaces perpendicular to $\underline{e}_1$ and $\underline{f}_1$, which are each smaller in dimension by one. The first choice has led to the loss of $p + q - 1$ degrees of freedom in the search for singular vectors, from which the total number of degrees of freedom remaining is

$$pq - ((p + q) - 1) = (p - 1)(q - 1).$$

This continues inductively until the space is reduced to the empty set: that is, until p or q is reached, whichever is smaller (the usual "full-rank" case for a matrix).

2.1.5 The Meaning of Singular Things

The meaning of the singular values changes, depending on the application one makes of them. For example, in the operator sense they represent the norms of a set of rank-one operators which best approximate the original operator by another operator of a given rank. In this case, the left singular vectors represent the successively less important parts of the range space, and the right singular vectors represent

successively less important parts of the complement of the null-space of the matrix operator.

From the "eigen-standpoint", note that, as the eigenvectors of $X^T X$ are mutually orthogonal, so are the singular vectors of X. Statisticians like to think of the values of λ^2 as variance, and use the decomposition on the mean-centered and unit-scaled matrix X to describe its variance structure in the procedure known as "Principal Components Analysis", or PCA (PCA will be described in detail in the next chapter). In this case, the intuitive understanding of the orthogonality and singular values is as follows: one can think of the rows of X as a set of points (in p-space); that is, as a point-cloud in p-space. This point-cloud could be best-fit by an ellipsoid. That ellipsoid has a direction along which it is most elongated, which, in statistical jargon, means that it has maximal variance; and that direction (a vector, referred to the center of mass of the point cloud) is the PSVector. The ellipsoid is described further by its dimensions along its axes orthogonal to that PSVector, and these additional directions are successive (by size) singular vectors.

In an image analysis problem, singular values are treated as amount of "information", and the corresponding Schmidt pair outer-products as decreasingly important basis images, or "eigenimages" [4]. In data compression, the sum of the retained singular values (or their squares) represent the proportion of the picture which will be reconstituted by the corresponding "singular pictures" (i.e., the dyadic pairs: see Figure (2.4), in which is shown an image of Abraham Lincoln with only half of the total information of the matrix).

2.2 The SVD as a Rapid Interpolator on a Grid

One can use the SVD of a matrix of gridded data (in two-dimensional space) to generate an interpolator of that data. This method has been elaborated in Preisendorfer [76], in his treatment of Principal Component Analysis[2]. Preisendorfer gives an extended development of PCA from the standpoint of spatial/temporal interpolation, relating the singular vectors to samples from continuous populations, although he did not provide as much detail as follows.

Let X represent an $n \times m$ matrix of data locations, taken from a grid, or tracks, in a field. Without loss of generality, assume that $n \geq m$. Think of the tracks as lying perpendicular or parallel to each other, although it is not necessary that the tracks occur with equal spacing (Figure (2.2)). The SVD of X is

$$X = Q_1 \Lambda Q_2^T,$$

[2]Page 25 and following material, only in reverse: he describes how singular vectors converge to eigenmodes, as data of a continuous phenomenon are gridded at finer and finer meshes. Thus, he describes how one can use the singular vectors to guess the form of continuous objects. One can turn around and ask that singular vectors from a fixed grid give us a guess as to what the continuous object would be, by interpolating or estimating the singular vectors.

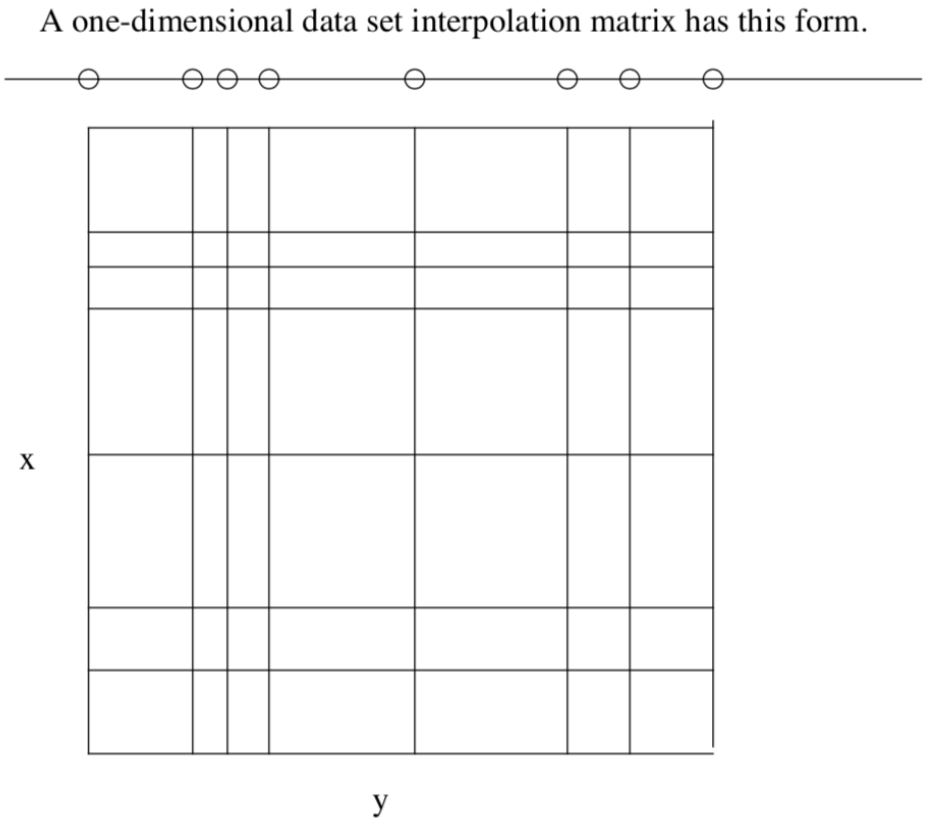


FIGURE 2.2. A two-dimensional data set of tracks in a field. It is not necessary that the tracks show the symmetry represented in this figure. Although not yet discussed, this is also the form of an interpolation matrix for a one-dimensional data set.

where Q_1 and Q_2 are orthogonal matrices of dimensions $n \times m$ and $m \times m$, respectively, whose columns are eigenvectors of XX^T and X^TX. Λ is also $m \times m$, and diagonal with positive entries.

If one interpolates the eigenvectors (using any interpolation scheme whatsoever) which make up Q_1 and Q_2 (while respecting the real distances which exist between their entries), then one will have successfully created a matrix function which interpolates X:

$$X(x, y) = Q_1(x)\Lambda Q_2(y)^T$$

(If the vector constituents of Q_1 and Q_2 had been merely estimated, then one would have estimated, rather than interpolated, X.)

In order to get an estimate of a row off of "the beaten tracks", at x_0 say, use the function

$$X(x_0, y) = Q_1(x_0)\Lambda Q_2(y)^T$$

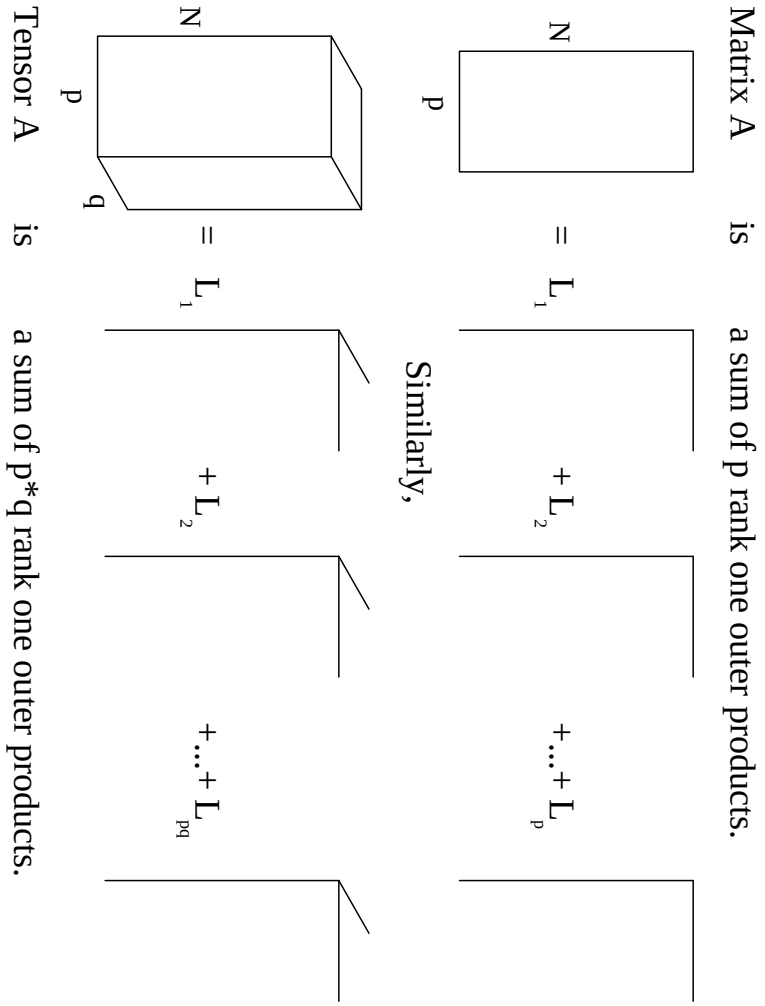


FIGURE 2.3. The SVD Interpolation Scheme: a grid is turned into two finite sets of functions, whose outer-products form functions of two variables. "Rows" (tracks running left to right) of the interpolating function are gotten by taking sums of the interpolated rows of V; "columns" by taking sums of interpolated columns of Q.

sketched in Figure (2.3). The entries of Q_1 at x_0 are computed, then one treats the rows of Q_2^T as functions. Similarly, for a column at y_0, use

$$X(x, y_0) = Q_1(x)\Lambda Q_2(y_0)^T \tag{2.2.7}$$

(Figure (2.3)). Note that a matrix multiplication is not really needed each time, as Λ is diagonal. What really results is a function

$$X(x, y_0) = \sum_{i=1}^{m} q_{1,i}(x)\lambda_i q_{2,i}(y_0) = \sum_{i=1}^{m} c_i q_{1,i}(x)$$

which can be computed at any point x (inside, or even outside, the convex hull of the data). If one estimates beyond the bounds of the grid, then the extrapolatory properties of the estimate are determined by those of the interpolator chosen for the vectors of Q_1 and Q_2: that is, if a cubic scheme were used, then there will be cubic growth, rather than a tendency to the mean, etc.

If the function need not interpolate the matrix (that is, pass through the points on the matrix), but only estimate it (smoothing, for example), then one can consider the usual practice of eliminating those rows and columns of Q_1 and Q_2 which corresponds to the small (presumably negligible) singular values. In fact, many of these may truly be negligible, and would hence permit us to make faster estimates.

The eigenvectors can be very smooth, a result of ordering the data properly (ordering is crucial, and accounts for the problem in generalizing this method beyond one-dimensional problems). Changing the order of rows in a matrix only changes the form of the column singular vectors: it has no effect on either the singular values or the row singular vectors. This smoothness implies that it may be relatively simple and painless to interpolate them (e.g. one might get away with linear interpolants of the singular vectors).

2.2.1 Illustrative Example: Abe

C. Long [53] obtained points from a statue of Abraham Lincoln on a 49×36 grid, which he then used to study properties of the SVD, the pseudo-inverse, and other interesting topics. We use the same data set to demonstrate the interpolation described above.

The matrix of data, 49×36, represents the 16th president of the United States, and interpolates a statue of Abe. The SVD of the matrix was computed, and a plot of Abe derived using linear interpolants of the singular vectors, taking values on a 100×70 grid (roughly twice the size in each direction), is compared to a plot of the original matrix in Figure (2.4). As one can see, the technique does a nice job of filling in Abe's face, smoothing it into a more natural image; and it is done quickly and easily.

This is a very general method, in the sense that there are an infinite number of implementations of it (corresponding to the various interpolators and estimators

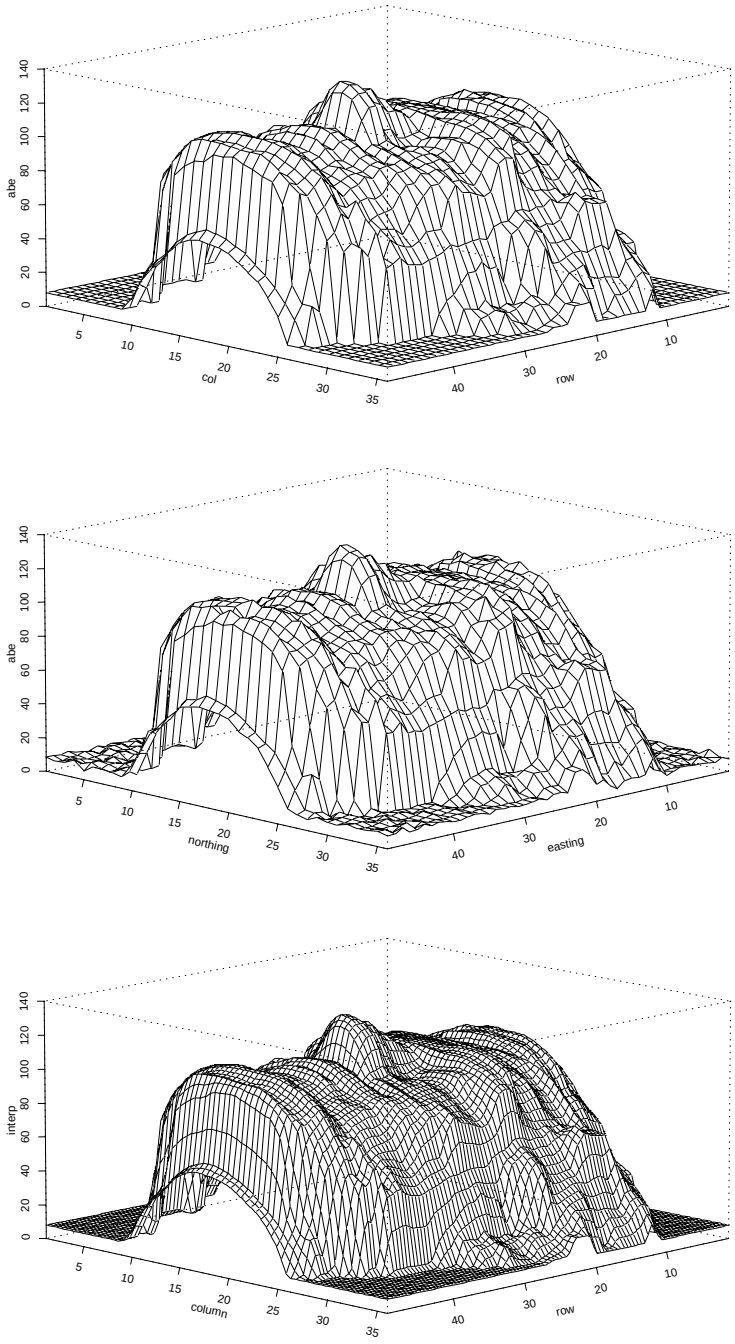


FIGURE 2.4. Abe himself, Abe represented using only half the information in his singular values and Schmidt pairs), and Abe "densified" by a spline-fitting his singular vectors.

that one might use), and each will give different results. A linear interpolant of the singular vectors is the safest, in that estimates cannot stray from the convex hull (in space) of the data: that is, values estimated on a line between adjoining grid points will be between the values at the gridpoints.

The question of the optimality of the method will not be discussed. The point to make is simply that this method exists, and works well at filling in gridded data. It provides one with a functional interpolator, or estimator (if one were to discard several singular values, for example, or if one merely estimated the singular vectors), which can even be used to extrapolate the gridded data; but again, the properties of the extrapolation will be those of the interpolator/estimator used on the singular vectors.

A comparison of this method to other methods, and a comparison of singular vector interpolators within this method are beyond the scope of this dissertation (but interesting topics, nonetheless!). One thing to note, however, is that, as the SVD really only makes comparisons vertically and horizontally (but never on a diagonal), so will the interpolator/estimator derived from it. One can see that the SVD has this property by considering that its vectors are determined from matrices composed of inner-products of rows or columns: thus, only elements which "collide" in such inner-products are compared (i.e., A_{ij} and A_{kl} such that $i = k$ or $j = l$). On the other hand, when it comes time to estimate or interpolate the singular vectors, it may be that every point on those two lines contributes to an estimate at the point of interest.

2.3 The Three-Dimensional Case

Motivation for the search for a generalization of the SVD came from two completely different areas at the same time: a course in categorical data analysis, using a textbook by Agresti [1] led to consideration of "three-dimensional data sets" (the first specific reference was to "Loglinear Models for Three Dimensions"[3]); and in an unrelated area, we were looking at the SVD as an interpolator (see the previous section), and asked ourselves what one would do if one wanted to treat three-dimensional data sets (e.g., room temperatures, given on a 3-d grid, etc.). "Generalize the SVD!" was the most sensible reply. In sections below are two examples which motivated our search.

2.3.1 Independence is Rank-One

Agresti considered the following problem: Given a three-way contingency table, for variables $X, Y,$ andZ of dimensions $I, J,$ andK (respectively), let $\{m_{ijk}\}$ denote the

[3]Section 5.3 heading, page 143

expected frequencies. Consider the following model for the logs of the frequencies:

$$\log(\mathrm{m_{ijk}}) = \mu + \lambda_{\mathrm{i}}^{\mathrm{X}} + \lambda_{\mathrm{j}}^{\mathrm{Y}} + \lambda_{\mathrm{k}}^{\mathrm{Z}},$$

where

$$\mu = \frac{1}{IJK} \sum_i \sum_j \sum_k \log(\mathrm{m_{ijk}}), \text{and}$$

$$\lambda_i^X = \frac{1}{JK} \sum_j \sum_k \log(\mathrm{m_{ijk}}) - \mu, \text{etc.},$$

which, one notes, implies that

$$\sum_i \lambda_i^X = 0, \text{etc.} \tag{2.3.8}$$

Now, although Agresti does not do so, one could re-write this equation in tensor form as

$$\log(\mathrm{M}) = \mu \underline{1}^{\mathrm{X}} \otimes \underline{1}^{\mathrm{Y}} \otimes \underline{1}^{\mathrm{Z}} + \underline{\lambda}^{\mathrm{X}} \otimes \underline{1}^{\mathrm{Y}} \otimes \underline{1}^{\mathrm{Z}} + \underline{1}^{\mathrm{X}} \otimes \underline{\lambda}^{\mathrm{Y}} \otimes \underline{1}^{\mathrm{Z}} + \underline{1}^{\mathrm{X}} \otimes \underline{1}^{\mathrm{Y}} \otimes \underline{\lambda}^{\mathrm{Z}},$$

where "$\otimes$" means the outer-product of vectors into the appropriate part of three-space (there are three dimensions here). In this way, $\log(M)$ is expressed as a sum of four orthogonal three-tensors (orthogonal due to the "zero-sum" conditions (2.3.8), with term-by-term multiplication (i.e., the Frobenius inner-product)).

It was evident to us that this was the case of independence of the three variables, however, and would be expressed better in terms of the original frequencies as

$$m_{ijk} = e^{\mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z} = e^{\mu} e^{\lambda_i^X} e^{\lambda_j^Y} e^{\lambda_k^Z},$$

or

$$M = c \underline{p}_i \otimes \underline{q}_j \otimes \underline{r}_k :$$

i.e., that the frequencies are given by a rank-one outer-product of three vectors, one in each dimension ($X, Y,$ and$\mathrm{Z}$): this is just another way of expressing the fact that independence requires that $m_{ijk} = c p_i q_j r_k$. Agresti did not represent the case in this tensor-oriented way, although he comments that this model represents independence. Our method seemed a more natural way of considering such a loglinear model, however.

We then began to search for ways of expressing the many other models he considers in terms of outer-products, and the notion of the TSVD popped up. Although we have not pursued this particular avenue, it looks like this might be a good place to begin looking at application of the technique of TSVD outside of interpolation and estimation.

2.3.2 Decomposing Three-Dimensional Objects

The SVD serves admirably as a rapid interpolator of a matrix: but it must be generalized to treat higher dimensional objects. A search for such a generalization yielded nothing, however: at least nothing of the right sort.

Preisendorfer [76] gives an extension of Principal Components Analysis to vector-valued fields (which one could consider as three-tensors) and matrix-valued fields (four-tensors), but it is not the extension that we sought. He essentially in-lined matrices (considering vectors as the special case) into a larger matrix, creating, from an $n \times p$ matrix of matrix components of size $l \times m$, a larger matrix of size $nl \times mp$. He then treated it in the normal PCA fashion. This is not even-handed in its treatment of the four dimensions involved, and completely suppresses two of the dimensions, as the resulting objects will be rank-mp in any case, in spite of the values of both l and n.

Lieven [51], in a personal communication, reports that he has undertaken a generalization, although he has not yet published it. He also claims to have an algorithm for computing it. Comparison of our decompositions on the same tensor shows his to be quite different, however; he has developed a different type of generalization. This brings up the point that there is more than one way to decompose a tensor: but if one wants to get at a true generalization of the SVD, then one needs to justify calling it by that name. A generalization of the SVD should earn its name by generalizing important properties of the SVD, and should include the SVD of a matrix as a special case.

The following problem is considered in the field of object-imaging: given an object, or rather points from an object in three-space, and an array of detectors, determine a linear relationship between the object and the detectors satisfying the equation

$$g_N = H_{N \times pqr} f_{pqr}$$

for f, where f is an image written as a vector composed of columns from the three-tensor of dimension $p \times q \times r$ (columns in an arbitrary dimension). This can be re-phrased as the following inverse problem: given the detector readings, tell us what the image must have been. This technique involves replacing the four-tensor (of dimensions $N \times p \times q \times r$) by H, where H is the $(N \times pqr)$-dimensional matrix with rows of the sample images and columns of detector responses to a set of unit impulses. They hope that those impulses and their responses will serve as a basis in which a typical object can be represented.

This creates a set of singular vectors

$$H = U \Lambda V^T$$

which, in the row space (i.e. the V vectors), are singular unit images (in the Frobenius norm) and in the column space are singular unit responses. The least-square solution

to the problem is then given by using the pseudo-inverse to obtain

$$f_{pqr} = (H^T H)^{-1} H^T g_N = V \Lambda^{-1} U g_N.$$

This image analysis procedure represents a different way of decomposing the tensor, with a preferential direction: the singular images are not rank-one objects, as are the singular images we seek: they are vectors turned back into tensors, with a potential for containing $pqr + N$ pieces of information (whereas a 4-tensor rank-one object would contain at most $p + q + r + N$).

Geladi et al. [30] have discussed applying PCA to "multivariate images", by which they mean a stack of images sorted by frequency, and represented as integers in the range [0,255]. They proceed in a way that seems very natural, taking a modified SVD (not truly PCA, since they do not mean-center their images) of the stack along the frequency dimension, thus obtaining eigenimages (Schmidt pairs) as scores on frequency factors.

This is equivalent to writing the tensor as a matrix: from $T_{i\times j\times k}$ they form $M_{(ij)\times k}$, decomposing that to get the singular vectors in the two directions. They then reform images from the ij vectors, by "unfolding" them.

Their paper represents a struggle with some of the same problems that will appear in this dissertation: notation and representation of three dimensional structures, and the mappings they represent. On the one hand, they did not need to bother, since they could represent their tensors as matrices. But they comment that they realized that they were losing information about the "contextual properties" of the pixels (that is, the fact that nearby pixels are correlated). This is a consequence of the fact that, once the tensor is transformed to a matrix, the ordering of the rows is completely arbitrary: that is, all rows (i.e. pixels) will be compared equally and in ignorance of actual proximity to other pixels.

A few comments on that paper are appropriate: the first is that by avoiding the mean-centering, their title is a mis-nomer; the second is that the eigenimages they obtain can be full rank; and, on a minor note, but one which is important nonetheless, they mistakenly assert that the first principal component "usually" contains all non-negative values. This is too weak a statement, as the Perron-Frobenius Theorem [4, 9] assures us that the principal Schmidt pair of these positive-valued matrices can be chosen so as to have positive values. This is important because the authors were concerned that the eigenimages be positive, so that they could be easily re-interpreted as images (recall that the multivariate images were defined on [0,255]). Of course, this only applies to the first pair: all others will have negative values (in general), and need to be rescaled to that interval.

We therefore seek our own generalization of the SVD of a matrix, motivated by the very geometrical idea that a matrix can be decomposed into an outer-product of vectors and the assumption that such should also be the case in higher dimensions. We have not discovered in it any source material.

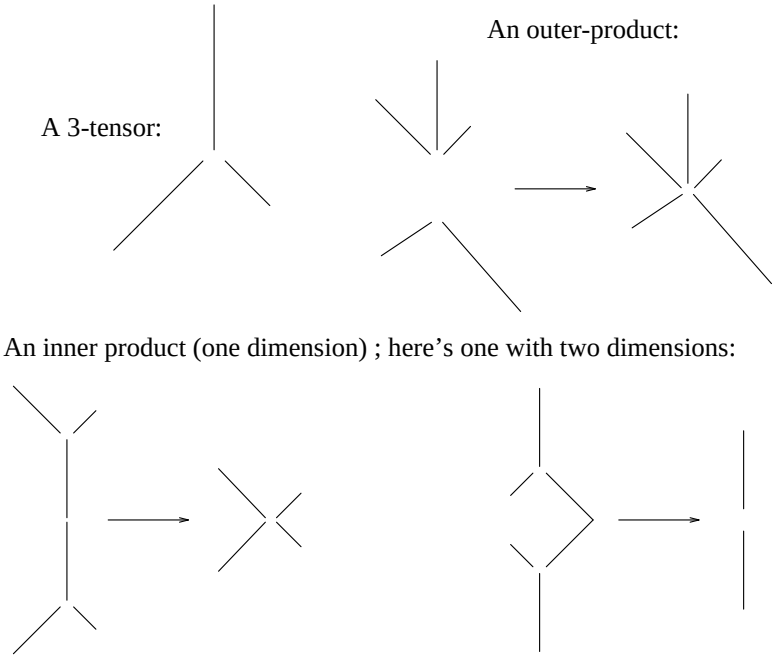


FIGURE 2.5. Tensor inner- and outer-product notation.

2.3.3 A Notation for Tensor Inner- and Outer-Products

For lack of a better way of keeping track of tensor inner and outer products, we "developed" (perhaps remembering it from some other field) the notation of Figure (2.5).

This notation is not meant to imply that the "stick figure" tensors are necessarily rank-one outer-products of vectors, as one might naturally infer: the sticks merely indicate that there are components in a given "direction" of the higher dimensional space which are represented (as well as possible) on a two-dimensional page.

2.3.4 Embedding Matrices into Tensors

In order to help the reader get a better feel for the three-tensor idea, several little "tensor games" are included; these are intended to show the relationships involved, and motivate some of the following material. While the three-tensors will be treated not as operators, but as elements of a vector space, it is still good to recall that each of the "taos" of the matrix case should have some corresponding "tao" in the tensor case. As the existence of the TSVD has not yet been proven, this section may seem premature: but the idea is to stimulate some interest and intuition into the tensor manipulations to follow.

■ Example 1

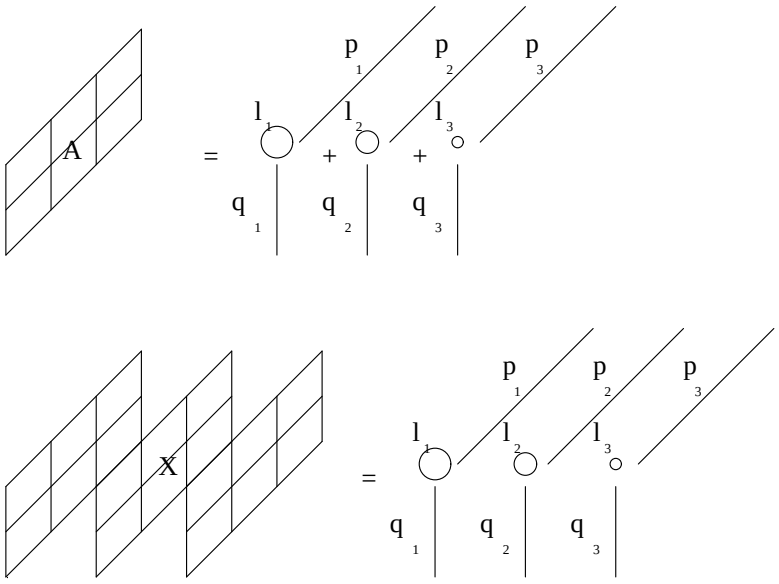


FIGURE 2.6. The most natural three-tensor to create from a matrix?

Consider matrix A as a $1 \times p \times N$ (thin!) three-tensor, and consider embedding the SVD of matrix A (and hence all the information of A), of rank k, in the three-tensor $X_{k \times p \times N}$ (Figure (2.6)). The tensor should then also be rank k, and decompose as each Schmidt pair times a Euclidean unit vector (Figure (2.7): the difference is that the plus signs have disappeared!).

This is a good time to extend the inner-product notation to serve in a more general capacity, beginning with the matrix case: for a matrix/vector product, for example, one could use

$$\langle A, \underline{x} \rangle \equiv A\underline{x}$$

as long as the dimensions are clear, and

$$\langle A_{ij}, \underline{x}_j \rangle \equiv A\underline{x}$$

to mean inner-product on the second dimension, say, if not. For a tensor/matrix product, whose result is a vector, one does the same: i.e.,

$$\langle T, M \rangle$$

as long as the dimensions are clear, and

$$\langle T_{ijk}, M_{ik} \rangle,$$

for example, otherwise. Notice that the second form is just the Einstein Summation Convention (ESC); but the brackets make us happier!

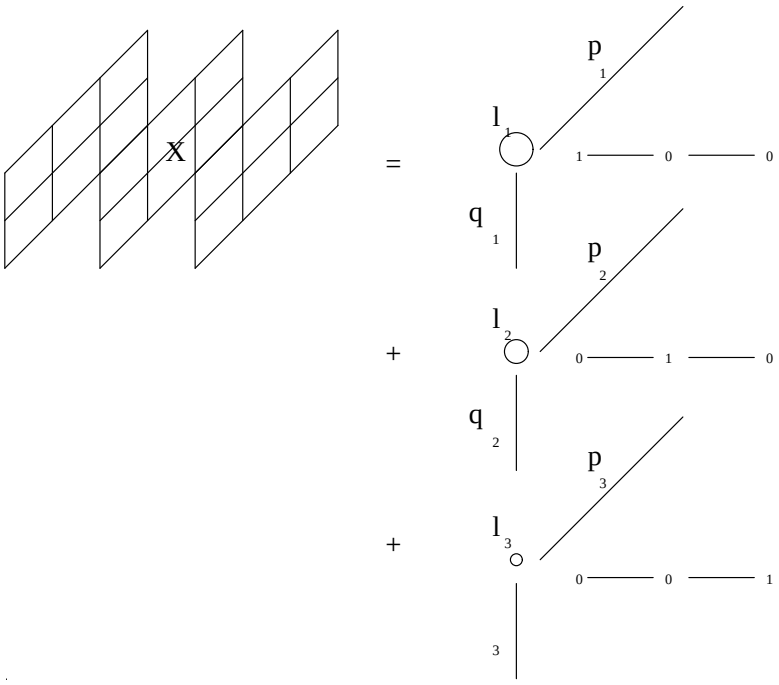


FIGURE 2.7. Tensor SVD of X is simple, given by construction!

Now, if one calculates the tensor products of X with itself, such that two dimensions are modded out by inner-products, then three different resultant matrices result: in the case of the figure, the result is

$$\langle X_{irs}, X_{jrs} \rangle \equiv \Lambda^2 = \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix}$$

$$\langle X_{ris}, X_{rjs} \rangle = \sum_{m=1}^{k} \lambda_m^2 q_m q_m^T = \sum_{m=1}^{k} \lambda_m^2 \begin{bmatrix} q_{m1}^2 & q_{m1}q_{m2} & q_{m1}q_{m3} \\ q_{m2}q_{m1} & q_{m2}^2 & q_{m2}q_{m3} \\ q_{m3}q_{m1} & q_{m3}q_{m2} & q_{m3}^2 \end{bmatrix}$$

$$\langle X_{rsi}, X_{rsj} \rangle = \sum_{m=1}^{k} \lambda_m^2 p_m p_m^T = \sum_{m=1}^{k} \lambda_m^2 \begin{bmatrix} p_{m1}^2 & p_{m1}p_{m2} & p_{m1}p_{m3} & p_{m1}p_{m4} \\ p_{m2}p_{m1} & p_{m2}^2 & p_{m2}p_{m3} & p_{m2}p_{m4} \\ p_{m3}p_{m1} & p_{m3}p_{m2} & p_{m3}^2 & p_{m3}p_{m4} \\ p_{m4}p_{m1} & p_{m4}p_{m2} & p_{m4}p_{m3} & p_{m4}^2 \end{bmatrix}.$$

If one had embedded A as $X = A \otimes e_1$ instead, then only the first matrix (Λ^2) would have been different.

$$X_{irs}X_{jrs} \equiv \Lambda^2 = \begin{bmatrix} \sum_{m=1}^{k} \lambda_m^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

If one had embedded A as $X = A \otimes \frac{1}{\sqrt{3}}(1,1,1)$, then once again only the Λ^2 matrix would have been different:

$$X_{irs}X_{jrs} \equiv \Lambda^2 = \sum_{m=1}^{k} \lambda_m^2 \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} = \sum_{m=1}^{k} \lambda_m^2 \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}.$$

The others remain unchanged.

Thus crossing the matrix A with different unit vectors only changes the first matrix Λ^2, and not the others.

Another way of thinking about these embeddings of matrices into tensors is as operators of the sort

$$T : \mathbb{R}^2 \longrightarrow \mathbb{R}^3,$$

where T represents a tensor acting on A in such a way to result in a three-tensor. For example, one could think of dotting (taking an inner-product of) the matrix A with different unit 5-tensors (eliminating two dimensions): in the first case the three unit 5-tensors used were $q_1 \otimes p_1 \otimes q_1 \otimes p_1 \otimes e_1$, $q_2 \otimes p_2 \otimes q_2 \otimes p_2 \otimes e_2$, and $q_3 \otimes p_3 \otimes q_3 \otimes p_3 \otimes e_3$. These are unit tensors using the 5-tensor Frobenius norm. More explicitly,

$$\langle A, \sum_{i=1}^{3} q_i \otimes p_i \otimes q_i \otimes p_i \otimes e_i \rangle = \sum_{i=1}^{3} \lambda_i q_i \otimes p_i \otimes e_i$$

which is a sum of three three-tensors, as illustrated in Figure (2.6), for example. (In the other two cases described above, the unit tensors used were:

$$q_i \otimes p_i \otimes q_i \otimes p_i \otimes e_1,$$

and

$$q_i \otimes p_i \otimes q_i \otimes p_1 \otimes \frac{1}{\sqrt{3}}(1,1,1).$$

Note that these products could be represented using the special operation notation as in Figure (2.8).

The generalization ultimately derived will decompose these singular tensors as expected.

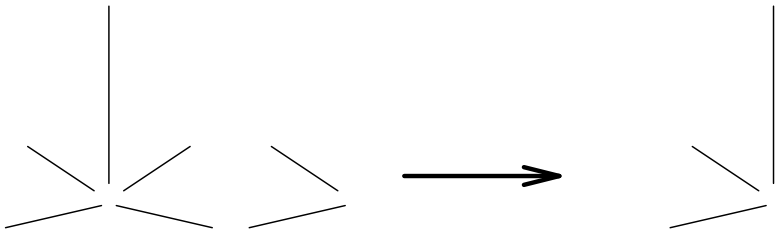
■ Example 2

Consider the simple case of a three-tensor X created by taking the outer-product of a matrix A and a vector v (Figure (2.9)).

If A has SVD $A = Q_1 \Lambda Q_2^T$, then it seems clear (as noted in Example 1) that the TSVD of X should be

$$T = Q_1 \Lambda Q_2^T \otimes \underline{v},$$

and that the rank of T, $R(T)$, should be the same as the rank of A, $R(A)$ (Figure (2.9)).



A 5-tensor dotted with a matrix yields a 3-tensor.

FIGURE 2.8. These are the tensor products referred to in the text

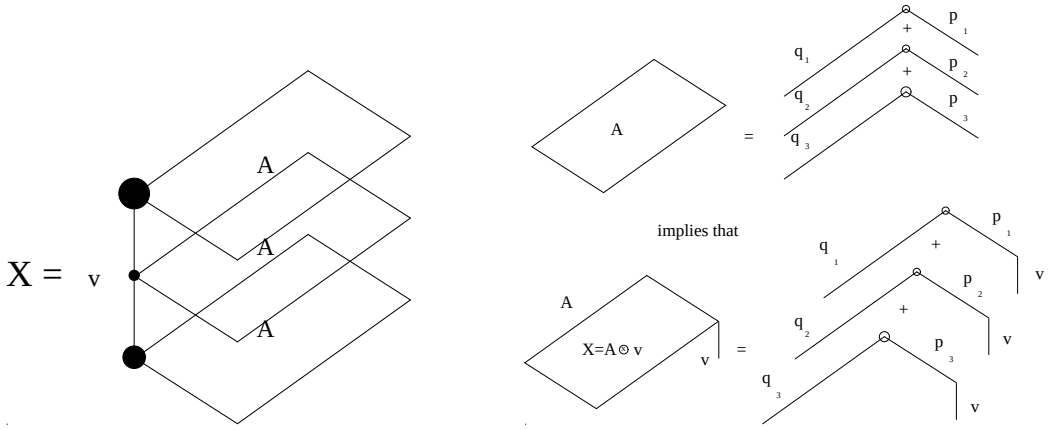


FIGURE 2.9. Left: $X = v \otimes A$ (components of vector v are indicated by their size as balls). Right: both A and X are rank-three in this example.

■ **Discussion of Example 2**

Note that if one uses the Frobenius inner-product (component-wise multiplication) to multiply matrices, then the singular matrices of A are orthogonal; likewise for the singular tensors of X.

If a rank-one tensor is added to X, then several special cases can be considered: if any singular tensors of the decomposition in Figure (2.9) are added, then there will be no change in the rank (unless a singular tensor is added with exactly the opposite weight as it has in the TSVD of X). Only the singular values will change, or the directions (adding a singular tensor with a negative coefficient has the effect of diminishing the singular values: if the coefficient drives the sign negative, the sign of one of the singular vectors is changed).

If any rank-one tensor that has v in the third dimension is added, then the result is still a rank-three object, with a potential perturbation in each singular matrix (rank-one, in the first and second dimensions of A). (Such an addition is equivalent to a change in the A matrix chosen at the outset.)

If, in fact, any three-tensor which is bi-orthogonal to each singular tensor of the TSVD of X is added, then a rank-four tensor results. Otherwise one must recalculate the TSVD, based on the new information contained in the additional rank-one tensor.

2.3.5 Maximum Rank of a Three-Tensor

Now consider the case of a three-tensor X of dimension $p \times q \times r$, with $p \leq q \leq r$. In this case X can be written in the form of a sum of $p \times q \times r$ outer-products of vectors as follows:

$$X = \sum_{i=1}^{p} \sum_{j=1}^{q} \sum_{k=1}^{r} x_{ijk} \underline{e}_i^X \otimes \underline{e}_j^Y \otimes \underline{e}_k^Z$$

where the "X" index on the unit vector $\underline{e}_i^X$ indicates that it is the Euclidean basis vector in the X-dimension of the space with the i^{th} component equal to 1, and all others equal to 0.

Writing this in a slightly different way shows the actual maximum rank of X:

$$X = \sum_{i=1}^{p} \underline{e}_i^X \otimes \sum_{j=1}^{q} \underline{e}_j^Y \otimes \sum_{k=1}^{r} x_{ijk} \underline{e}_k^Z;$$

but the third sum is a single vector in the third space, and hence of rank one: therefore,

$$X = \sum_{i=1}^{p} \sum_{j=1}^{q} \underline{e}_i^X \otimes \underline{e}_j^Y \otimes \underline{v}_{ij}^Z \tag{2.3.9}$$

where the indices on $\underline{v}$ are not components, but rather different vectors living in the third (Z) dimension, subscripted by ij. Thus, X is the sum of pq orthogonal

outer-products of vectors, and is hence of rank at most pq. In fact,

$$X = \sum_{i=1}^{pq} \lambda^i \underline{r}^i \otimes \underline{s}^i \otimes \underline{t}^i \equiv \sum_{i=1}^{pq} \lambda^i X^i,$$

where $\lambda^i \geq 0$ and the vectors $\underline{r}, \underline{s}, \text{and} \underline{\text{t}}$ are unit vectors in their respective spaces, and the X^i are mutually orthogonal rank-one tensors. They are mutually orthogonal as the Frobenius inner-product of two of them is

$$\langle \underline{e}_i^X \otimes \underline{e}_j^Y \otimes \underline{v}_{ij}^Z, \underline{e}_{i'}^X \otimes \underline{e}_{j'}^Y \otimes \underline{v}_{i'j'}^Z \rangle = \delta_{ii'}\delta_{jj'} \langle \underline{v}_{ij}^Z, \underline{v}_{i'j'}^Z \rangle,$$

and they are distinct iff $\delta_{ii'}\delta_{jj'} = 0$

Now a question posed in the matrix case naturally poses itself again: can one find a "best" set of such outer-products, such that

- the outer-products are mutually orthogonal; and
- each successive outer-product $X^i, i = 1, \cdots, pq$ yields the maximal coefficient λ^i over all other outer-products of the reduced tensor $X - \sum_{k=1}^{i-1} X_k$?

One natural approach would be to attempt to turn this problem into a matrix problem and perform the usual SVD on the matrix. The following example will help show why it is not enough to group the vectors of a single dimension, perform matrix SVD on them, and then use the singular vectors of that dimension.

■ Example 3

Consider a case where X is "decomposed" as in equation (2.3.9); take the vectors $\underline{v}_{ij}^Z$, form a matrix (call it $M_{r \times pq}$), and perform the SVD on that matrix to obtain the decomposition of $M = U\Lambda V^T$, where U is $r \times r$. Thus,

$$X = \sum_{i=1}^{p} \sum_{j=1}^{q} \underline{e}_i^X \otimes \underline{e}_j^Y \otimes \underline{v}_{ij}^Z = \sum_{i=1}^{p} \sum_{j=1}^{q} \underline{e}_i^X \otimes \underline{e}_j^Y \otimes \sum_{k=1}^{pq} \alpha_k^{ij} \underline{u}^k.$$

If X is now multiplied along its Z-columns by U^T, whose first pq rows are the vectors $\underline{u}^k$, then the result is a three-tensor whose first pq XY-planes are filled with the components of the singular value decomposition (that is, with the elements of the columns of $\alpha = V\Lambda$ (of length pq), but arranged into a matrix, and whose last $r - pq$ XY-planes contain all zeros. I.e., computing the inner-product on the Z dimension (resulting in a product $r \times r$ and $r \times p \times q$ on the adjacent r dimensions),

$$X'_{ijk} = \langle U, X \rangle_{ijk} = \alpha_k^{ij}$$

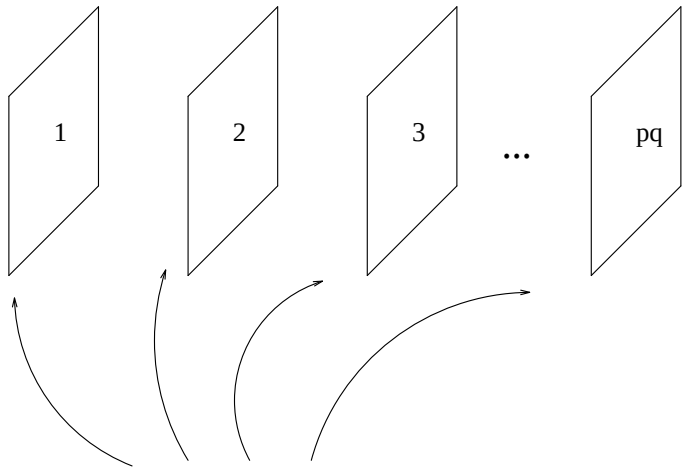


FIGURE 2.10. The shortened (non-zero portion of the) three-tensor X after multiplication by the orthogonal matrix of singular vectors in the long dimension.

and

$$X'_{ijk} = 0,\ k > pq.$$

(Notice that this shows that the general problem of tensor decomposition reduces to tensors of dimension $p \times q \times pq$, since an orthogonal matrix multiplication leaves a tensor with all zeros beyond the pq^{th} layer along the Z edge. This is important in actual computations using power methods, described in the algorithm section, as it may lead to a great decrease in the size of the tensor on which one must perform multiplications, etc.)

Thus the amount of total information contained in the first plane will be λ_1^2, and the amount in the second will be λ_2^2, etc.

One might expect that the Principal Singular Value of the tensor would be found in the first face, as it is weighted by λ_1. However, that is not generally the case, for the following reason: suppose that the information (as measured by the Frobenius norm of a matrix) of the first face is dispersed about the matrix (that is, that it is full rank, say), but that the information in the second is concentrated (of rank-one, say). Then the best outer-product may not appear in the first face, but rather possibly in the second, as is shown in Figure (2.11).

The difference between the matrix and tensor case occurs because a vector is broken into a matrix, which assigns it a rank (something which made no sense previously).

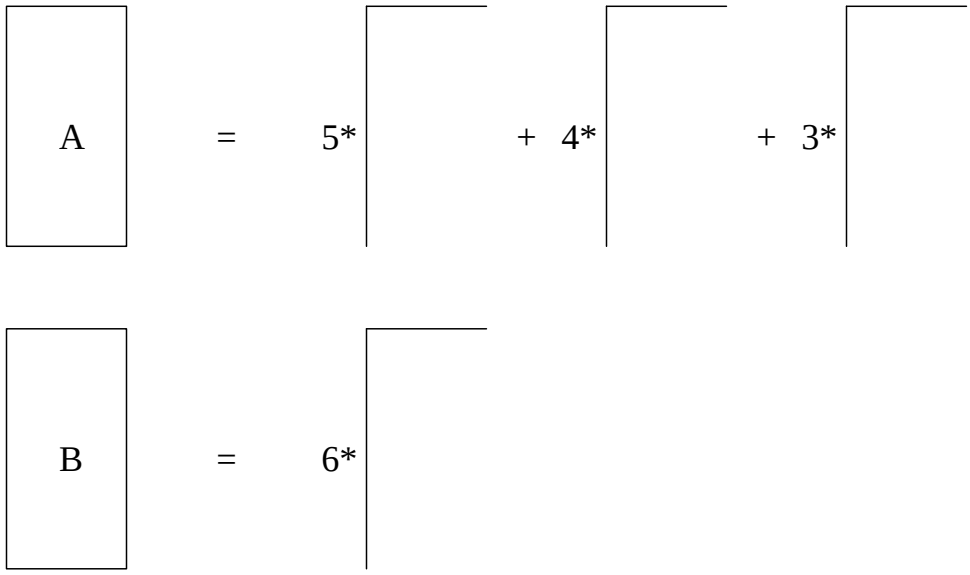


FIGURE 2.11. Face A of the tensor contains more "information" overall, but face B has the most heavily weighted single outer-product.

2.3.6 The Eigen Tao of TSVD

Begin with a three-tensor $X_{p\times q\times r}$ ("an affine tensor of rank three" [54], to be precise; however, their use of the term "rank" and ours do not agree: often, the word "rank" is used to indicate the number of dimensions; we have just described our use of the term in the preceding section). At the risk of confusion, we use the letter p to represent vectors in the p-dimension, etc. Here is the generalization to the three-tensor case (using the Eigen Tao approach): maximize the norm of the inner-product of the rank-one matrix $\underline{p}\otimes\underline{q}$, of unit vectors $\underline{p}$ and $\underline{q}$, and X:

$$\|\langle X, \underline{p}\otimes\underline{q}\rangle\| = \|\lambda\underline{u}\| = |\lambda|,$$

where $\underline{u}$ is a unit vector. Again note that X is acting as a bounded linear operator $X\colon \mathbb{R}^p\otimes\mathbb{R}^q \to \mathbb{R}^r$. Since the domain is closed, the range will be closed, so the norm over the range will take a maximum. This will be a maximum iff the following is a maximum:

$$\langle\langle X, \underline{p}\otimes\underline{q}\rangle, \langle X, \underline{p}\otimes\underline{q}\rangle\rangle = \lambda^2 \tag{2.3.10}$$

Adding the requirements that $\underline{q}$ and $\underline{p}$ be unit vectors adds two Lagrange multipliers to the optimization problem: so finally, the objective is to maximize

$$E(\underline{p},\underline{q},L_1,L_2) = \langle\langle X, \underline{p}\otimes\underline{q}\rangle, \langle X, \underline{p}\otimes\underline{q}\rangle\rangle - L_1(\underline{q}^T\underline{q}-1) - L_2(\underline{p}^T\underline{p}-1). \tag{2.3.11}$$

The left-hand side of equation (2.3.10) is

$$\sum\sum\sum\sum\sum p_i q_j x_{ijk} x_{lmk} p_l q_m.$$

Differentiating equation (2.3.11) with respect to the p_i, q_j and L gives a system of $p \times q + 2$ equations: for the i^{th} component of $\underline{p}$

$$\sum\sum\sum\sum q_j x_{ijk} x_{lmk} p_l q_m - L_2 p_i = 0, \tag{2.3.12}$$

and similarly for the q_j:

$$\sum\sum\sum\sum p_i x_{ijk} x_{lmk} p_l q_m - L_1 q_j = 0, \tag{2.3.13}$$

with the two constraints that

$$\sum p_i^2 = \sum q_j^2 = 1.$$

For solution vectors $\underline{p}$ and $\underline{q}$ (dotting through by the appropriate vector),

$$L_1 = \sum\sum\sum\sum\sum p_i q_j x_{ijk} x_{lmk} p_l q_m = L_2,$$

which means that the two Lagrange multipliers can be replaced by a single one, call it λ^2 (it is obviously positive as the norm square of the vector $\langle X, \underline{p} \otimes \underline{q} \rangle$).

The first two sets of equations, (2.3.13) and (2.3.12), written in matrix form are essentially generalized eigenvalue problems for the $\underline{q}$ and $\underline{p}$:

$$Q(\underline{p})\underline{q} = \lambda^2 \underline{q}$$

and

$$P(\underline{q})\underline{p} = \lambda^2 \underline{p},$$

or

$$\begin{bmatrix} Q(\underline{p}) & 0 \\ 0 & P(\underline{q}) \end{bmatrix} \begin{bmatrix} \underline{q} \\ \underline{p} \end{bmatrix} = \lambda^2 \begin{bmatrix} \underline{q} \\ \underline{p} \end{bmatrix}.$$

These matrix equations comprise a generalized eigenvalue problem for the four-tensor $X^T X$:

$$\langle \langle X_{ijk}, X_{lmk} \rangle, \underline{p} \otimes \underline{q} \rangle = \lambda^2 \underline{p} \otimes \underline{q} \tag{2.3.14}$$

with the last equation again requiring that the solution be a "unit object": a unit matrix (two-tensor) of rank one.

Now the question becomes: does it also have a full set of orthogonal solutions? If so, then we will have found a Tensor Singular Value Decomposition, just as in the SVD case (or *vice versa*).

2.4 Proof of the Existence of the Tensor SVD in Three Dimensions

The proof of the existence of the first singular tensor is simple, and is valid in arbitrary tensor dimensions. Begin with an $n-$tensor T, of dimensions $d_1 \times d_2 \times \cdots \times d_n$.

We use T to define a natural linear functional, representable as the inner-product of T and all other elements of the tensor space. The inner product used is the Frobenius inner-product on the linear tensor space, which is also sometimes called the Euclidean inner-product:

$$\langle T, S \rangle = T_{ijk} S_{ijk},$$

where S is an arbitrary tensor of the same size as T, and where we have used the Einstein Summation Convention: sum over repeated indices.

Identify T with this functional, since the vector space and the space of such linear functionals are isomorphic, and consider T as acting on a special subset of the vector space, the set of rank-one tensors D:

$$T \colon D \to \mathbb{R},$$

where $D = S^{d_1} \otimes S^{d_2} \otimes \cdots \otimes S^{d_n}$, and S^{d_i} is the unit sphere in d_i-space, with

$$\langle T, \underline{v}^1 \otimes \underline{v}^2 \otimes \cdots \otimes \underline{v}^n \rangle = T_{i_1 \cdots i_n} v^1_{i_1} \cdots v^n_{i_n} \in \mathbb{R}. \tag{2.4.15}$$

The domain D is a subset of the linear tensor space, but does not constitute a subspace, as it is not closed under linear combinations. One can, however, form a basis for the space from elements of D. This means that the effect of T (as a functional) on any element of the space can be calculated by knowing its effect on D.

D is closed and bounded as a set (and hence compact), and the operator is continuous and hence bounded, so it must attain a maximum on the domain. Corresponding to this maximum is an element of D, a rank-one tensor which will be called a **principal singular tensor** $T_1 \equiv \underline{v}_1 \otimes \underline{v}_2 \otimes \cdots \otimes \underline{v}_n$. The maximum value of the functional is the **principal singular value** λ_1. Note that $\lambda_1 \geq 0$, as T is a linear functional, i.e.

$$\langle T, \underline{v}^1 \otimes \underline{v}^2 \otimes \cdots \otimes (-\underline{v}^j) \otimes \cdots \otimes \underline{v}^n) \rangle = -\langle T, \underline{v}^1 \otimes \underline{v}^2 \otimes \cdots \otimes \underline{v}^j \otimes \cdots \otimes \underline{v}^n) \rangle.$$

Having established the existence of the first, we now show that T can be successively decomposed until at last there are no more singular tensors. Although the following theorem is only stated in the three-tensor case, it is conjectured that the TSVD can be extended to arbitrary tensor dimensions.

Theorem 2.4.1. *Any three-tensor $T_{p \times q \times r}$, $p \leq q \leq r$, can be written as a sum of at most pq mutually bi-orthogonal rank-one three-tensors such that*

$$T = \sum_{i=1}^{pq} \lambda_i T_i \equiv \sum_{i=1}^{pq} \lambda_i p_i \otimes q_i \otimes r_i, \tag{2.4.16}$$

with coefficients $\lambda_i \geq 0$, satisfying

$$\|T - \sum_{i=1}^{k} \lambda_i T_i\|^2 = \|T\|^2 - \sum_{i=1}^{k} \lambda_i^2 = \sum_{i=k+1}^{pq} \lambda_i^2 \leq \|T - \sum_{i=1}^{k} a_i R_i\|^2 \tag{2.4.17}$$

$\forall k \leq pq, \forall \{a_i\}_{i \in \{1,\cdots,k\}} \subset \mathbb{R}$ and $\{R_i\}_{i \subset \{1,\cdots,k\}} \in S^p \otimes S^q \otimes S^r$, with S^d the unit sphere in d-space.

Proof: Consider a three-tensor, T, on a space of dimensions $p \times q \times r$, where, WLOG, $p \leq q \leq r$. The existence of the first singular tensor $T_1 \equiv p_1 \otimes q_1 \otimes r_1$ has already been established.

We begin by proving bi-orthogonality. Having found the p_1 and q_1 vectors of T_1, the r_1 vector is determined, by the product $\langle T, p_1 \otimes q_1 \rangle = \lambda_1 r_1$. We now show that, in fact,

$$T \perp \left(p_1^\perp \otimes q_1 \otimes r_1\right) \cup \left(p_1 \otimes q_1^\perp \otimes r_1\right) \cup \left(p_1 \otimes q_1 \otimes r_1^\perp\right); \tag{2.4.18}$$

for, if not, then (examining, WLOG, the r direction) $\exists u \in r^\perp$ (a unit vector) such that

$$T(p_1 \otimes q_1 \otimes u) = \beta_1 \neq 0.$$

Therefore one could increase the principal singular value of λ_1 by choosing, instead of r_1, the unit vector

$$r_1' = \frac{\lambda_1 r_1 + \beta_1 u}{\sqrt{\lambda_1^2 + \beta_1^2}} :$$

when one take the inner-product of that tensor with T, the result is

$$\langle T, p_1 \otimes q_1 \otimes r_1' \rangle = \frac{\lambda_1^2 + \beta_1^2}{\sqrt{\lambda_1^2 + \beta_1^2}} = \sqrt{\lambda_1^2 + \beta_1^2} > \lambda_1,$$

which is a contradiction: $p_1 \otimes q_1 \otimes r_1$ maximized the functional T. Therefore, succeeding singular tensors are at least bi-orthogonal to the first.

Thus the search for the next singular tensor can be restricted to the domain D_1, defined as the set of outer-products bi-orthogonal to T_1:

$$\begin{aligned} D_1 &= (p_1^\perp \otimes q_1^\perp \otimes S^r) \cup (p_1^\perp \otimes S^q \otimes r_1^\perp) \cup (S^p \otimes q_1^\perp \otimes r_1^\perp) \\ &= (p_1^\perp \otimes S^q \otimes S^r) \cup (S^p \otimes q_1^\perp \otimes S^r) \cup (S^p \otimes S^q \otimes r_1^\perp) \\ &\quad -p_1^\perp \otimes q_1 \otimes r_1 - p_1 \otimes q_1^\perp \otimes r_1 - p_1 \otimes q_1 \otimes r_1^\perp. \end{aligned}$$

Successive domains (intersections of compact subsets) can be written as the set of outer-products perpendicular to all previously obtained singular tensors, or

$$D_k = \bigcap_{i=1}^{k} \left((p_i^\perp \otimes q_i^\perp \otimes S^r) \cup (p_i^\perp \otimes S^q \otimes r_i^\perp) \cup (S^p \otimes q_i^\perp \otimes r_i^\perp)\right) \equiv \bigcap_{i=1}^{k} O_i,$$

where O_i denotes the subset of D bi-orthogonal to the i^{th} singular tensor T_i.

As each successive D_k is compact, a new singular tensor of T is obtained until all dimensions of the Hilbert space have been exhausted. Thus, in the end, there is a set of bi-orthogonal tensors. There can be at most pq bi-orthogonal singular tensors, as is easily seen: since there are at most p independent vectors in the x direction, and at most q independent vectors in the y direction, there can be at most pq such bi-orthogonal singular tensors.

Now the question of satisfying the equality of the theorem arises: is it true that

$$T = \sum_{i=1}^{pq} \lambda_i T_i,$$

and does the difference satisfy the minimization constraint of the theorem?

The equality will be shown making use of the following

Lemma 2.4.1.

$$T - \sum_{i=1}^{k} \lambda_i T_i \perp D - D_k.$$

Proof: Note first that the actions of T and $T - \lambda_1 T_1$ on D_1 are the same:

$$\langle T - \lambda_1 T_1, D_1 \rangle = \langle T, D_1 \rangle - \langle \lambda_1 T_1, D_1 \rangle = \langle T, D_1 \rangle,$$

as D_1 was defined to be orthogonal (bi-orthogonal) to T_1. Furthermore $T - \lambda_1 T_1$ is orthogonal to $D - D_1$ (the set of elements of D <u>not</u> bi-orthogonal to T_1), as T is orthogonal to all things which are merely orthogonal to T_1 (2.4.18), and $T - \lambda_1 T_1 \perp T_1$:

$$\langle T - \lambda_1 T_1, T_1 \rangle = \langle T, T_1 \rangle - \lambda_1 \langle T_1, T_1 \rangle = \lambda_1 - \lambda_1 = 0.$$

Therefore, as any element of $D - D_1$ can be written as a sum of elements of the first and second types, the anchoring case is established:

$$T - \lambda_1 T_1 \perp (D - D_1).$$

This means that the tensor T has been split into two parts:

$$T = (T - \lambda_1 T_1) + \lambda_1 T_1,$$

with

$$T - \lambda_1 T_1 \perp D - D_1 \text{and} \lambda_1 \mathrm{T}_1 \perp \mathrm{D}_1,$$

and that decomposing T on D_1 will be the same as decomposing $T - \lambda_1 T_1$ on D_1.

Next comes induction: suppose that

$$T - \sum_{i=1}^{j} \lambda_i T_i \perp D - D_j, \forall j \leq k.$$

Consider the action of $T - \sum_{i=1}^{k} \lambda_i T_i$ on D_k; it determines a singular tensor T_{k+1}. Construct D_{k+1}, and consider the three-tensor $T - \sum_{i=1}^{k+1} \lambda_i T_i$. Now

$$(T - \sum_{i=1}^{k} \lambda_i T_i) - \lambda_{k+1} T_{k+1} \perp D_k - D_{k+1} = (D - O_{k+1}) \cap D_k,$$

by an argument equivalent to the one given in the anchor case: $T - \sum_{i=1}^{k} \lambda_i T_i \perp (D - O_{k+1})$, as is T_{k+1}, and their effect on D_k is equivalent. But this implies that

$$T - \sum_{i=1}^{k} \lambda_i T_i - \lambda_{k+1} T_{k+1} \perp D - D_{k+1},$$

as one can see by considering

$$\langle T - \sum_{i=1}^{k} \lambda_i T_i - \lambda_{k+1} T_{k+1}, D - (D - D_k) - D_{k+1} \rangle.$$

It has already been seen that

$$\langle T - \sum_{i=1}^{k} \lambda_i T_i, D - D_k \rangle = 0,$$

and T_{k+1} is orthogonal to $D - D_k$ as it is in D_{k+1}. Therefore,

$$\langle T - \sum_{i=1}^{k+1} \lambda_i T_i, D_k - D_{k+1} \rangle = \langle T - \sum_{i=1}^{k+1} \lambda_i T_i, D - D_{k+1} \rangle = 0,$$

which proves the lemma.

What this means, of course, is that when one gets to the last bi-orthogonal tensor, T_{pq},

$$\langle T - \sum_{i=1}^{pq} \lambda_i T_i, D - D_{pq} \rangle = \langle T - \sum_{i=1}^{pq} \lambda_i T_i, D \rangle = 0;$$

as $D_{pq} \equiv \emptyset$. This says precisely that

$$T - \sum_{i=1}^{pq} \lambda_i T_i = 0,$$

as was claimed, because the null space of $T - \sum_{i=1}^{pq} \lambda_i T_i$ is the entire space of D, from which one can form a basis of the tensor space (e.g., the Euclidean basis of outer-products $e_i^p \otimes e_j^q \otimes e_k^r$).

As for the minimization constraint, maximizing $\langle T, t\rangle$ over all $t \in D$ is equivalent to minimizing

$$\langle T - \lambda t, T - \lambda t\rangle = \langle T, T\rangle - 2\lambda\langle T, t\rangle + \lambda^2$$

for λ (differentiating with respect to λ gives $\lambda = \langle T, t\rangle$). Therefore

$$\langle T - \lambda_1 T_1, T - \lambda_1 T_1\rangle = \langle T, T\rangle - \lambda_1^2$$

is a minimum for T_1, as λ_1 is a maximum over all rank-one tensors: so the minimization condition is satisfied for the first singular tensor.

Again use induction: suppose that $T - \sum_{i=1}^{j} \lambda_i T_i$ minimizes the norm $\forall j \leq k$ over D. Consider $T - \sum_{i=1}^{k+1} \lambda_i T_i$, where T_{k+1} was chosen to maximize $\lambda = \langle T - \sum_{i=1}^{k} \lambda_i T_i, t\rangle$ over all $t \in D_k$: therefore, by the reasoning above,

$$\|T - \sum_{i=1}^{k+1} \lambda_i T_i\|^2 = \|T - \sum_{i=1}^{k} \lambda_i T_i\|^2 - \lambda_{k+1}^2 = \|T\|^2 - \sum_{i=1}^{k+1} \lambda_i^2,$$

which is a minimum over all D, as $T - \sum_{i=1}^{k} \lambda_i T_i \perp D - D_k$.

We have therefore shown that there is an optimal, bi-orthogonal decomposition of a three-tensor, which, by analogy with the Singular Value Decomposition, deserves the name "Tensor Singular Value Decomposition".

2.4.1 Special Case: the Bi-Symmetric Three-Tensor

One important special case is when the three-tensor V is symmetric in two of the three dimensions. This is the case which is especially relevant from the standpoint of geostatistics, as it represents the problem of decomposing (and then modelling) the variogram matrix.

In this case, the decomposition of V takes a special form, because of the bi-orthogonality and symmetry of the decomposition of the symmetric tensor V. In order to have full bi-orthogonality, the decomposition must take the form

$$V = \sum_i \sum_j \lambda^{ij} \underline{b}_i \otimes \underline{b}_j \otimes \underline{r}^{ij}.$$

To demonstrate this, consider the following

Question: Can one assume that the most weight is on terms of the form $\underline{b}_i \otimes \underline{b}_i \otimes \underline{r}^{ii}$?
Answer: There are only two ways in which one can have singular tensors in the symmetric case:

- $\lambda_{ii} \underline{b}_i \otimes \underline{b}_i \otimes \underline{r}^{ii}$; and
- $\lambda_{ij}[\underline{u} \otimes \underline{v} + \underline{v} \otimes \underline{u}] \otimes \underline{r}^{ij}$, where unit vectors $\underline{u}$ and $\underline{v}$ are not necessarily orthogonal.

Note that the second type is really a pair of the first type, as

$$\underline{u} \otimes \underline{v} + \underline{v} \otimes \underline{u} = \underline{b}_i \otimes \underline{b}_i - \underline{b}_j \otimes \underline{b}_j \text{where} \mathrm{b_i} \equiv \frac{1}{\sqrt{2}}(\underline{\mathrm{u}} + \underline{\mathrm{v}}) \text{and} \mathrm{b_j} \equiv \frac{1}{\sqrt{2}}(\underline{\mathrm{u}} - \underline{\mathrm{v}});$$

note also that b_i and b_j are not necessarily unit vectors, but are mutually orthogonal. Thus it might be better to write

$$\lambda_{ij}[\underline{u} \otimes \underline{v} + \underline{v} \otimes \underline{u}] \otimes \underline{r}^{ij} = \lambda_{ij}[\|\underline{b_i}\|^2 \underline{\beta}_i \otimes \underline{\beta}_i - \|\underline{b_j}\|^2 \underline{\beta}_j \otimes \underline{\beta}_j] \otimes \underline{r}^{ij}$$

and consider the orthogonal unit vectors $\underline{\beta} = \frac{\underline{b}}{\|\underline{b}\|}$ instead. So if the most weight were on terms of the second type, then one could simply rewrite them as two terms of the first type. Thus the answer is "yes" to the question. In fact, it is clear that one can reduce the search for successive singular tensors to those of the symmetric form $\underline{b}_i \otimes \underline{b}_i \otimes \underline{r}^{ii}$.

2.5 TSVD: Rapid Interpolator in Higher-Dimensions

The TSVD, as a generalization of the SVD, is now a rapid interpolator in the higher-dimensional case, just as the SVD is one in two-space. The idea is exactly the same, if one keeps one's attention separately on the singular vectors in each direction. One interpolates the singular vectors, and reconstructs as much of the original data as one considers signal, leaving out the noise. One now has a method for quickly producing a functional interpolator/estimator of a high-dimensional data set defined on a grid.

Applications include data transmission, where one could reduce the bandwidth of images by sending first the singular images, and then only the components of the largest singular tensors needed to achieve a chosen level of reproduction; and in the medical field, Stytz and Parrott [88] note that "the three estimation methods typically used for interpolation [of 3D medical images] are nearest neighbor, linear interpolation, and trilinear interpolation", and then apply kriging to the interpolation problem. The TSVD offers another alternative.

2.6 A Solution Algorithm

One of the most obvious ways of obtaining the singular tensors is by a power method. One can iterate the non-linear matrix equations (2.3.14), starting from a random rank-one outer-product, and hope to converge on the principal singular tensor. Then, using deflation (by removing the component of this tensor), one moves on down through all the singular tensors of the tensor X.

There are problems with this procedure even in the matrix case, so it is only of limited utility. Still, it often works, and has allowed us to begin experimenting with decompositions. It is also the only algorithm we currently have! The main problem

with deflation is that components of once-removed tensors can be reintroduced in the succeeding steps of the process, through round-off errors. The less deflation an algorithm requires, the better.

Here is the algorithm used in the Matlab program *sympower.m*, given in the appendix, for symmetric three tensors:

- A) Read in the symmetric three tensor $T_{m\times m\times n}$;
- B) Scale the tensor to unit Frobenius norm: $T = \frac{T}{\|T\|}$ (This doesn't change the singular tensors, only the singular values; correct for the scaling at the end.)
- C) for i = 1:m
  - 1) Randomize the first unit vector $\underline{q}_i$ (hoping it has a component in the direction of the principal singular vector $\underline{q}_1$);
  - 2) Compute the four tensor $A_{m\times m\times m\times m} = T^T T$;
  - 3) for j = 1, maximum iterations
    - a) Compute the new estimate:

$$\underline{q}_i^* = \langle A, \underline{q}_i \otimes \underline{q}_i \otimes \underline{q}_i \otimes \underline{q}_i \rangle$$

    - b) check for improvement:

$$\underline{q^*}_i = \frac{\underline{q}_i^*}{\|\underline{q}_i^*\|}$$

$$\underline{qdiff} = \underline{q}_i - \underline{q}_i^*$$

$$\underline{q}_i = \underline{q}_i^*$$

      if $\|\underline{qdiff}\| << \epsilon$, goto 4).
    - c) Choose one of two techniques: either
      - orthogonalize $\underline{q}_i$ against the previously obtained $\underline{q}_k$, $1 \le k < i$, and iterate, or
      - Proceed without orthogonalizing. In this case, symmetric pairs will be found in pairs according to the rule

$$b_i b_j^T + b_j b_i^T = c_i c_i^T - c_j c_j^T,$$

        where

$$c_i \equiv \frac{b_i + b_j}{\sqrt{2}} \text{and} c_j \equiv \frac{b_i - b_j}{\sqrt{2}}.$$

        Both methods converge to the zero tensor. We have found, however, that the z-vectors are slightly different if we let the second method run its course: therefore, it is wise to introduce a routine to force them to be the same.

– 4) Compute $\underline{r}_{ii} = \langle T, \underline{q}_i \otimes \underline{q}_i \rangle$, $\lambda_i = \|\underline{r}_{ii}\|$, and $\underline{r}_{ii} = \frac{\underline{r}_{ii}}{\lambda_i}$, and set

$$T = T - \lambda_i \underline{q}_i \otimes \underline{q}_i \otimes \underline{r}_{jj}$$

- D) Compute all cross outer-products, and the third dimension vectors $\underline{r}_{jk}$ and singular values λ_{ij}, and rescale the λ values.

The symmetric tensor T will then be decomposed into a sum of (at most) q^2 singular tensors of the form

$$T = \sum_{i=1}^{q} \sum_{j=1}^{q} \lambda_{ij} \underline{q}_i \otimes \underline{q}_j \otimes \underline{r}_{ij},$$

such that the products $\underline{q}_i \otimes \underline{q}_j \otimes \underline{r}_{ij}$ are mutually bi-orthogonal.

A Matlab program, *unsymsort.m*, is given in the appendix, with comments, for the unsymmetric algorithm. Note, however, that in the unsymmetric case the complete rank-one portions of the orthogonal complements of the vectors already found are searched first, then a comparison is made between the singular value obtained on that space and those of matrix subspaces corresponding to projecting out each vector already in a singular tensor. If a singular value of the subspaces is nearly as large as that of the singular tensor in the completely orthogonal space, one may take it instead. This may help avoid the problems of straying into already eliminated components of the space due to round-off.

Of course, *unsymsort.m* may be used for the symmetric case as well.

Chapter 3

Variogram Analysis

3.1 Introduction

Variogram Analysis is a multivariate method designed to determine the relationship between the covariance structure of the variables in a study, and the spatial aspects of the problem. As a multivariate method, variogram analysis thus results in the study of auto-correlations and cross-correlations. It is a non-standard type of multivariate analysis, in the sense that it is not yet well formalized, and too little-used: but for phenomena exhibiting spatial correlation (images, for example, or meteorological problems), it is extremely useful and important.

Variogram analysis is essentially a generalization of Principal Components Analysis (PCA), which looks at the correlations that exist between variables at the same site, and between sites on the same variable. Variogram analysis goes beyond PCA, however, to consider the correlations that exist between different variables at different sites. This type of analysis allows one to identify directional trends in the data, and the coherence distance (defined as the distance over which two variables are correlated) of the variables. One can sum up variogram analysis by saying that, through it, one attempts to establish the relationship of neighboring sites as a function of position for all pairs of variables. Often this is a precursor to the estimation of values at a given site using information from only its neighbors, which are weighted according to the relationships uncovered in variogram analysis.

Geostatistical methods are grounded in the belief that the phenomena of interest are stationary to some extent. Stationarity will now be made precise. Consider a single variable, perhaps nitrate, sampled in three-space. Consider the data to be a non-random sample from one realization of a random function. That is, at each point in space Ω there is a random variable (e.g. potential nitrate concentration values at that position), and the collection of these random variables forms a random function. There is at each point a realized value, the collection of which compose the realization. Assume that data values are known for N locations, the realized values of the random variables at those sites.

Now consider the various forms of stationarity. The random function Z has a marginal at each point in space. Strong stationarity means that the marginals are all the same, and more generally that

Definition 3.1.1. *$Z(x)$ is* **stationary** *if, for any h and for any finite number N of points $x_1, \ldots, x_N$, the joint distribution of $Z(x_1), \ldots, Z(x_N)$ is the same as the joint distribution of $Z(x_1 + h), \ldots, Z(x_N + h)$. (See [15], p. 273-276.)*

There are various weaker forms of stationarity (see [68]), among which are:

- Second-order stationarity:

  Definition 3.1.2. *Z(x) is* **second-order stationary** *if cov[Z(x + h), Z(x)] exists and depends only on h.*

  Note that stationarity does not imply second-order stationarity: this is not an inherited property. This form of stationarity implies that Var[$Z(x)$] and E[$Z(x)$] exist and do not depend on x.

- The Intrinsic Hypothesis, which means that the differences of variable Z are second-order stationary:

  Definition 3.1.3. *Z(x) satisfies the* **intrinsic hypothesis** *if* $E[Z(x+h) - Z(x)] = 0 \ \ \forall x$ *and h; and* $\gamma(h) = \frac{1}{2}\mathrm{Var}[\mathrm{Z(x+h)} - \mathrm{Z(x)}]$ *exists and depends only on h.*

- and other forms, such as weak stationarity with drift, and the intrinsic hypothesis of order k (succeedingly weaker stationarity, up to higher differences classes than variograms, for instance).

In many instances, it is essential to specify exactly which form of stationarity is required.

3.1.1 Principal Components Analysis and Similar Techniques

Since variogram analysis is a generalization of PCA, it is appropriate to begin with a description of that technique. Let X be a data matrix, of N sample locations, each with p measurements which represent different variables. Principal Components Analysis is a technique used to study the variables and cases of a matrix by studying the matrix decomposition (or the decomposition of a related matrix) by SVD.

Begin with a transformation of X, computing the means and standard deviations of the variables (usually the columns of the matrix), then centering the matrix so that variables have mean 0 and variance 1. Let

$$\sigma = \mathrm{diag}(\Sigma)^{\frac{1}{2}}$$

be the matrix of standard deviations of the variables; then let

$$A \equiv (I_{N\times N} - \frac{1}{N}\underline{1}_N\underline{1}_N^T)X\sigma^{-1}.$$

(The first multiplication centers the matrix X, whereas the second scales it. This is done to make the components unique: otherwise, a change of units would affect the components obtained. One consequence is that it weights the variables equally in the

decomposition. Notice, however, that this is not true of the sites: that is, that the technique is not so even-handed in the rows as it is in the columns.) The SVD then provides the decomposition

$$A = Q_1 \Lambda Q_2^T.$$

This decomposition may be useful for exploratory reasons: one may be interested in seeing how variables are correlated, and hope to get some insight by viewing the singular vectors as representative sites or variables. Just as images may be decomposed via a series of singular images, so can a data matrix be decomposed. One may find some process to associate with certain of the Schmidt Pairs.

One might use PCA as a means of "improving" the representation of the information contained in the matrix X:the matrix X, with its p variables, is exchanged for the matrix Q_1, with its p (or fewer, depending on the rank) variables, which are just linear combinations of the original p variables (linear combinations which serve to make the new variables perpendicular in p-space).

Comparisons are made horizontally and vertically in the matrix, but no comparisons are made in other directions (diagonally, say). Each site's variable u is compared with every other site's u, and variables u and v of a given site are compared; but the comparison ends there. Often (and particularly outside of geostatistics) this makes sense, as entries in different rows and columns (my cholesterol level and your pulse) are expected to be independent. But samples may represent wells, for example, in which case spatial proximity may correspond to correlation (Jack's well's nitrate level and Jill's well's sodium level). Incorporating this type of comparison means that one must look beyond the "zeroth lag": that is, comparisons between different sites using distance and angle as a guide are required. This is the task of variogram analysis.

A technique related to PCA, Correspondence Analysis (CA) [6], proceeds in a similar way: the following matrix operations are performed prior to decomposition: given a matrix X with positive entries, one first sums up all entries and divide to get what looks like a frequency matrix, F: defining

$$\hat{x} \equiv \sum_{i=1}^{N} \sum_{j=1}^{p} x(i,j),$$

$$F \equiv \frac{1}{\hat{x}} X.$$

Compute row and column sum vectors, $\underline{f}_N$ and $\underline{f}_p$ by

$$\underline{f}_N = F \underline{1}_p,$$

and

$$\underline{f}_p = F^T \underline{1}_N,$$

and create a pair of diagonal matrices D_N and D_p by setting D_N =diag($\underline{f}_N$) and D_p =diag($\underline{f}_p$). Then the matrix A, defined below, is decomposed, using the SVD:

$$A = D_N^{-\frac{1}{2}}(F - \underline{f}_N \underline{f}_p^T) D_p^{-\frac{1}{2}} = Q_1 \Lambda Q_2^T. \tag{3.1.1}$$

The Q_1 and Q_2 matrices are not quite the matrices which represent the coordinates which are often plotted in CA analysis: they are, however, up to a final matrix multiplication. The matrix generally studied and plotted is

$$P = \Lambda^{-1} Q_2^T D_p^{-\frac{1}{2}},$$

but this is essentially a set of scaled singular vectors.

The components can be treated as a variation from independence (which is removed in the subtraction of (3.1.1)). CA is even-handed in its treatment of rows and columns because of the complete symmetry in the process, in contrast to PCA.

The steps of these (and other) matrix analysis techniques can be summarized as follows:

- make the matrix transformations of interest,
- perform the SVD on the resultant matrix, and
- interpret the singular vectors and values based on the results expected by the transformations performed.

Once again the SVD proves its value in important statistical, and geostatistical, techniques.

3.1.2 What is the Variogram, and Why Use It?

When estimating at an unsampled location one uses information obtained from neighboring locations (neighbors), which must be weighted according to some scheme. In a truly random field, neighbors are equally helpful (they each contribute to the sample mean surface, which might well be used as the estimate); in a spatially well-correlated one, near neighbors may prove to be more reliable estimators of the values at a site, and so they may be valued more (weighted more) than farther sites. The job of variogram analysis is to determine where a given case lies between these two extremes, and then to suggest models for the weight function.

The principal geostatistical assumption in variogram analysis is that the correlation structure of variables, which variogram analysis uncovers, is a function of distance and direction. The forthcoming analysis will generally be concerned with increments (the distance and angle between points): thus, it is often preferable to think in terms

of loss-of-correlation functions, which are called **variograms**, in the case of a single variable, and **cross-variograms** when the effect of variables on each other is considered.

Variograms are characterized by features which may include a **range**, a **nugget**, and a **sill**, as well as by evidence for what are called **drift** and **anisotropy**. In addition, they may exhibit inflection points, infinite growth, and other features which are related to **model type**.

All of these characteristics of the variogram have some relationship with the underlying phenomenon: the range quantifies the distance over which sites are correlated; the nugget may tell indicate how much noise there is in the data, or the extent to which sampling has not been carried out at the smallest distance scales; the sill relates to the variance of the variable; drift and anisotropy will be described in a section below, but roughly describe the role varying direction plays in the phenomenon of interest; and model type has in some cases been identified with certain types of spatial interaction (as a spherical model has been shown to be natural for certain Poisson processes).

The word "trend" is sometimes used in place of "drift", but this is unfortunate: trend is usually found via a least-squares procedure on the data, using a set of independent functions and the position coordinates, whereas drift is the (non-constant) mean surface structure of the random function [41]. Drift is sometimes estimated with the trend surface, which only contributes to the confusion.

The theoretical variogram matrix is defined as

$$\Gamma(\underline{h}) = \frac{1}{2} E\left[(\underline{z}(\underline{x}+\underline{h}) - \underline{z}(\underline{x}))(\underline{z}(\underline{x}+\underline{h}) - \underline{z}(\underline{x}))^T\right] \tag{3.1.2}$$

where $\underline{z}$ is a centered data vector (that is, its mean has been subtracted off), and $\underline{x}$ and $\underline{h}$ are vectors relating positions in space. Variograms lie on the diagonal, and cross-variograms off the diagonal (for variables i and j in element Γ_{ij}), and E is the expectation function. One may attempt to model this variogram matrix function after inspecting the sample variogram matrices obtained from the data.

One thing to note is that the variogram matrix function is nonnegative definite at each lag (that is, for each value of $\underline{h}$), as it is the expected value of nonnegative definite matrices (rank-one outer-products). Nonnegative definite matrices form a positive cone in the vector space of matrices of the given size.

On the other hand, Matheron [58] showed that the variogram γ of an individual variable is a conditionally negative definite function (CND), satisfying

$$-\int\int d\mu(x)\gamma(x-y)d\lambda(y) \geq 0 \text{ if } \int \mathrm{d}\mu(\mathrm{x}) = \int \mathrm{d}\lambda(\mathrm{y}) = 0 \tag{3.1.3}$$

for any non-zero measures λ, μ with finite support. It must also satisfy the limit condition

$$\lim_{|h|\longrightarrow\infty} \frac{\gamma(h)}{|h|^2} = 0.$$

(A good overview of positive definite functions can be found in Stewart [86], although he does not treat these additional forms.) Satisfying these two positive definite conditions will constitute our biggest modelling headache.

A variogram is, quite simply, a spatial decomposition of the variance. Inspection of the variogram allows one to spot distances or directions at which variance is small (that is to say, variables are well-correlated). It is natural to presume that sites located at positions for which the variance is low will be more reliable predictors of values at the site of interest. Similarly, cross-variograms are spatial decompositions of the covariance of two variables, as shown below.

3.2 The Variogram: Spatial Decomposition of Variance

We derive this decomposition as follows, starting with the sample variance S computed by the usual formula:

$$S = \frac{1}{N-1}\sum_{i=1}^{N}(\underline{z}_i - \overline{\underline{z}})(\underline{z}_i - \overline{\underline{z}})^T$$

Replacing the mean vector $\overline{\underline{z}}$ by the sum which defines it,

$$S = \frac{1}{N^2(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}(\underline{z}_i - \underline{z}_j)(\underline{z}_i - \underline{z}_k)^T.$$

Adding an appropriate form of zero,

$$S = \frac{1}{N^2(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}(\underline{z}_i - \underline{z}_j)(\underline{z}_i - \underline{z}_j + \underline{z}_j - \underline{z}_k)^T,$$

which is

$$S = \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}(\underline{z}_i - \underline{z}_j)(\underline{z}_i - \underline{z}_j)^T - \frac{1}{N^2(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}(\underline{z}_j - \underline{z}_i)(\underline{z}_j - \underline{z}_k)^T.$$

Notice that the second sum as exactly S, so finally

$$S = \frac{1}{2N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}(\underline{z}_i - \underline{z}_j)(\underline{z}_i - \underline{z}_j)^T,$$

or

$$S = \frac{1}{2N_p}\sum_{i=1}^{N}\sum_{j=i+1}^{N}(\underline{z}_i - \underline{z}_j)(\underline{z}_i - \underline{z}_j)^T,$$

where N_p is the total number of distinct pairs of data positions, of which there are $\frac{N(N-1)}{2}$.

Now each pair of data locations is placed into a "lag class", determined by a vector $\underline{h}$, which means that the two points are separated by (roughly) $\underline{h}$. Define N_c classes, each described by a set $P_{\underline{h}}$ of pairs of indices, such that

$$(i,j)\epsilon P_{\underline{h}} \iff \underline{d}(\underline{x}_i, \underline{x}_j) \approx \underline{h}$$

where $\underline{d}(a,b)$ gives the difference vector between data locations.

The moment estimator of the variogram function for lag $\underline{h}$ is

$$\Gamma^*(\underline{h}) = \frac{1}{2N_{\underline{h}}} \sum_{(i,j)\epsilon P_{\underline{h}}}^{N_{\underline{h}}} (\underline{z}_i - \underline{z}_j)(\underline{z}_i - \underline{z}_j)^T, \tag{3.2.4}$$

where $N_{\underline{h}}$ is the number of distinct pairs of data values, placed in the set $P_{\underline{h}}$ (pairs displaced by the vector $\underline{h}$) [1].

Thus the sample variance, S, can be written as a weighted sum

$$S = \sum_{c=1}^{N_c} \frac{N_{\underline{h}}}{N_p} \left[\frac{1}{2N_{\underline{h}}} \sum_{(i,j)\epsilon P_{\underline{h}}}^{N_{\underline{h}}} (\underline{z}_i - \underline{z}_j)(\underline{z}_i - \underline{z}_j)^T \right],$$

or

$$S = \sum_{c=1}^{N_c} \Gamma(\underline{h}) \left(\frac{N_{\underline{h}}}{N_p} \right). \tag{3.2.5}$$

Or, even more generally,

$$S = \int_V \Gamma(\underline{h}) d\mu(\underline{h}),$$

where $\mu(\underline{h})$ is a measure which represents the "number" of distinct pairs of a certain lag class. In the case of the finite measure,

$$d\mu(\underline{h}) = \frac{N_{\underline{h}}}{N_p}.$$

To demonstrate this explicitly, consider a realization of a continuous random function defined on a continuous interval (which can be taken as $[0,1]$, WLOG). The experimental variogram $s^2(h)$ is defined by

$$s^2(h) = \frac{\int_0^{1-h} (z(x+h) - z(x))^2 dx}{2\int_0^{1-h} dx},$$

[1] In reality this is only approximately true, of course: one generally groups all pairs to get a finite, relatively small number of lag classes.

or

$$s^2(h) = \frac{1}{2(1-h)} \int_0^{1-h} (z(x+h) - z(x))^2 dx.$$

The sample variance s^2 is

$$s^2 = \int_0^1 (z(x) - \int_0^1 z(h)dh)^2 dx = \int_0^1 \left(\int_0^1 (z(x) - z(h))dh \right)^2 dx.$$

Therefore

$$s^2 = \int_0^1 dx \left[\int_0^1 (z(x) - z(h))^2 dh + \int_0^1 \int_0^1 (z(x) - z(h))(z(h) - z(y))dhdy \right]$$

$$= \int_0^1 \int_0^1 (z(x) - z(h))^2 dxdh - s^2.$$

Thus,

$$s^2 = \frac{1}{2} \int_0^1 \int_0^1 (z(x) - z(y))^2 dxdy,$$

from which one concludes that s^2 is the mean of all pairs of data differences squared.

Now

$$\int_0^1 \int_0^1 f(x,y)dxdy = 2 \int_0^1 \int_0^{1-h} f(x+h,x)dxdh$$

for symmetric functions, i.e. $f(x,y) = f(y,x)$. So

$$s^2 = \int_0^1 \int_0^{1-h} (z(x+h) - z(x))^2 dxdh,$$

$$= \int_0^1 \left[\frac{1}{2(1-h)} \int_0^{1-h} (z(x+h) - z(x))^2 dx \right] 2(1-h)dh,$$

$$= \int_0^1 s^2(h)[2(1-h)dh],$$

or

$$s^2 = \int_0^1 s^2(h)\mu(dh). \tag{3.2.6}$$

This demonstration should put to rest a common error, the assertion (see Freek [89], for example) that "...the sill is equal to the variance in the data set." This is obviously incorrect for monotonically increasing variogram models, as the sill (the mean of all variogram values) would be greater than all the variogram values of the sample population! It is true that the sill of a variogram is equal to the population variance in the second-order stationary case, as Barnes [8] shows. He also discusses

the conditions under which one may properly use the sample variance in estimating the sill, but suggests rather that one might consider using an obvious sill to estimate the population variance!

If the range of a spatial phenomenon is finite, then the far variance (that is, the contribution to the variance corresponding to pairs found at large lags) has to be greater than the average variance, in order for it to compensate for the short range (low) variance. It is possible, of course, for this effect to be washed out: as the range tends to zero compared to the spatial extent of a (stationary) phenomenon, the sill will tend to the variance.

One may now put what I call the "fundamental equation of geostatistics" (either of equation (3.2.5) or equation (3.2.6)) to work in the following way: given a problem, for which a model has been chosen, compare the computed sample variance with the sample variance we obtain from the model, using the appropriate fundamental equation. Or, from [8],

$$E(s^2) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\gamma(x_i, x_j).$$

Noting that the fundamental equations are linear in the sill of the variogram model, one might, then, adjust the sill of the model so that the two match up; or one could match the model sample variance to some other estimator of the population variance (as the model is presumed valid for the whole population). This is an example of a reasonable constraint that one could introduce easily into a model.

Returning to the estimator of the variogram matrix function (Equation (3.2.4)) for a moment, each pair has a place in some lag; in fact, the whole procedure could be described as exchanging the population of data for the (much larger) population of all differences of pairs, each associated with a position vector $\underline{h}$. One then asks to what extent is the variance of the population related to position. The variogram estimator can be written in this sense as

$$\Gamma^*(\underline{h}) = \frac{1}{2N_{\underline{h}}} D^T(\underline{h}) D(\underline{h}), \tag{3.2.7}$$

where $D(\underline{h})$ represents all the differences falling (approximately) into the class given by $\underline{h}$. This representation will prove handy later on, when we discuss results of the chapter on interpolation.

The use of the sample variogram has been questioned and criticized, especially by Cressie ([19, 36]), but by others as well; this is because of its sensitivity to "outliers" (that is, extreme-valued data, which may be the result of errors which are then propagated heavily through the analysis). This is due to the squared nature of the variogram. Cressie and others, such as Journel ([47]), have proposed using exponents other than 2 in the spatial correlation function (e.g. the rodogram, with exponent .5, or the madogram with exponent 1). Obviously the variogram is serving as some

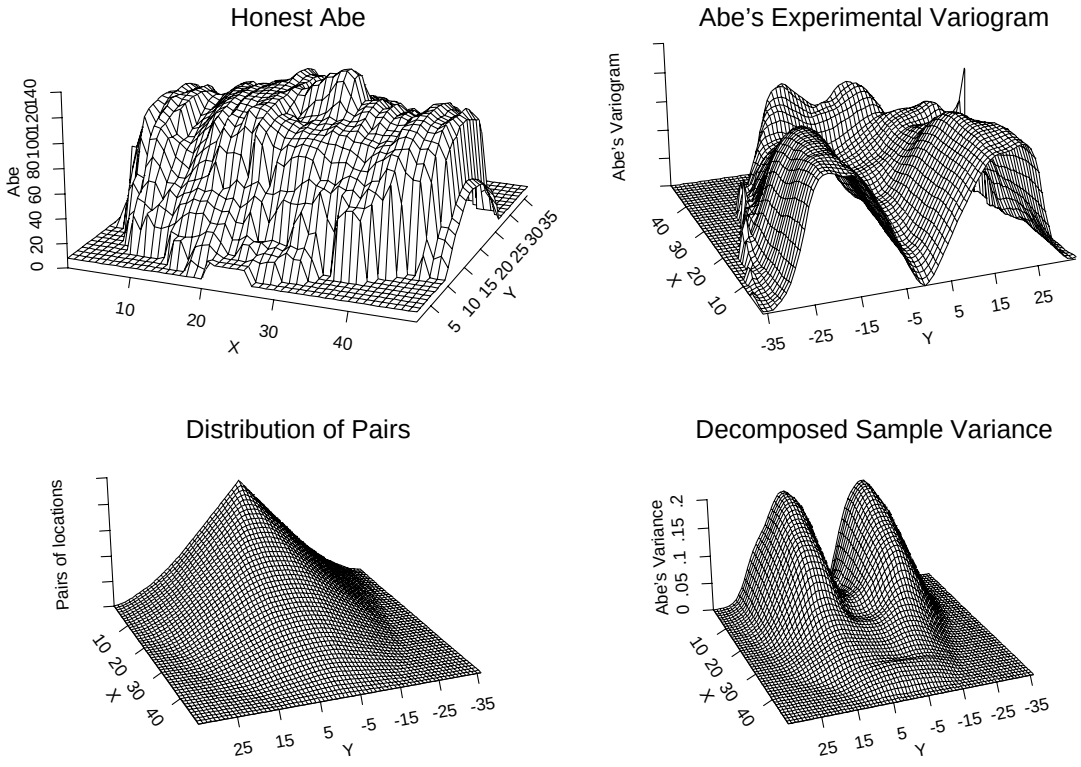


FIGURE 3.1. Abe Lincoln has his sample variance decomposed by the variogram. The variogram, weighted by the measure of the distribution of pairs, gives the variance. The panel at bottom-right represents the integrand, the product of the variogram and the measure.

sort of "metric", by which to weight pairs at different distances, and the choice of the metric will play an important role in the estimation problems to be described later on.

In spite of the non-robustness of the variogram, its intuitive definition, relationship to the variance, etc. make it the obvious choice for most problems, especially where data quality is high.

3.3 Variogram Analysis as a Multivariate Analysis Tool

Journel [46] (p. 126) notes that "Any serious practitioner of geostatistics would expect to spend a good half of his or her time looking at all faces of a data set, relating them to various geological interpretations, prior to kriging; he or she may even decide not to do any kriging!"

This remark illustrates the importance of variogram analysis beyond its role in the kriging equations. One must examine the variograms and cross-variograms for features which one can identify with the problem at hand:

- whether to assume isotropy or anisotropy;
- the degree of stationarity (or rather, the degree to which the distribution of values is independent of position);
- the "shape" of the spatial correlation, and whether it is indicative of known processes; and
- the relationship between variables, one to another.

The sample variograms and cross-variograms can help one to answer these and other important questions, and guide one in deciding whether to krige, cokrige, or to leave off kriging all together (as Journel suggests one may).

We differ, though, with him [45] (p. 6) when he argues that "...geostatisticians never consider experimental variograms beyond one-half of the maximum experimental inter-distance available". As just seen, the sample variogram, weighted by pair numbers per class, shows "where the variance is", and in this case it would be foolish to ignore the contributions that come from the back half of the distance classes.

His comment reflects his concern about several potential problems: there is an aliasing effect which is likely to occur if one goes beyond the halfway point. Also, the number of pairs that occur at great distance may decrease dramatically, meaning that the reliability of the sample variogram values as an estimator of the true variogram at those lags may not be good. Furthermore, if the study region is strangely shaped, then pairs at great distance may all come from the same direction, leading to overemphasis on that directions if one is inspecting isotropic variograms; this will not be a problem if one inspects variograms while including angle information.

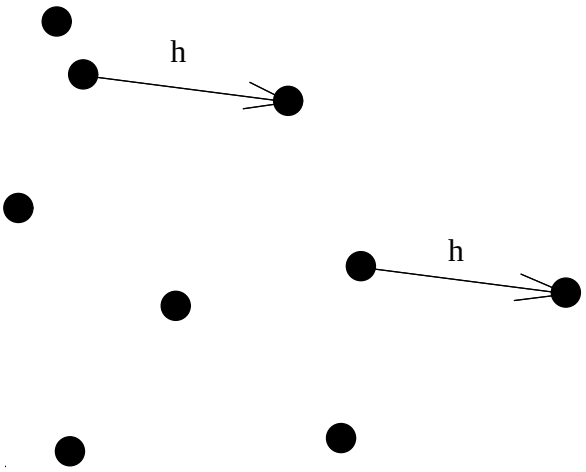


FIGURE 3.2. Data are compared at sites separated by (roughly) the same angle and distance.

In contrast to PCA, described above, variables are now compared at sites separated by a vector $\underline{h}$, representing both a distance and an angle (Figure (3.2)). If one finds that the angle is essentially irrelevant, then one may choose to assume isotropy. In the figure of Abe and his decomposed variance (Figure (3.1)), the variogram seems to have a valley along which it is small, and two (fairly symmetric) hills. There is obviously some anisotropy there, which can be understand as follows: Abe's face is much more similar along the vertical direction than it is along the horizontal direction. Although one may encounter a chin, or a nose, one doesn't see the sharp differences in the same number of pixels that one does in the horizontal direction, where one can go from a nose to an ear in a few pixels. The aliasing effect is evident in the two hills: because the variogram is drawn the full distance to the left and to the right (facing Abe), there is an approximately 70 pixel difference in the variogram image, compared to only 36 pixels of Abe in the horizontal direction. This has led to dual humps in the variogram. If the distance classes had been restricted to only 18 pixels to the left and right, the edges would have been cut out, from $y \in [0, 18]$ and $y \in [54, 70]$. (By the way, we left out the "front half" of the variogram, as it is symmetric: the angle/distance classes from $[0, 180]$-degrees are merely repeated from $[180, 360]$.)

3.4 Modelling the Variograms and Cross-Variograms

Variogram analysis is both an art and a science. In the early going of a study, one may simply be looking at variograms and cross-variograms in the hopes of discovering unanticipated structure in the phenomenon being studied. Much may be learned from even a very cursory inspection: for example, if the variogram of a variable is flat over distance, for all directions, then spatial analysis will provide no benefit over traditional

statistics. This is obviously an important insight, quickly and easily determined by simple inspection.

In the multivariate (non-univariate) case (that is, number of variables ≥ 2), one inspects the experimental variograms and cross-variograms of the variables of interest seeking similar features. If possible, one would like to deduce physical relationships (such as chemical pathways, for example) from any similarities that one might find. However, the interpretation of the sample variogram matrix is still something of an art.

Variogram and cross-variogram modelling (of the sample variogram matrix) is not easy, which may account for the fact that the techniques of kriging and cokriging have not been more widely used. Cokriging in particular is more complicated (than kriging) because it requires cross-variogram modelling.

Variogram modelling is only necessary if one is going to use the information contained in them for some further step (e.g. interpolation or simulation). If one is merely using the sample variogram as a multivariate analysis tool, then this step may be skipped entirely.

3.4.1 Variograms

We will begin with variograms, as they are essential for both kriging and cokriging, and serve as a good jumping off point to the messier problem of cross-variograms.

The constraint that the variogram be conditionally negative definite (CND) opens the door to a group of models which are commonly used to fit sample variograms. Any function which is CND can be used as a model: however, to ensure that the kriging system is invertible, it is essential that the CND function be strictly CND. The isotropic models (provided in the popular public-domain geostatistical software package Geo-EAS [24]) are:

- the nugget model:

$$\gamma(h) = \begin{cases} 0, & h = 0; \\ n, & h > 0 \end{cases} \tag{3.4.8}$$

- the linear model:

$$\gamma(h) = \begin{cases} 0, & h = 0; \\ ch, & h > 0 \end{cases} \tag{3.4.9}$$

- the gaussian model:

$$\gamma(h) = \begin{cases} 0, & h = 0; \\ c\left(1 - e^{-\ln(20)\frac{h^2}{r^2}}\right), & h > 0 \end{cases} \tag{3.4.10}$$

- the spherical model:

$$\gamma(h) = \begin{cases} 0, & h = 0; \\ c\frac{h}{2r}\left(3 - (\frac{h}{r})^2\right), & 0 < h < r; \\ c, & h > r \end{cases} \tag{3.4.11}$$

- the exponential model:

$$\gamma(h) = \begin{cases} 0, & h = 0; \\ c\left(1 - e^{-\ln(20)\frac{h}{r}}\right), & h > 0 \end{cases} \tag{3.4.12}$$

(The factor of ln(20) in the gaussian and exponential models appears because the range r of those models is defined as the point at which the model attains 95% of its sill, c.)

Since conditionally negative definite functions form a positive cone in the space of functions (which is to say that positive linear combinations of such functions are again conditionally negative definite), one strategy for modelling variograms is to use positive combinations of known valid models to fit sample variograms. No one has yet developed, to our knowledge, a more general method.

Cressie [18] describes a weighted-least squares method for determining a nested model of variograms that best-fits a sample variogram. Although Cressie's method is described in the literature, a brief description is included here. The method is designed to give more weight to close pairs, and to distances associated with many pairs. This philosophy is incorporated into the minimization function

$$C(\gamma(\underline{h};\underline{\lambda})) \equiv \sum_{j=1}^{k} N_{\underline{h}(j)} \left[\frac{\gamma^*(\underline{h}(j))}{\gamma(\underline{h}(j);\underline{\lambda})} - 1\right]^2, \tag{3.4.13}$$

where $\gamma(\underline{h}(j);\underline{\lambda})$ is the value of the model with parameters $\underline{\lambda}$ at the angle/distance given by $\underline{h}(j)$, and $\gamma^*(\underline{h}(j))$ is the sample value at the same lag; $N_{\underline{h}}$ is the number of distinct pairs at lag $\underline{h}$; and k is the number of lag classes.

The Geostatistics Group at the University of Arizona ported Geo-EAS to UNIX, and incorporated a modified version of Cressie's variogram-fitting method into the UNIX version, allowing one to take advantage of the greater power of the workstation. The UNIX version is public domain, and requests for software and assistance have come from many places, including universities and government agencies throughout the United States and Europe.

A brief description of the relevant features of the UNIX software is appropriate, as it contributed heavily to this dissertation. The software is (for the moment, anyway) restricted to two spatial dimensions.

It is not possible to do general anisotropic modelling with the software, as released. It would be necessary to add several modules to the package in order to get to that point. On the other hand, Geo-EAS permits one to inspect sample variograms along different directions, so that in principle, one could model directions separately and attempt (on one's own) to deduce geometric anisotropy information (essentially an ellipsoidal shape in the variogram of the phenomenon).

Consider, therefore, only the isotropic case:

- A pair of optimization methods is used to model the sample variogram, but they require some start-up information. Basic characteristics of variograms (as mentioned above) were used to estimate bounds for the nugget, range, and sill.
- A Monte-Carlo method is used first, testing several of the most popular models (spherical, exponential, and gaussian) to find a linear combination of these models (with variable parameter values, within the limits set out in the first step) which minimizes the Cressie function (3.4.13).
- Steepest descents follows, altering the parameters so as to achieve a minimum of (3.4.13).
- An additional "refine" procedure is included, which starts with a model (which the user may supply) and uses only the steepest descent algorithm to get to a better model. This is handy for those who want to dictate a model, because of some *a priori* information for example.

In the case of models used for the Nitrate study, additional software, which has not been released for the general public, was used in a final attempt to improve the model for the variogram: a sort of crude genetic algorithm, in which additional models were added to compare against those already in the mix, existing models were removed (as their sills approach zero relative the the other sills), and models coalesced as their parameters tended to the same values.

Results for several variograms are presented in Figure (3.3), for variables from the Nitrate study data set discussed in a later chapter. These variogram models were derived using only the automated variogram fitting of the UNIX Geo-EAS software.

■ Example: Variogram of a Weiner Process

We mentioned that some models have been shown to correspond to certain phenomena. For an explicit example, consider the model corresponding to Brownian motion (see [25], page 98) in a single dimension.

For a Weiner process, $z(x+h) - z(x) \sim N(0, \sigma)$, where $\sigma^2 = \epsilon^2(h)$ represents the variance for the h differences. The expected value of the square of this random variable is elementary: it is the variance, since the mean is zero. Thus

$$\gamma(h) = \frac{1}{2} E[(z(x+h) - z(x))^2] = \frac{1}{2}\epsilon^2(h).$$

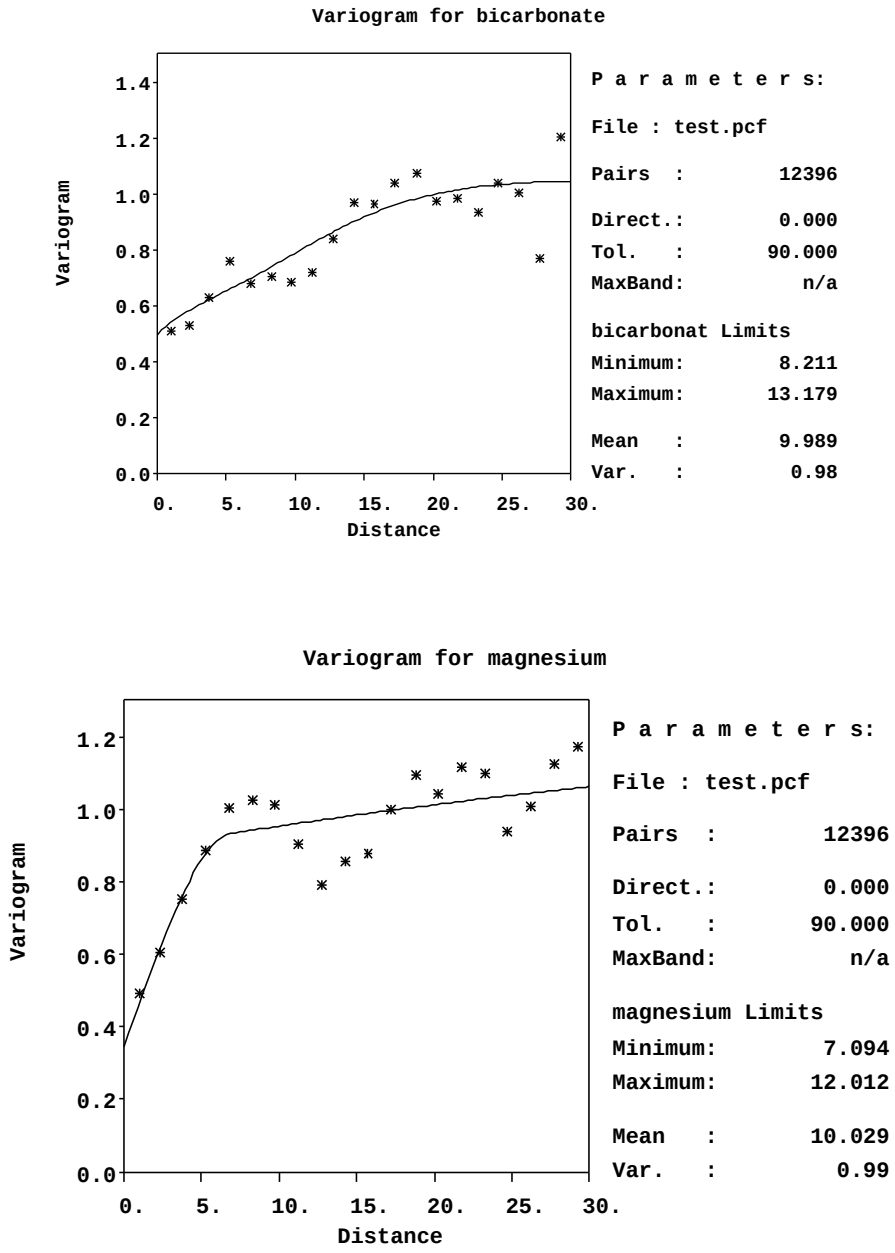


FIGURE 3.3. Model variograms (of variables from the Nitrate Study), calculated and modelled using the Geo-EAS automated technique.

The beauty of this result is that the sample variogram of a Weiner process is an estimator of the variance structure of the motion: i.e., $2\gamma^*(h)$ estimates $\epsilon^2(h)$.

For Brownian motion, $z(x+h) - z(x) \sim N(0, \sigma^2|h|)$, where $\sigma^2 = \epsilon^2(h)$ represents the variance for the h differences. Thus

$$\gamma(h) = \frac{1}{2}E[(z(x+h) - z(x))^2] = \frac{1}{2}\sigma^2|h|,$$

gives a linear variogram. Therefore, if one finds in the course of a variogram analysis that the variogram appears linear, one should ask whether Brownian motion is an appropriate model for the phenomenon under study.

3.4.2 Cross-Variograms

Cross-variogram modelling is really only necessary for cokriging, or multivariate simulations, or other purposes for which a model is an essential component. Otherwise, inspecting the cross-variograms for interesting identifiable features may be enough. One should keep in mind what is being modelled: as was shown above, the cross-variogram is a spatial decomposition of the covariance of two variables.

Myers [66] described a method whereby these functions could be modelled as linear combinations of models of variograms: i.e.,

$$\gamma_{ij}(h) = \frac{1}{2}[\gamma^+(h) - \gamma_{ii}(h) - \gamma_{jj}(h)], \tag{3.4.14}$$

which, as he later showed, could also be represented as

$$\gamma_{ij}(h) = \frac{1}{2}[\gamma_{ii}(h) + \gamma_{jj}(h) - \gamma^-(h)],$$

or, putting those two together,

$$\gamma_{ij}(h) = \frac{1}{4}[\gamma^+(h) - \gamma^-(h)].$$

γ_{ij} is the cross-variogram of variables i and j, γ^+ is the variogram of the sum of variables i and j, and γ^- is the variogram of the difference of variables i and j. This method is an important first step, at least: it permitted data analysts to get the ball rolling, as methods for modelling variograms were already in use and could be called into service for estimating cross-variograms. Some [33], however, have criticized the method, as it operates in a pairwise fashion, and does not ensure that the Cauchy-Schwartz condition,

$$|\gamma_{ij}(h)| \leq \sqrt{\gamma_{ii}(h)\gamma_{jj}(h)}, \tag{3.4.15}$$

is satisfied. As Myers noted, one must verify that separately. Furthermore, however, Goovaerts [34] gives an example in the three-variable case where the variables satisfy

FIGURE 3.4. Automated cross-variogram modelling in action. This cross-variogram was obtained by software automatically modelling the variograms of the two variables, their sum and difference, and choosing the best of the three possibilities according to Myers's scheme.

the Cauchy-Schwarz condition pair-wise, but the 3×3 variogram matrix function fails to be nonnegative definite. Thus, checking the Cauchy-Schwartz conditions for variables pair-wise does not suffice to guarantee that the variogram matrix function model is nonnegative definite at an arbitrary lag.

One of the advantages of Myers's method is that it gives three different ways to model the cross-variogram, so all three can be used and compared, to see which seems best, how much they vary, etc.

3.5 TSVD and TSVD-like Methods in Variogram Analysis

One approach to the problem of variogram matrix modelling is the method of "Coregionalization" (see [94] and [90]), in which one assumes a matrix model of the form

$$V(h) = \sum_{k=1}^{s} \gamma_k(h) V^k. \tag{3.5.16}$$

The variogram matrix function $V(h)$ is given as a sum of s products of valid variogram models (the $\gamma_k(h)$) and nonnegative definite matrices V^k (the superscript is an index, not an exponent).

The use of this model requires the hypothesis of intrinsic stationarity, and assumes that each $\gamma_k(h)V^k$ represents the correlation structure of an underlying spatial mul-

tivariate process, with the ranges of the associated variograms indicating the range of influence of each particular process [94].

As one can see, the cross-variograms are indirectly modelled as sums and differences of variogram models, as in Myers's method, but the coefficients are determined in another manner (and, in particular, a method which never actually models more than variograms):

$$V_{ij}(h) = \sum_{k=1}^{s} \gamma_k(h) V_{ij}^k.$$

Therefore, by using the coregionalization model, one has only replaced the problem of how to model cross-variograms with that of choosing the number of structures s, the nonnegative definite matrices V^k, and then modelling the variograms for each structure.

Some authors have proposed computational methods ([10] and [92], with examples), but have done so by assuming that the number and type of structures (i.e. variograms) have been identified, proceeding from there to the estimation of the corresponding V^k. We now describe ways of picking out the matrices first, leading to the modelling of the variogram matrix function and coregionalization.

Flury [26] in his book "Common Principal Components", presents the following problem: given a set of $p \times p$ correlation matrices, what is the best single matrix approximation to them all? To turn this question into a problem of TSVD-type, "What is the best rank-p symmetric (in two of the three dimensions) three-tensor approximation to the three-tensor given by the stack of matrices?" Flury uses one definition of "best", while the TSVD uses another; and while the two are different each is applicable to the same task, namely variogram analysis.

The first question to ask, then, is "best in what sense?". Flury's approach, which translated into tensor form, uses the Frobenius norm of the three-tensor of off-diagonal elements of $BTB^T - T^*$, where

$$T^* = \sum_{i=1}^{p} \underline{b}_i \otimes \underline{b}_i \otimes \langle T, \underline{b}_i \otimes \underline{b}_i \rangle,$$

and where B is an orthogonal matrix. The correlation matrices themselves are nonnegative definite (ND), which is also a relevant consideration.

This is equivalent to the problem of near-simultaneous diagonalization of matrices: Flury seeks to find an orthogonal matrix B such that the stack of matrices given in T, when multiplied on the left by B and on the right by B^T, leads to a stack of nearly-diagonal matrices: i.e.,

$$D_{mnk} \equiv \langle \langle B_{mi}, T_{ijk} \rangle, B_{nj} \rangle,$$

where the tensor D has most of its weight on its "diagonal" (components d_{mmk}).

Flury addresses the usefulness of this particular decomposition in his book, so we will not do so here. However, that is not the only use for his near-simultaneous diagonalization of matrices.

Xie [98] used Flury's method for variogram matrix modelling: he took the sequence of sample variogram matrices at 50 lags, which constitute a spatial decomposition of the sample covariance matrix, and computed the single best full-rank matrix approximation to the sequence. That is, he found that matrix A which best approximates the stack of matrices in the sequence (which he could have weighted, but did not weight, by some distance scheme, to emphasize those matrices nearest to lag zero and those having the most neighbors, and hence presumably greater validity). The matrix A has SVD

$$A = B\Lambda B^T,$$

so he defined a new sample variogram sequence

$$D_i = B^T V_i B,$$

which he found to be essentially diagonal in the sense described above.

In the following Xie's solution to the variogram modelling problem is compared to the solution of a particular TSVD problem, and we demonstrate that the two solutions are very similar.

Consider the isotropic case: start with a three-tensor of sample variogram matrices $V(h)$ (which will also be called V), where $h \in \{1, \cdots, L\}$ is the lag value, which serves as an index in the third-dimension. Since each $V(h)$ is ND,

$$V(h) = \sum_{i=1}^{p} \mu^i(h) \underline{q}^i(h) \otimes \underline{q}^i(h).$$

Xie, et al., seek to diagonalize tensor V (or some weighted version of it) in the sense described above, finding an orthogonal matrix B such that

$$\phi(B) \equiv \|BVB^T - \mathrm{diag}(BVB^T)\|^2 \tag{3.5.17}$$

is minimal (in the sense of the Frobenius norm). The tensor products BVB^T are to be understood as B and B^T acting on each layer of V. Note that if tensor V is diagonalizable, then the quantity 3.5.17 will be zero. If one considers $B = I$, then initially $\|V - \mathrm{diag(V)}\|$: the Frobenius norm of the off-diagonal elements of V.

Let $B = \sum_{i=1}^{p} \underline{b}_i \otimes \underline{b}_i$, with b_i mutually orthogonal (i.e., B is an orthogonal matrix), and seek B which minimizes the off-diagonal entries of BVB^T. Now

$$\phi(B) = \|B^T BVB^T B - B^T \mathrm{diag(BVB^T)B}\|^2,$$

since the Frobenius norm of a product of a matrix or tensor with an orthogonal matrix is unchanged:

$$\|OM\| = \|M\|, \mathrm{where} O \mathrm{isorthogonal}.$$

Hence this problem can be rephrased as finding the "largest" diagonal tensor to BVB^T: minimize

$$\phi(B) = \|V - B^T \Lambda B\|^2 = \sum_h \|V(h) - B^T \Lambda(h) B\|^2 = \sum_h \|V(h) - \sum_{i=1}^{p} \lambda_i(h) \underline{b}_i \underline{b}_i^T\|^2,$$

over all B, where Λ is the "diagonal" tensor $\mathrm{diag}(BVB^T)$.

$$\phi = \sum_h \left[\sum_{i=1}^{p} \mu_i^2(h) - 2\langle \sum_{i=1}^{p} \mu_i(h) \underline{q}_i(h) \underline{q}_i(h)^T, \sum_{j=1}^{p} \lambda_j(h) \underline{b}_j \underline{b}_j^T \rangle + \sum_{k=1}^{p} \lambda_i(h)^2 \right],$$

which means that

$$\sum_h \left[\sum_{i=1}^{p} \mu_i^2(h) - 2 \sum_{i=1}^{p} \sum_{j=1}^{p} \mu_i(h) \lambda_j(h) \langle \underline{q}_i(h), \underline{b}_j \rangle^2 + \sum_{k=1}^{p} \lambda_i(h)^2 \right].$$

To minimize this expression with respect to the values of $\lambda(h)$ (unconstrained optimization) differentiate with respect to $\lambda_j(h)$, and set the results equal to zero (to obtain a minimum):

$$-2 \sum_{i=1}^{p} \mu_i(h) \langle \underline{q}_i(h), \underline{b}_j \rangle^2 + 2\lambda_j(h) = 0.$$

This is solved, to give

$$\lambda_j(h) = \sum_{i=1}^{p} \mu_i(h) \langle \underline{q}_i(h), \underline{b}_j \rangle^2 = \langle V(h), \underline{b}_j \otimes \underline{b}_j \rangle$$

Substituting these values back into the function ϕ,

$$\phi = \sum_h \left[\sum_{i=1}^{p} \mu_i^2(h) - 2 \sum_{j=1}^{p} \langle V(h), \underline{b}_j \otimes \underline{b}_j \rangle \sum_{i=1}^{p} \mu_i(h) \langle \underline{q}_i(h), \underline{b}_j \rangle^2 + \sum_{j=1}^{p} \langle V(h), \underline{b}_j \otimes \underline{b}_j \rangle^2 \right]$$

$$= \|V\|^2 \sum_h \left[-2 \sum_{j=1}^{p} \langle V(h), \underline{b}_j \otimes \underline{b}_j \rangle^2 + \sum_{j=1}^{p} \langle V(h), \underline{b}_j \otimes \underline{b}_j \rangle^2 \right]$$

$$= \|V\|^2 - \sum_h \sum_{j=1}^{p} \langle V(h), \underline{b}_j \otimes \underline{b}_j \rangle^2$$

$$= \|V\|^2 - \sum_{j=1}^{p} \|\langle V, \underline{b}_j \otimes \underline{b}_j \rangle\|^2.$$

Thus ϕ is minimized by maximizing

$$\sum_{j=1}^{p} \|V(\underline{b}_j \otimes \underline{b}_j)\|^2.$$

This as precisely the condition of maximizing the inner-product of tensor V with respect to a set of p mutually bi-orthogonal symmetric tensors of rank-one, and the best ND matrix approximation to V will be given by those p outer-products

$$S_i = \underline{b}_i \otimes \underline{b}_i.$$

The rank-one tensors will be given by

$$T_i = \langle V, S_i \rangle \otimes S_i \equiv \underline{\gamma}_i \otimes S_i,$$

where new variogram structures $\underline{\gamma}_i$ have been defined. The norm of the approximation is given by

$$\sum_{i=1}^{p} \|T_i\| = \sum_{i=1}^{p} \|S_i \otimes \underline{\gamma}_i\| = \sum_{i=1}^{p} \sqrt{\sum_{h=1}^{L} \gamma_i^2(h)};$$

that is, by the sum of squares of the values of the new variogram structures γ_i.

This procedure has therefore led to the following coregionalization: calling the models of the diagonal elements $\gamma_1, \gamma_2, \cdots, \gamma_p$, then (using a more standard notation)

$$V(h) = \sum_{i=1}^{p} \gamma_i(h) \underline{b}_i \underline{b}_i^T \equiv \sum_{i=1}^{p} \gamma_i(h) S_i,$$

where each S_i is a nonnegative definite matrix (of rank-one). Recall that this is the definition of a coregionalization model (equation (3.5.16)).

In very similar fashion, the TSVD can be used as a means to coregionalization also. The difference is in the quantity maximized: rather than the quantity ϕ (3.5.17) of Xie, TSVD uses the quantity

$$\psi = \max \|\langle \mathrm{V}, \mathrm{b} \times \mathrm{b} \rangle\|,$$

finding b_1, then b_2 orthogonal to b_1, etc., until a basis is obtained (which gives an orthogonal matrix B).

While the two approaches are similar, they do not achieve the same thing. Xie's procedure is better "balanced" than the TSVD: it will sacrifice some weight on the first singular tensor in order to get a better second singular tensor. The TSVD method results in as much weight as possible on the first, and then as much as possible on the second, orthogonal to the first, etc.; the TSVD method does not require that the

first p singular tensors be symmetric and bi-orthogonal in the two symmetric spaces (bi-orthogonality could come about using the long dimension, instead).

Thus, if a multivariate process is rank-one (leading to a single variogram for all variables), then the TSVD would be best: if, on the other hand, the phenomenon is thought to require a set of differing variogram models (true coregionalization), then one might prefer Xie's method. However, as shown below (in a case taken from real data), it made very little difference which of several methods we used, including some spurious-seeming and unmotivated ones!

■ A Comparison of Results

Xie [98] applied his method to the variogram matrix of three variables (which, not uncoincidentally, came from the Nitrate Study which is the subject of Chapter Six). He found that the following matrix nearly simultaneously diagonalized the sample variogram tensor:

$$B = \begin{bmatrix} 0.4013 & -0.9145 & -0.0502 \\ 0.6194 & 0.3114 & -0.7207 \\ 0.6747 & 0.2581 & 0.6915 \end{bmatrix} \tag{3.5.18}$$

The matrix B gave a diagonalization efficiency, defined as

$$\text{efficiency} = \frac{\sum_{\mathrm{i}} \sum_{\mathrm{h}} (\underline{\mathrm{b}}_{\mathrm{i}}^{\mathrm{T}} \mathrm{V}(\mathrm{h}) \underline{\mathrm{b}}_{\mathrm{i}})^2}{\|\mathrm{V}\|^2},$$

of 0.99280.

Using the power method as implemented in the matlab code *sympower.m* (found in the appendix), and requiring successive orthogonality of the singular tensors, the TSVD gave the matrix

$$B = \begin{bmatrix} 0.4023 & -0.9140 & -0.0518 \\ 0.6192 & 0.3134 & -0.7200 \\ 0.6743 & 0.2576 & 0.6921 \end{bmatrix}$$

which also had an efficiency of 0.99280. The tensor results are almost identical: that is, the diagonalized tensors which result are essentially indistinguishable. The TSVD method was also implemented in a fortran code, using double-precision arithmetic, to compare it with Xie's which was likewise in double-precision fortran: results were the same.

However, several reasonable alternative tensor decomposition methods gave the same results! For example, constructing a block circulant matrix using the 3×3 layers of the sample variogram tensor, or taking the mean 3×3 layer, computing its SVD, and deducing the TSVD, gave the result:

$$B = \begin{bmatrix} 0.3974 & -0.9163 & -0.0502 \\ 0.6194 & 0.3083 & -0.7220 \\ 0.6770 & 0.2558 & 0.6900 \end{bmatrix}$$

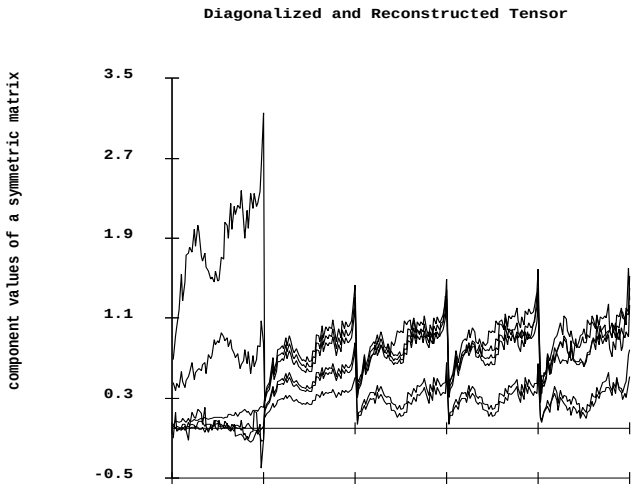


FIGURE 3.5. Five $3 \times 3 \times 50$ tensors shown in columns: diagonalized tensor; rank 1,2, and 3 reconstructions; and the original tensor at right. Since the 3×3 matrices are symmetric, only 6 components appear.

with an efficiency of 0.99278 (essentially identical). Using Geladi, et al.'s method ([30]), i.e. taking the SVD of $\langle T, T \rangle$ (the inner-product resulting in a $p \times p$ matrix), and using the symmetric tensors formed of those singular vectors,

$$B = \begin{bmatrix} 0.4028 & -0.9139 & -0.0505 \\ 0.6190 & 0.3126 & -0.7205 \\ 0.6742 & 0.2589 & 0.6917 \end{bmatrix}$$

with an efficiency of 0.99280. However, as mentioned, these three additional methods (circulants matrix, average layer, Geladi) have not been explored in this dissertation: they were simply tried on an *ad hoc* basis for comparison; and as the comparison shows, all five of these methods give essentially the same results!

■

Judging from that example, one might think that all methods are equivalent; however, other problems showed that it definitely does matter which method one uses. The reason that these methods gave such similar results is certainly indicative of the structure of the variogram tensor. It could, for example, indicate that there is drift appearing in the sample variogram (computed assuming constant mean), which then leads to "spurious" correlation between the vectors of the variogram tensor in the lags; it is also surely the case that sample variogram tensors already have most of their weight piled onto the variogram (diagonal) terms, which disposes them to

Method	W1	W2	W3	W4	W1 + W2
TSVD:	1.8349	0.5518	0.1782	0.1782	2.3867
Xie's:	1.7670	0.7111	0.1325	0.1325	2.4780
SVD:	1.6922	0.3923	0.3923	0.2664	1.9586
Average Layer:	0.8660	0.5775	0.5277	0.5277	1.4435

TABLE 3.1. Example results: weights on rank-one tensors, and best pair.

near-diagonalization, and into three symmetric components this way; furthermore, the variogram matrix function layers are nonnegative definite, which is not true in the general symmetric case.

So the test was applied to a strange example. On the other hand, geostatisticians seeking coregionalizations will always have sample variogram tensors with these properties, so that it may be that they will have their choice of methods for diagonalizations.

In order to demonstrate that things are not always so simple as in the example above, consider a case for which the methods give very different results. An arbitrary symmetric tensor was formed as the sum of two rank-one tensors, i.e.

$$T = \sum_{i=1}^{2} \lambda_i \underline{p}_i \otimes \underline{p}_i \otimes \underline{r}_i,$$

where the pairs were

```
P  =                R  =                  lambda  =
  0.2900   0.7986     0.1351    0.0600     1.4142    1.0000
  0.5160  -0.2521    -0.5944    0.2730
 -0.8060  -0.5465     0.4155   -0.1834
                      0.0482    0.3548
                     -0.1578   -0.7781
                     -0.4128    0.3256
                      0.0618    0.1313
                      0.5044   -0.1833
```

As shown in Table (3.5), the most weight appeared on the singular tensor found by the power method; but the most weight for a rank-two approximation of the form Xie sought is given by his method: 2.4780 versus 2.3867 for the power method.

This is typical: the power method finds the best single tensor with which to recompose, while Xie's method finds the best set of p such tensors. As a tensor decomposition, Xie's method suffers the disadvantages of being valid only for symmetric matrices (Flury's method, as originally given, was only valid for Positive Definite matrices). The TSVD works on stacks of general matrices.

Given that the TSVD works on tensors more general than those of the problem of coregionalization, applications beyond the variogram modelling problems of geostatistics are being sought. Another occasion for an application, treated in Chapter Six, involves looking at samples from the same wells over time. In such a case, one has a three-tensor T(samples,wells,time), which one could decompose to look for rank-one objects representing the best single sample profile over all wells over all time. The diagonalization is carried out in Chapter Six, and the improvement in information representation by rank discussed.

Finally, consider the anisotropic variogram matrix modelling problem, and how one might generalize this method of obtaining a linear coregionalization to cover it. One could approach the problem by using the following process:

- stack together all sample variogram matrices, by direction of interest;
- solve for the TSVD coregionalization corresponding to each direction;
- use univariate anisotropic modelling methods on the many univariate variograms on the diagonals;
- identify common structures, determine the ellipsoid in space which best describes the geometric anisotropy; and
- rescale space, reverting to the isotropic case.

3.6 Choosing Variables for Combined Analysis

We know that one of the applications of variogram analysis is as a precursor to kriging or cokriging, i.e. estimation or interpolation. How do we know if two variables will be better modelled together than separately? How do we know if multivariate analysis will be an improvement over univariate analyses? In particular, when do we cokrige two variables, say, rather than krige them separately?

To this last question we hypothesize an answer: for maximum benefit the correlation of the two variables should be strong locally, and fall off. That is, the correlation is "packed up" locally, and not spread all around. We also propose that, in order to determine this, one consider what might be called a "corhogram", defined as

$$\rho_{ij}(h) \equiv \frac{\gamma_{ij}(h)}{\sqrt{\gamma_{ii}(h)\gamma_{jj}(h)}}.$$

As one can see, $\rho_{ij}(h) \leq 1$ for valid models (a restatement of the Cauchy-Schwartz condition). Matheron [57] calls this the "codispersion coefficient", and Goovaerts [33, 35] and Wackernagel [93] have recently begun to study it as an aid in multivariate analysis. Wackernagel [91] showed that for second-order stationary phenomena, $\rho_{ij}(h)$ tends to the correlation coefficient as h tends to infinity.

This spatial statistic shows how close to the "Cauchy-Schwartz Envelope" two variables come: that is, how close they are to complete dependence. If two variables push the envelope (if $\rho_{ij}(h) \approx 1$), then they are almost completely correlated, and are hence surrogates (and certainly well-adapted to simultaneous interpolation in the case where one of the two variables is undersampled (due to expense or other constraints, for example the measurement of rainfall by ground gauges and by radar [7])). On the down side, using both could lead to a degeneracy in the kriging equations (as we will see); for example, by using the same variable twice: in that case, $\rho(h) \equiv 1$, but then the kriging equations will be degenerate.

Figure (3.6) shows the corhogram models for the pairs of variables deemed most successful in cokriging in the Nitrate study (to be discussed in Chapter Six). The examples demonstrate the "pile up" of correlation that occurs at short lags. All sample corhograms are plotted in the appendix, and as one can see by inspection, the corhograms for these best cokriging variables are at least as "piled up" as the others.

We also show, in Figure (3.7), all corhograms obtained from a study by Wackernagel [90], who used a very simple coregionalization model, consisting of only a nugget model and a spherical model. Notice the wide variation that occurs between the corhograms of the nine variables, and also note that only a few appear to be "piled up".

Corhograms failing to satisfy the rule above may give an improvement, although the best cokriging improvements in terms of cross-validation statistics in the Nitrate study (see Chapter Six) were achieved with corhograms having this property. For example, Carr et al. [13] demonstrated that they achieved improvement in estimation with cokriging, showing that estimates on one of two variables were better (in a mean-square sense, say) than those obtained by kriging. This was true in spite of the use of an invalid cross-variogram model, leading to the invalid corhogram of Figure (3.7). Marcotte [55] pointed out that the cross-variogram model was invalid. However, the second variable was more poorly reconstructed (again, using a mean-square criterion), which was not noted (mean-square difference of 11.74 for cokriging, versus 11.43 for kriging).

We will show in the chapter on interpolation methods that, if the corhogram of two variables is constant, no matter how large its value, there may be absolutely no gain in using both in the cokriging system; and we speculate, based on the results of the Nitrate study, that if the corhogram is not "piled up" near zero, cokriging will offer little or no gain over kriging.

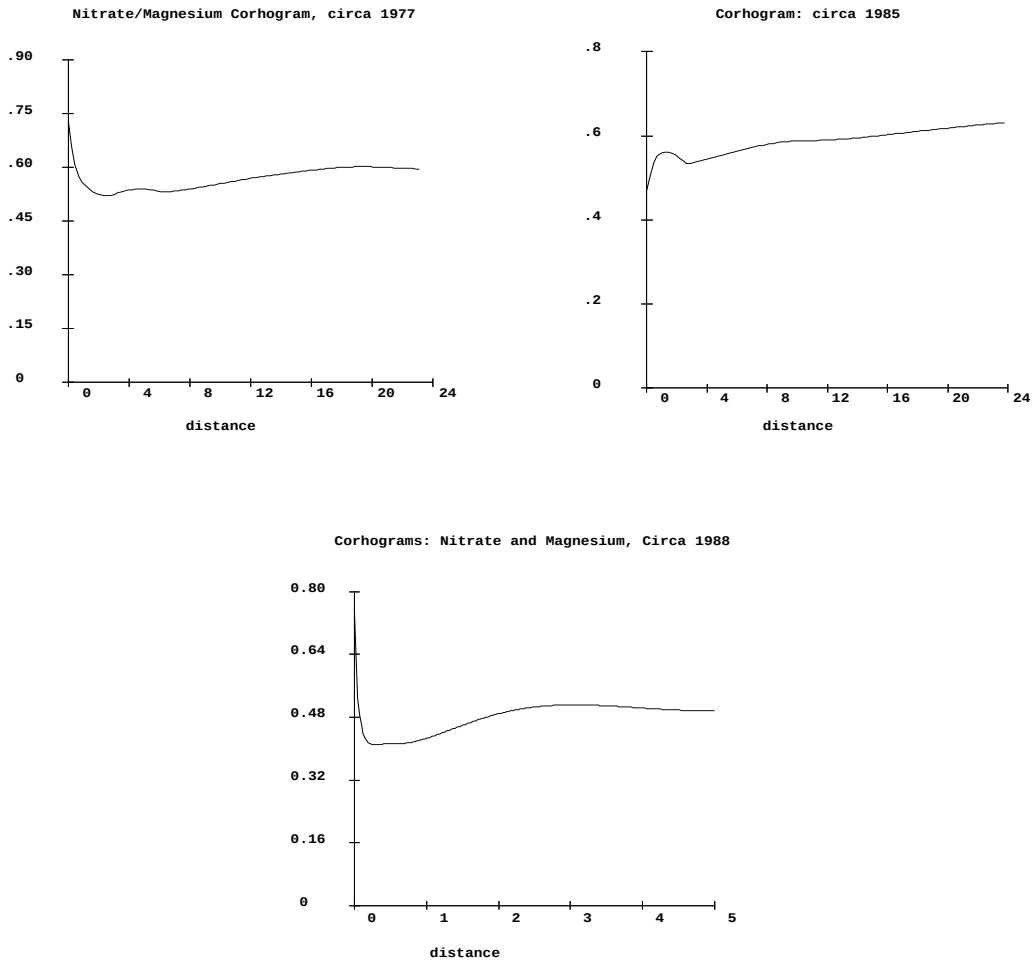


FIGURE 3.6. Corhogram model from 1977 data winner magnesium, 1985 data winner calcium, and 1988 data winner magnesium.

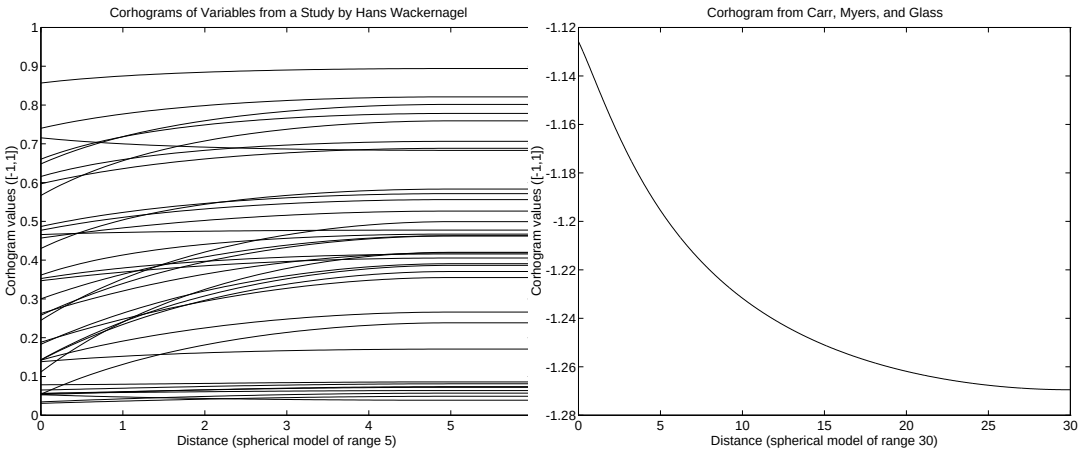


FIGURE 3.7. Left: all corhograms for a coregionalization of nine variables, using only nugget and spherical models (from a study by Wackernagel). Right: invalid corhogram for which cokriging seemed to lead to a substantial improvement over kriging (from a study by Carr et al.).

Chapter 4

Interpolation/Estimation Methods

4.1 Historical/Kernel Methods - Simple, Fast, Stable

Historically, kernel methods were employed whenever estimation or interpolation of a data set was needed. This was necessitated by the lack of computing power at the time. The most popular method then (still used today!) was some form of inverse-distance weighting: the estimate of z at the point x_0, z_0^*, is given by

$$z_0^* = \frac{1}{T} \sum_{i=1}^{N} \frac{1}{(x_i - x_0)^p} z_i$$

where $p \geq 0$ determines the rate at which the weights falls off with distance, and

$$T \equiv \sum_{i=1}^{N} \frac{1}{(x_i - x_0)^p} \tag{4.1.1}$$

is introduced to make the scheme unbiased (in the sense that the mean of the estimates is the same as the mean of the data, provided all data are used for every estimate). The extrapolatory nature of the kernel is clear: far from the data locations, the weights converge to a common value, which means that the estimate will be given by the arithmetic average of the data.

The value of N was not specified above: it could be the number of data locations, but historically N was some small number (perhaps four) which allowed for calculations to be carried out in an age without much computing power. Using only part of the data set will obviously affect the unbiasedness described above, as some points may be used more in the interpolation than others, which would lead to their influence being greater overall. One way to nullify this effect is to focus on the neighborhood size, rather than the nearest N neighbors: that is, to take all locations within a certain neighborhood, rather than a certain number of closest neighbors.

There are still those who use simple and unmotivated methods such as that described above. Kane et al. [49] explored models of this form for geochemical problems, developing an algorithm and program for optimizing the choice of p and neighborhood size (which affects the value of N) for a multivariate data set centering on uranium. They found that the values of p obtained varied widely from variable to variable, and even from place to place for the same variable.

The continued use of kernel methods is not hard to understand: these methods are easy, fast, and well-conditioned. Unfortunately, these three selling points do not necessarily add up to good maps!

What is a good map? A good map should reflect, as accurately as possible, the actual values of the quantity being mapped. That is, if one measures the quantity at unsampled locations, the correspondence between the measured values and the values represented on the map should be the best obtainable from the information that one had to make a map. Certainly differing measures of "correspondence" will lead to a variety of schemes, but a common measure is the square-root of the mean of squared differences:

$$\text{Error} = \sqrt{\frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{z}_\mathrm{i}^* - \mathrm{z}_\mathrm{i})^2},$$

where n is the number of locations at which one has tested what one might call the "map hypothesis".

With the advent of the computer, and, in particular, computer access, more sophisticated methods became available. Better and faster algorithms for the solution of linear systems meant that methods based on some principal of optimality became tractable. As is usual in any area of human endeavor, however, application has lagged far behind theory (as demonstrated by those who continue to use methods such as inverse-distance weighting).

One of the serious problems with kernel methods is that they do not take the sampling pattern into account, which means that they can be "snowed" by data which are surrogates for one another. For example, if the four neighbors used in a hypothetical case happen to come from a common tiny area, and estimation is taking place far from that site, then those four values may effectively represent only a small region; in such a case it might be wise to average those four and use three additional (better dispersed) sites for an estimate at the location of interest.

Data redundancy and other shortcomings are remedied by the kriging method, and the multivariate version called cokriging. Other methods also take these factors into account, but we will concentrate on these two.

Warrick et al. [95] made a comparison of kriging with other schemes, in particular inverse distance weighting with $p = 2$, on five separate data sets. They found that kriging was in all cases as good or better than the other methods. We obtain similar results in Chapter Six, where we include cokriging as well.

4.2 Kriging and Cokriging - Complex, Slow, Risky

4.2.1 Kriging

We begin by examining the interpolation procedure known as kriging, as it is a little simpler than cokriging, and serves as a good introduction. Given the title of this section, it is a wonder that anyone kriges at all! However, the fact that kriging is optimal in a sense to be described, and also has many good properties, justifies its use.

The usual scenario leading up to kriging is this: one is interested in a quantity, distributed in space, for which a non-random sample (the data) exists. Estimates of the values of the quantity at sites where there are no data are desired.

Kriging is based on a probabilistic assumption: that the data are a non-random sample of one realization of a random function, satisfying a stationarity condition. What this means, in the very strongest case, is that the random variables which occur at each point in space (giving rise to data sets, which are the non-random samples of a realization) are distributed according to the same (fixed) distribution everywhere in space.

The justification for the application of the probabilistic theory of regionalized variables - of attempting to estimate values of a single realization of a random variable - has been attacked upon occasion (see [81], and especially Philip and Watson [73]). These attacks have been met by responses from (Myers in [68], Journel in [44] and [46]), and by Matheron himself [63]. Matheron's "defense" [61] was actually written long before the cited attacks (1978), but Hasofer translated it into English because of "the appearance ... of a virulent attack on probabilistic models in Geostatistics ... by Philip and Watson."

Our interest in this dissertation is not particularly in arguing about the foundations of geostatistics, however: we do not seek to either prop up or tear down the structure, *per se*, but only to elaborate some methods for improving techniques which geostatisticians will use anyway.

First the ordinary kriging equations are derived, using the weakest stationarity assumption, i.e. intrinsic, which allows for the estimation of the theoretical variogram from the sample variogram. The kriging equations are obtained in the course of finding the best unbiased linear interpolator of a variable which minimizes its estimation variance: that is,

$$z^*(x) = \sum_{i=1}^{N} b_i(x; \{x_j\}) z_i, \tag{4.2.2}$$

such that

$$\mathrm{Var}(\mathrm{z}^*(\mathrm{x}) - \mathrm{z}(\mathrm{x})) \mathrm{is a minimum and}$$

$$\sum_{i=1}^{N} b_i(x; \{x_j\}) = 1 (\text{unbiasedness}).$$

One proceeds via constrained optimization: taking advantage of the constraint, note first of all that

$$E[z - \sum_{i=1}^{N} b_i z_i]^2 = E\left[\sum_{i=1}^{N} b_i (z - z_i)\right]^2,$$

where the reliance of the b on x and x_j has been suppressed. Minimize the function

$$E\left[\sum_{i=1}^{N} b_i(z - z_i)\right]^2 - 2\mu(1 - \sum_{i=1}^{N} b_i), \tag{4.2.3}$$

where $z(x)$ is the true value of the realization at the location x. Expanding the sum

$$E\left[\sum_{i=1}^{N} b_i(z - z_i)\right]^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} b_i b_j E[(z - z_i)(z - z_j)],$$

note that

$$E[(z - z_i)(z - z_j)] = \gamma(x_i - x) + \gamma(x_j - x) - \gamma(x_i - x_j).$$

This follows from the expansion

$$\gamma(x_i - x_j) =$$

$$\frac{1}{2}E[z(x_i) - z(x_j)]^2 = \frac{1}{2}E[z(x_i) - z(x) + z(x) - z(x_j)]^2 =$$

$$\frac{1}{2}E[z(x_i) - z(x)]^2 + E[(z(x_i) - z(x))(z(x) - z(x_j))] + \frac{1}{2}E[z(x_j) - z(x)]^2 =$$

$$\gamma(x_i - x) - E[(z(x_i) - z(x))(z(x_j) - z(x))] + \gamma(x_j - x),$$

which is solved for $E[(z(x_i) - z(x))(z(x_j) - z(x))]$.

Thus

$$E\left[\sum_{i=1}^{N} b_i(z - z_i)\right]^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} b_i b_j[\gamma(x_i - x) + \gamma(x_j - x) - \gamma(x_i - x_j)],$$

which reduces to

$$E\left[\sum_{i=1}^{N} b_i(z - z_i)\right]^2 = 2\sum_{i=1}^{N} b_i\gamma(x_i - x) - \sum_{i=1}^{N}\sum_{j=1}^{N} b_i b_j\gamma(x_i - x_j).$$

So the function to minimize is

$$2\sum_{i=1}^{N} b_i\gamma(x_i - x) - \sum_{i=1}^{N}\sum_{j=1}^{N} b_i b_j\gamma(x_i - x_j) + 2\mu(1 - \sum_{i=1}^{N} b_i).$$

Differentiating with respect to the b_i and μ leads to the following linear system:

$$\sum_{i=1}^{N} \gamma(x_k - x_i)b_i + \mu = \gamma(x_k - x) \text{ and } \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{b_i} = 1 \tag{4.2.4}$$

This system can be written in the matrix form

$$\begin{bmatrix} \Gamma & \underline{1} \\ \underline{1}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{b} \\ \mu \end{bmatrix} = \begin{bmatrix} \underline{\gamma}^x \\ 1 \end{bmatrix}, \tag{4.2.5}$$

where Γ is the matrix of variograms ($\Gamma_{ij} = \gamma(x_i - x_j)$), $\underline{1}$ is a column vector of 1's, and $\underline{\gamma}^x$ is a vector of variogram values relating the position at which one wishes the estimate (x) to the data locations (x_i): $\gamma_i^x = \gamma(x_i - x)$. Myers [66] has shown that this system can be extended in a very simple manner to the matrix (cokriging) case by replacing variograms where they appear in the kriging equations by variogram matrices, and the 1's by identity matrices of the proper size.

There are many forms of kriging (and cokriging):

- simple - assumes known mean;
- ordinary - estimates mean (the name comes from the fact that this is the most commonly used form [42]);
- universal - estimates drift, or mean surface, as a linear combination of a set of given linearly independent functions (ordinary is the special case of using only one function, a constant) [21];
- local - uses a moving neighborhood from which to choose locations;
- global - uses all data locations (the advantages of which are explored in [20]);
- disjunctive - a non-linear transformation of the data is carried out via Hermite polynomials to "normalize it", followed by a modified system of equations [60, 5, 77];
- indicator - a transformation of the data values to the set $\{0, 1\}$, which is useful (for example) when one is only interested in whether a quantity exceeds a certain threshold [43, 83];
- factorial - cokriging applied to chosen linear combinations of the original variables, usually obtained from principal components analysis [79, 28];
- point - values are assumed to come from point sources, and point values elsewhere are estimated;
- block - estimation of spatial averages, rather than point values;

and the list goes on. Some of these types of kriging are generally combined, e.g. local and ordinary; others represent transformations of the data which are carried out before the kriging process begins, according to the standard kriging equations (e.g. factorial, indicator).

While Matheron is credited with developing kriging, under the theory of regionalized variables, it is interesting to note that kriging arose at about the same time in meteorology: according to Cressie [17], Gandin in the Soviet Union developed some of the same concepts as those found in geostatistics under the name of "Objective Analysis". In place of the variogram, they use the "homogeneous structure function"; in place of simple kriging, "optimum interpolation"; ordinary kriging is "optimum interpolation with normalization of weighting factors"; and simple cokriging is called "optimum matching fields".

Myers [67] proved that strict conditional negative definiteness of the variogram model implies invertibility of the coefficient matrix of the kriging system. The proof is worth including. A conditionally negative definite function G is one satisfying

$$\sum\sum \lambda_i\lambda_j G(x_i - x_j) \leq 0, \quad \forall\{\lambda_i\} \ni \sum \lambda_i = 0;$$

a strictly conditionally negative definite function G satisfies

$$\sum\sum \lambda_i\lambda_j G(x_i - x_j) < 0$$

under the same conditions, with equality only if $\underline{\lambda} \equiv \underline{0}$.

The system (4.2.5) is non-invertible if and only if $\exists$ non-zero vector $\begin{bmatrix} \underline{U} \\ V \end{bmatrix} \ni$

$$\begin{bmatrix} \Gamma & \underline{1} \\ \underline{1}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{U} \\ V \end{bmatrix} = \begin{bmatrix} \underline{0} \\ 0 \end{bmatrix}.$$

This is true if and only if

$$\Gamma\underline{U} + \underline{1}V = \underline{0} \text{and} \underline{1}^{\mathrm{T}}\underline{\mathrm{U}} = 0.$$

These two equations imply that

$$\underline{U}^T\underline{1} = 0 \text{and} \underline{\mathrm{U}}^{\mathrm{T}}\Gamma\underline{\mathrm{U}} + \underline{\mathrm{U}}^{\mathrm{T}}\underline{1}\mathrm{V} = \underline{\mathrm{U}}^{\mathrm{T}}\Gamma\underline{\mathrm{U}} + 0\mathrm{V} = \underline{\mathrm{U}}^{\mathrm{T}}\Gamma\underline{\mathrm{U}} = 0$$

But $\underline{1}^T\underline{U} = 0$ means that the weights $\underline{U}$ satisfy the constraint of conditional negative definiteness, and strict conditional negative definiteness means that

$$\underline{U}^T\Gamma\underline{U} = 0 \Longrightarrow \underline{U} = \underline{0};$$

and consequently that

$$\underline{1}V = \underline{0}, \Longrightarrow V = 0.$$

Thus, only the zero vector is in the null-space, which means that the system is invertible. Note, however, that invertibility does not imply that the system is well-conditioned! In fact, it is known that for certain standard models, e.g. the gaussian

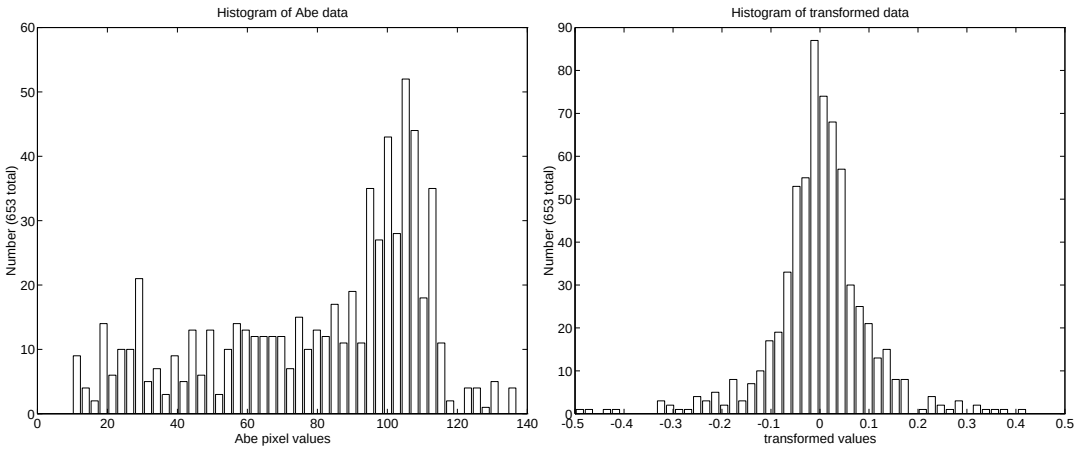


FIGURE 4.1. Histograms of Abe's pixel values (original data) and the transformed data of the dual kriging equations.

without a nugget, the coefficient matrix of the kriging system may be very poorly conditioned indeed [74, 71].

The estimate at position x is then given by:

$$z^*(x) = \left[\begin{array}{cc} \underline{z}^T & 0 \end{array}\right] \left[\begin{array}{c} \underline{b} \\ \mu \end{array}\right] = \left[\begin{array}{cc} \underline{z}^T & 0 \end{array}\right] \left[\begin{array}{cc} \Gamma & \underline{1} \\ \underline{1}^T & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \underline{\gamma}^x \\ 1 \end{array}\right]. \tag{4.2.6}$$

In practice it may be best to compute what we call the "transformed data", by first multiplying

$$\left[\begin{array}{cc} \underline{z}^T & 0 \end{array}\right] \left[\begin{array}{cc} \Gamma & \underline{1} \\ \underline{1}^T & 0 \end{array}\right]^{-1} \equiv \left[\begin{array}{cc} \underline{\hat{z}} & m \end{array}\right]:$$

by doing so, the interpolating function is expressed in the following simple form:

$$z^*(x) = \langle \underline{\hat{z}}, \underline{\gamma}^x \rangle + \langle m, 1 \rangle = \sum_{i=1}^{N} \hat{z}(x_i)\gamma(x - x_i) + m. \tag{4.2.7}$$

The variogram thus serves as a kernel function for z using the transformed data $\hat{z}$, and that the mean of the realization is also estimated. This form of the estimator is called the **dual form**. Estimation is cheap using the dual form: estimation at x occurs at the cost of an inner-product of two N-vectors, one of which must be calculated from the variogram model; plus the addition of the computed mean. Figures (4.2) and (4.1) show the form the transformed data takes for a well-known figure in American history.

Notice that the variogram is actually weighting locations which are far from the point at which the estimate is desired more than nearby locations. This is exactly the

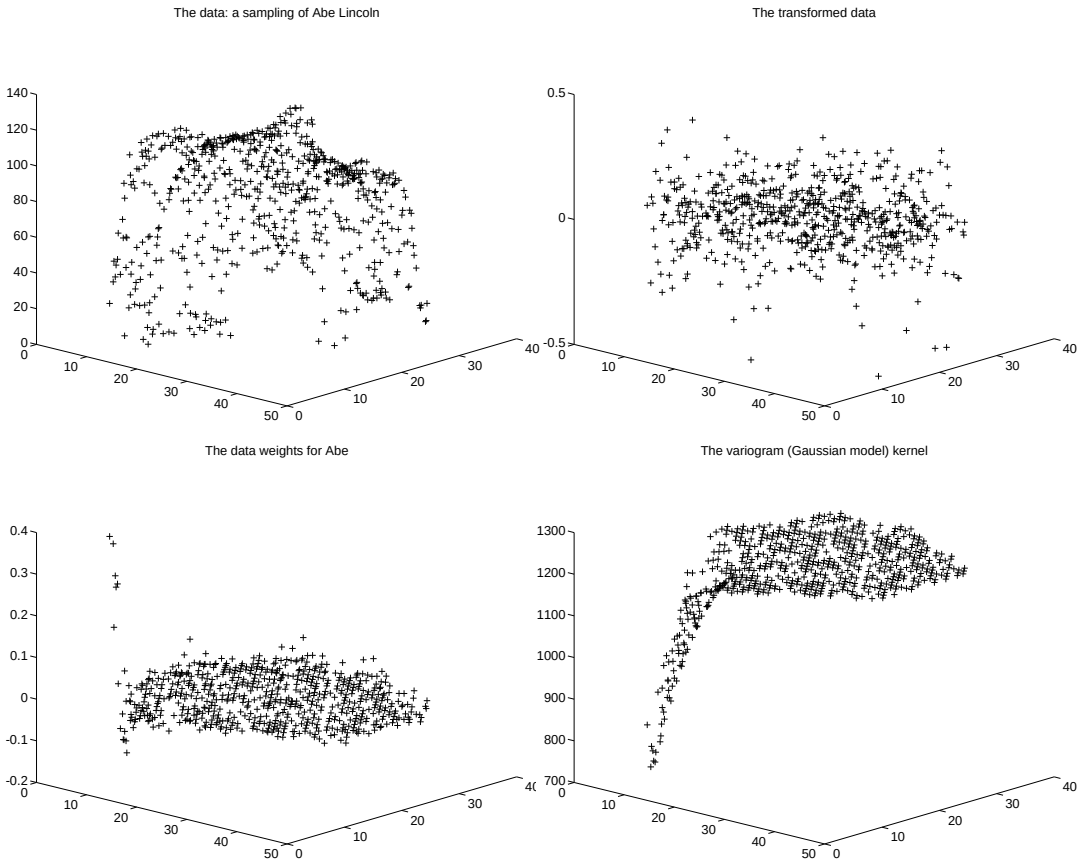


FIGURE 4.2. A sample of Abe Lincoln's face was used to estimate the missing portion. The dual form required the computation of the transformed data, which is obviously not as smooth as Abe! The variogram acts as an interpolating kernel on this transformed data, while the data weights are used with the actual values of Abe's face to get the estimate (here taken in the upper left corner, in his hair).

opposite of the data weights, which are larger for nearby locations. Furthermore, the variogram of the transformed data has a very interesting property(Figure (4.3)): it is highest at the origin! This very peculiar behavior indicates that close neighbors in the transformed data are actually correlated less than neighbors farther apart: there is an "opposites attract" interaction.

The dual form of the kriging equations also may be put to good use in other ways [29, 78, 75]. For example, it shows that the interpolating function is as differentiable as the variogram model. This is an important consideration if one has some insight into the differentiability of the underlying random function. Thus, while there may be little apparent difference in using a spherical model rather than an exponential model (both monotonically increasing without inflection points, tailing off quickly to their sills), one model has zero fourth derivative, and the other has non-zero derivatives of all orders. This provides more insight into the importance of the variogram modelling step.

The estimator of a realization thus also gives us estimates of the derivatives. For general models such as the gaussian, exponential, etc., one obtains derivatives of all orders, which may be necessary for groundwater modelling, say, and thus may need to be estimated anyway.

One can also show that the kriging situation reduces to easily understood forms in certain cases. An example follows.

■ **Example 1: One-dimensional linear model**

The linear model has the form

$$\gamma(x-y) = c|x-y|.$$

The dual form of kriging indicates that, after solving the global kriging system, the interpolating function will be

$$z^*(x) = c\sum \hat{z}_i|x-x_i| + m(x).$$

But this interpolator is linear in x, and must pass through the data points, which means that one need only play "connect the dots", indicating that solving the kriging equations is unnecessary.

One also sees that the interpolator will not be differentiable (in general) at the data locations: its derivative is

$$(z^*)'(x) = c\sum \hat{z}_i H(x-x_i).$$

Thus, the transformed data value $\hat{z}_i$ represents the magnitude of the jump discontinuity for the linear model at x_i.

■ **Example 2: One-dimensional model with nugget**

What is the effect of a nugget? The dual form of the kriging equations for an arbitrary variogram model γ with a nugget n is

$$z^*(x) = \sum \hat{z}_i(n(x-x_i) + \gamma(x-x_i)) + m, \tag{4.2.8}$$

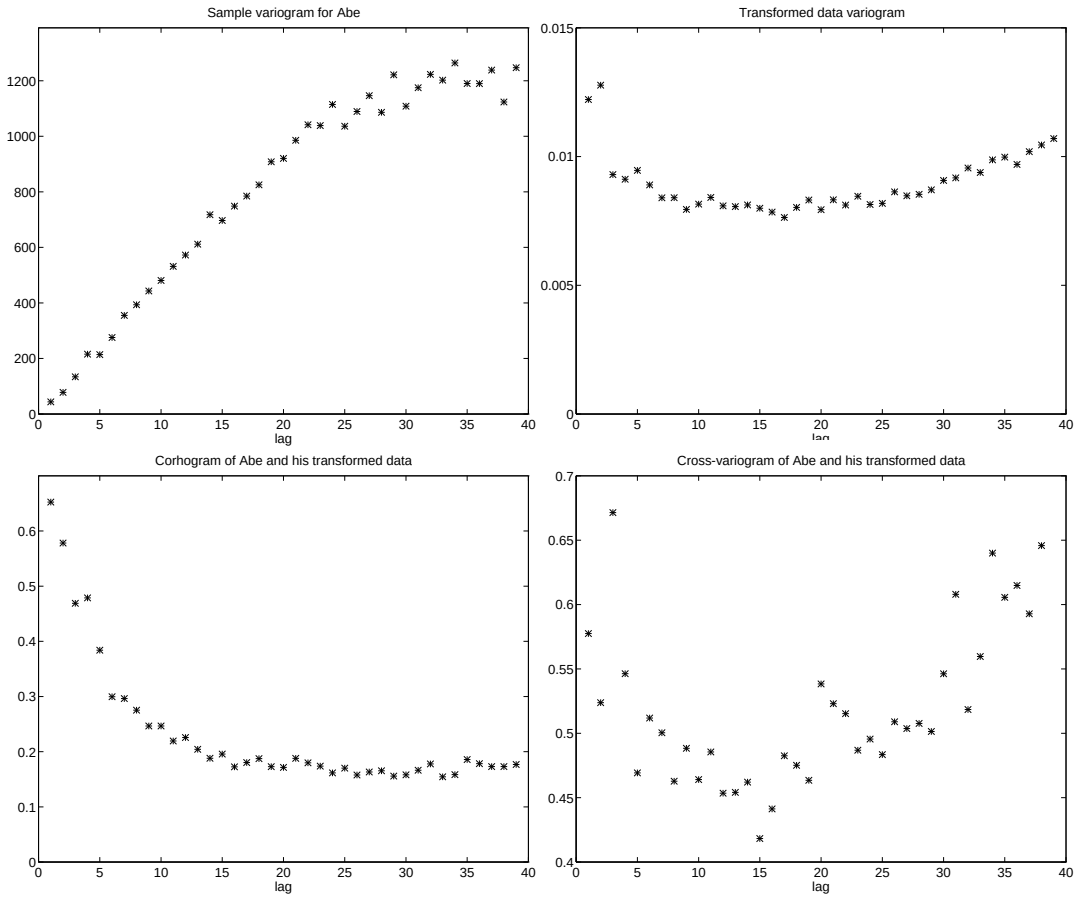


FIGURE 4.3. Top-left: Abe's isotropic sample variogram; top-right: transformed data sample variogram. Notice that the transformed data variogram is better correlated at mid-range, which makes it rather strange as variograms go. Abe's is much more typical. The corhogram (of Abe and Abe transformed) (bottom-left) is also striking, quite piled up, and more open to interpretation than the ill-mannered cross-variogram (bottom-right).

where

$$n(h) = \begin{cases} n, & h \neq 0, \\ 0 & h = 0. \end{cases}$$

Rewriting (4.2.8) (relying on the fact that $\sum \hat{z}_i = 0$),

$$z^*(x) = \begin{cases} c \sum \hat{z}_i \gamma(x - x_i) + m & x \neq x_i \; \forall i \\ c \sum \hat{z}_i \gamma(x - x_i) + m - n\hat{z}_i & x = x_i. \end{cases}$$

One remarks that, although kriging is an exact interpolator, there will be a discontinuity at data points whenever a nugget is used (which is frequently the case in models found in the literature). Thus, the "kriging surface" (i.e., the surface of the function one obtains by estimating at all points) need not be continuous, and pass through the data; kriging in this case basically smooths, but may leap up discontinuously from the smooth surface at data locations to reach a data value [75].

The jump discontinuities at the data locations are given by the value of the transformed datum times the (negative of the) nugget. This provides an intuitive understanding to the values of the transformed data: the absolute value of transformed data represents the deviation of that location from the "estimation surface" (i.e. the limiting value kriging would attribute to a data location arbitrarily close to x_i). Thus, where transformed data are higher, there is less of what one tends to think of as smooth interpolation taking place (see Figures (4.2)) and (4.1): the nugget in Abe's case was about 1.41, so, since the transformed values were between -.5 and .5, expect jumps of at most .7 in the map of Abe; this is rather small given Abe's pixel values).

■

By virtue of the condition used to derive the kriging equations, the minimized estimation variance at x is obtained, which is called the **kriging variance**. It is given by

$$\text{krigingvariance(x)} = \begin{bmatrix} (\underline{\gamma}^x)^T & 1 \end{bmatrix} \begin{bmatrix} \Gamma & \underline{1} \\ \underline{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \underline{\gamma}^x \\ 1 \end{bmatrix} \tag{4.2.9}$$

$$= \begin{bmatrix} (\underline{\gamma}^x)^T & 1 \end{bmatrix} \begin{bmatrix} \underline{b} \\ \mu \end{bmatrix} = \sum_{i=1}^{N} b_i \gamma(x - x_i) + \mu.$$

The number of operations required to calculate the kriging variance is much larger than the number required for the estimates, unfortunately: one must actually obtain the weights (rather than using the transformed data) for each value of x, which means storing the matrix inverse (which is $(N + p) \times (N + p)$), and multiplying the inverse and the weight vector for each estimate.

The name "variance" is a misnomer of sorts: it obviously depends on the model chosen for the kriging, which makes it liable to errors in modelling. It is the true estimation variance only if the model chosen actually corresponds to the random

function which gave rise to the realization. One notes also (from equation (4.2.9)) that the kriging variance is completely independent of the data values.

On the other hand, the kriging variance can give us some insight into the sampling pattern: it is low where there is much data, typically, and high where there is not much data, so it shows us problem spots in the sampling pattern. The variances will also be used in the calculation of theoretical characteristics of the cross-validation results, which are discussed later.

Issues and Problems:

- Note that, for second-order stationary phenomena, one can write $c(x_i - x_j) = c(0) - \gamma(x_i - x_j)$, and so one can rewrite the equations (4.2.4) in terms of the covariances.
- The dual form is equivalent to radial basis function interpolation, with the variogram serving as the natural kernel function on transformed data. Radial basis function users usually make their choice for a kernel arbitrarily (based on "visual pleasure" in the fit, or the like), rather than on a spatial statistic like the variogram. Myers [70] has shown the equivalence, and that cokriging is a natural generalization of radial basis functions in multivariate problems.
- The variance one obtain in the course of kriging is a function of the model choice, and it is independent of the sample values. It should be interpreted with these factors in mind.
- Global kriging may lead to large linear systems; the matrix condition number may become large, even infinite, leading to solver problems; but the interpolator which results is easy to use from a computational standpoint.
- Local kriging requires that one sort data into the local neighborhoods, form a linear system (and solve it) for each estimate desired. It also may lead to discontinuities in the computed drift surface.
- Kriging is not restricted to interpolation: for example, Yfantis et al. [100] used it in the problem of data compression (comparing it with the JPEG procedure, another lossy algorithm). They found that their kriging compression algorithm gave more "graceful degradation" than the JPEG scheme; however, since kriging involves some variogram analysis prior to compression, the kriging method was slower.

Certainly there exist other, better references for this development. The point of this introduction is not to show the derivation of the kriging equations, *per se*, however, but rather to give the reader some of the flavor of the technique before we delve into it more deeply.

We now consider the cokriging equations, but do so in the context of a new formulation which constitutes one of our contributions to this subject.

4.2.2 A Better Algorithm for Cokriging

Myers [66] gave form to the cokriging equations, but, as will be shown, a form which suffers from the unfortunate property that it entails the solution of a system of equations much larger than necessary. We transform Myers's system into a set of smaller systems, whose solution provides simultaneously both the kriging and cokriging results.

The ultimate cokriging method would solve a p-way cokriging system by giving the results of all $p-1, p-2, \cdots$, 1-way (kriging) systems as well, in which case one would simply cokrige all variables, and, based on cross-validation results of each subset of cokrigings, choose that combination which did the best. While still short of that goal, the new formulation leads to one set of $p-1, p-2, \cdots, 2-$way, and all kriging solutions in the process of cokriging a set of p variables.

The Two Variable Case

The universal cokriging estimator for the intrinsic vector-valued random function $\underline{z}$ is given by the equations

$$\underline{z}^*(x_0) = \sum_{i=1}^{N} \Gamma_i^T \underline{z}(x_i),$$

where the weight matrices Γ_i satisfy the conditions that

$$\sum_{i=1}^{N} F_l(x_i)\Gamma_i = F_l(x_0), \quad l \in \{1, \ldots .p\}; \tag{4.2.10}$$

where the p matrices F_l are given by

$$F_l(x) = f_l(x) * I,$$

and the $f_l(x)$ are independent functions forming a basis for the drift surface [66]. The weight matrices are determined by the $N+p$ sets of equations given by the constraints (4.2.10), and the sets of linear equations

$$\sum_{i=1}^{N} V(x_i - x_j)\Gamma_j \underline{z}(x_i) + \sum_{l=1}^{p} F_l(x_i)\mu_l = V(x_i - x), \;\; i \in \{1, \cdots, N\}.$$

V is the variogram matrix function, and the μ_l are matrices of Lagrange multipliers. Note that this is precisely the form of the universal kriging equations, where scalar quantities have been replaced by matrices.

Consider first at the two-variable case (that is, cokriging two variables), in order to determine what change is involved. Myers's formulation, i.e. the system of size $2(N+p) \times 2(N+p)$, is given by

$$\begin{bmatrix} V & F \\ F^T & 0 \end{bmatrix} \begin{bmatrix} \Gamma \\ \mu \end{bmatrix} = \begin{bmatrix} V_0 \\ F_0 \end{bmatrix}, \tag{4.2.11}$$

where the elements of V are the block variogram matrices (which, at the risk of confusion, will also be called V) made up of the variograms and cross-variogram of the two variables for each pair of data locations, and F is the matrix function of p linearly independent F_l matrix functions (whose coefficients are to be determined by cokriging). On the right-hand side is the "column matrix" of variogram matrices referred to x_0, the location at which the estimate is desired; and similarly for F_x. This is represented (in all its glory) by

$$\begin{bmatrix} V(x_1-x_1) & V(x_1-x_2) & \cdots & V(x_1-x_N) & F_1(x_1) & \cdots & F_p(x_1) \\ V(x_2-x_1) & V(x_2-x_2) & \cdots & V(x_2-x_N) & F_1(x_2) & \cdots & F_p(x_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ V(x_N-x_1) & V(x_N-x_2) & \cdots & V(x_N-x_N) & F_1(x_N) & \cdots & F_p(x_N) \\ F_1(x_1) & F_1(x_2) & \cdots & F_1(x_N) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ F_p(x_1) & F_p(x_2) & \cdots & F_p(x_N) & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_N \\ \mu_1 \\ \vdots \\ \mu_p) \end{bmatrix} = \begin{bmatrix} V(x_1-x_0) \\ V(x_2-x_0) \\ \vdots \\ V(x_N-x_0) \\ F_1(x_0) \\ \vdots \\ F_p(x_0) \end{bmatrix},$$

where the subscripts refer to the data points determining the distance used by the matrix variogram function.

The trick is simply to permute rows and columns of this large matrix so that the variograms (diagonal elements of the block matrices of V) and cross-variogram (off-diagonal elements) get separated. For two variable cokriging, define the permutation matrix

$$P \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix},$$

or

$$P \equiv \begin{bmatrix} \underline{e}_1 & \underline{e}_{N+p+1} & \underline{e}_2 & \underline{e}_{N+p+2} & \cdots & \underline{e}_{N+p} & \underline{e}_{2(N+p)} \end{bmatrix},$$

where $\underline{e}_i$ is the Euclidean unit vector with 1 in the ith place, and zeros elsewhere. The generalization is obvious for other numbers of variables, and the result is the same, in the sense that the variables get separated similarly. The choice of this permutation is motivated by considering a cokriging matrix which has all cross-variogram terms zero: it is obvious that it can be split into two separate kriging matrices, and P is the permutation matrix which accomplishes that.

Define

$$X \equiv P \begin{bmatrix} V & F \\ F^T & 0 \end{bmatrix} P^T \equiv \begin{bmatrix} K_1 & C \\ C & K_2 \end{bmatrix},$$

where K_1 and K_2 represent the coefficient matrices of the kriging systems for the two variables, and C represents the cross-variogram information given in off-diagonal of the variogram matrix model. The inverse of the matrix X is given simply in terms of the matrix inverses of K_1 and K_2 (which are needed to get the kriging results) and the matrix inverses of two other $N \times N$ matrices:

$$M_1 \equiv I - K_1^{-1} C K_2^{-1} C \tag{4.2.12}$$

and

$$M_2 \equiv I - K_2^{-1} C K_1^{-1} C. \tag{4.2.13}$$

The form of these matrices and their consequences of their invertibility suggest a link to the Cauchy-Schwartz condition, which, for a pair of variables, is

$$\sigma_{12}^2 \leq \sigma_1^2 \sigma_2^2,$$

where σ_{12} is the covariance of the two, and on the right-hand side are the variances σ_1^2 and σ_2^2. Rewrite that as

$$m_1 \equiv 1 - (\sigma_1^2)^{-1}(\sigma_{12})(\sigma_2^2)^{-1}(\sigma_{12})$$

with the condition that

$$m_1 \geq 0.$$

Comparing m_1 and (4.2.12) shows that the kriging matrices are playing the roles of the variances (appropriately enough, as the variogram is the decomposition of the variance) and the cross-variogram matrix is playing the role of the covariance. The Cauchy-Schwartz condition, reflected in the inequality above, guarantees strict positive definiteness in a 2×2 matrix, which guarantees unique solvability of the system. How is the inequality reflected in this matrix case?

M_1 is non-invertible iff $\exists \underline{x} \ni$

$$M_1 \underline{x} = \underline{0}, \iff K_1^{-1} C K_2^{-1} C \underline{x} = \underline{x}.$$

This does not happen if

$$1 - \| K_1^{-1} C K_2^{-1} C \|_2 \geq 0,$$

and similarly for the case of (4.2.13).

Invertibility of the coefficient matrix is therefore possible (provided the kriging systems are invertible) only if the largest singular values of the matrices $K_1^{-1} C K_2^{-1} C$ and $K_2^{-1} C K_1^{-1} C$ are less than 1.

One can gain some appreciation for this by starting with two independent variables, in which case C is zero: then the two matrices $K_1^{-1} C K_2^{-1} C$ and $K_2^{-1} C K_1^{-1} C$ are also zero. Now, as correlation is "added", via the cross-variogram, allowing the norm of the matrix C to increase, the singular values of $K_1^{-1} C K_2^{-1} C$ and $K_2^{-1} C K_1^{-1} C$

move continuously on the real line, out from zero (the "singular value" of the zero matrix). At some point, the largest singular value (and hence the norm of these matrices) may increase beyond 1, at which time the system will no longer be invertible for all right-hand sides.

If the kriging matrices and M_1 and M_2 are invertible, then inverting X is easy:

$$\begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} K_1 & C \\ C & K_2 \end{bmatrix} = \begin{bmatrix} I & K_1^{-1}C \\ K_2^{-1}C & I \end{bmatrix},$$

and

$$\begin{bmatrix} I & -K_1^{-1}C \\ -K_2^{-1}C & I \end{bmatrix} \begin{bmatrix} I & K_1^{-1}C \\ K_2^{-1}C & I \end{bmatrix} = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}$$

Let

$$\begin{gathered} A_1 \equiv K_1^{-1}C \\ \text{and} \\ A_2 \equiv K_2^{-1}C; \end{gathered} \tag{4.2.14}$$

then

$$X^{-1} = \begin{bmatrix} M_1^{-1} & 0 \\ 0 & M_2^{-1} \end{bmatrix} \begin{bmatrix} I & -A_1 \\ -A_2 & I \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix}.$$

Computation of Estimates

The kriging estimates are

$$\underline{z}_k^*(x_0) = \begin{bmatrix} V_{10}^T & 0 \\ 0 & V_{20}^T \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix},$$

whereas the cokriging estimates are given by

$$\underline{z}_c^*(x_0) = \begin{bmatrix} V_{10}^T & C_0^T \\ C_0^T & V_{20}^T \end{bmatrix} \begin{bmatrix} M_1^{-1} & 0 \\ 0 & M_2^{-1} \end{bmatrix} \begin{bmatrix} I & -A_1 \\ -A_2 & I \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}.$$

In either case one must compute

$$\begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \equiv \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}.$$

Then the kriging results are given by the simple dot products

$$\underline{z}_k^*(x_0) = \begin{bmatrix} V_{10}^T & 0 \\ 0 & V_{20}^T \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} V_{10}^T\delta_1 \\ V_{20}^T\delta_2 \end{bmatrix},$$

while the cokriging estimates can be simplified still further:

$$\underline{z}_c^*(x_0) = \begin{bmatrix} V_{10}^T & C_0^T \\ C_0^T & V_{20}^T \end{bmatrix} \begin{bmatrix} M_1^{-1} & 0 \\ 0 & M_2^{-1} \end{bmatrix} \begin{bmatrix} I & -A_1 \\ -A_2 & I \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix}$$

$$= \begin{bmatrix} V_{10}^T & C_0^T \\ C_0^T & V_{20}^T \end{bmatrix} \begin{bmatrix} B_1 & -B_1 A_1 \\ -B_2 A_2 & B_2 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix},$$

where

$$B_1 \equiv M_1^{-1} \text{ and } \mathrm{B}_2 \equiv \mathrm{M}_2^{-1}.$$

All this can be stored in the same size matrix as originally given, once the matrix products have been carried out.

That is followed by one last multiplication, so that, in the end, cokriging at a particular site will take twice the computation that kriging requires:

$$\underline{z}_c^*(x_0) = \begin{bmatrix} V_{10}^T & C_0^T \\ C_0^T & V_{20}^T \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix},$$

in a form entirely analogous to that of the kriging estimates, with

$$\begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} \equiv \begin{bmatrix} B_1 & -B_1 A_1 \\ -B_2 A_2 & B_2 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix}.$$

One of the advantages of this scheme is that the inversion of the $2(N+p)\times 2(N+p)$ matrix is replaced by the inversion of four $(N+p)\times(N+p)$ matrices (the "Cauchy-Schwartz" matrices M_1 and M_2 in addition to the two $(N+p)\times(N+p)$ kriging matrices). The kriging matrices will need to be calculated anyway, however, so this is really quite a savings.

In the appendix is an example, using a Matlab code, demonstrating that cokriging using Myers's approach involves the inversion of a large matrix with high condition number, whereas the procedure described herein means the inversion of the kriging systems (which is presumably necessary anyway), which may have condition numbers on the same order, but somewhat smaller, followed by the inversion of two matrices with small (≈ 1) condition numbers.

This also suggests that experimenting with various cross-variograms is easier than before: the kriging systems are solved once and for all, then the variety of cross-variograms of interest can be tried with much less computation.

First-Order Approximation of Cokriging Improvement

If $\|A_1 A_2\| << 1$ and $\|A_2 A_1\| << 1$, then

$$B_1 = M_1^{-1} = (I - A_1 A_2)^{-1} \approx I + A_1 A_2;$$

and similarly for B_2. Then

$$\underline{z}_c^*(x_0) \approx$$

$$\begin{bmatrix} V_{10}^T & C_0^T \\ C_0^T & V_{20}^T \end{bmatrix} \begin{bmatrix} I + A_1 A_2 & 0 \\ 0 & I + A_2 A_1 \end{bmatrix} \begin{bmatrix} I & -A_1 \\ -A_2 & I \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

$$= \left(\begin{bmatrix} V_{10}^T & 0 \\ 0 & V_{20}^T \end{bmatrix} + \begin{bmatrix} 0 & C_0^T \\ C_0^T & 0 \end{bmatrix} \right) \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} A_1A_2 & 0 \\ 0 & A_2A_1 \end{bmatrix} \right) \times$$
$$\left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} 0 & A_1 \\ A_2 & 0 \end{bmatrix} \right) \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$
$$= \left(\begin{bmatrix} V_{10}^T & 0 \\ 0 & V_{20}^T \end{bmatrix} + \begin{bmatrix} 0 & C_0^T \\ C_0^T & 0 \end{bmatrix} \right) \times$$
$$\left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} A_1A_2 & -A_1(I - A_2A_1) \\ -A_2(I - A_1A_2) & A_2A_1 \end{bmatrix} \right) \times$$
$$\begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

$$= \underline{z}_k^*(x_0)$$
$$+ \left(\begin{bmatrix} V_{10}^T & 0 \\ 0 & V_{20}^T \end{bmatrix} + \begin{bmatrix} 0 & C_0^T \\ C_0^T & 0 \end{bmatrix} \right) \begin{bmatrix} A_1A_2 & -A_1(I - A_2A_1) \\ -A_2(I - A_1A_2) & A_2A_1 \end{bmatrix} \times$$
$$\begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} 0 & C_0^T \\ C_0^T & 0 \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

$$= \underline{z}_k^*(x_0) +$$
$$\begin{bmatrix} V_{10}^T & C_0^T \\ C_0^T & V_{20}^T \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} CK_2^{-1}C & -C(I - K_2^{-1}CK_1^{-1}C) \\ -C(I - K_1^{-1}CK_2^{-1}C) & CK_1^{-1}C \end{bmatrix} \times$$
$$\begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} 0 & C_0^T \\ C_0^T & 0 \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

$$= \underline{z}_k^*(x_0) + \begin{bmatrix} C_0^T K_2^{-1} d_2 \\ C_0^T K_1^{-1} d_1 \end{bmatrix} +$$
$$\begin{bmatrix} V_{10}^T & C_0^T \\ C_0^T & V_{20}^T \end{bmatrix} \begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} CK_2^{-1}C & -C(I - K_2^{-1}CK_1^{-1}C) \\ -C(I - K_1^{-1}CK_2^{-1}C) & CK_1^{-1}C \end{bmatrix} \times$$
$$\begin{bmatrix} K_1^{-1} & 0 \\ 0 & K_2^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}.$$

Keeping only the terms linear in C (or C_0),

$$\underline{z}_c^*(x_0) \approx \underline{z}_k^*(x_0) - \begin{bmatrix} d_2^T K_2^{-1}(CK_1^{-1}V_{10} - C_0) \\ d_1^T K_1^{-1}(CK_2^{-1}V_{20} - C_0) \end{bmatrix},$$

or, recalling that the kriging weights are $\Gamma_i = K_i^{-1} V_{i0}$,

$$\underline{z}_c^*(x_0) \approx \underline{z}_k^*(x_0) - \left[\begin{array}{c} d_2^T K_2^{-1}(C\Gamma_1 - C_0) \\ d_1^T K_1^{-1}(C\Gamma_2 - C_0) \end{array} \right].$$

Making use of the transformed data $(d')_i^T = d_i^T K_i^{-1}$,

$$\underline{z}_c^*(x_0) \approx \underline{z}_k^*(x_0) - \left[\begin{array}{c} (d_2')^T(CK_1^{-1}V_{10} - C_0) \\ (d_1')^T(CK_2^{-1}V_{20} - C_0) \end{array} \right] \equiv \underline{z}_k^*(x_0) - \left[\begin{array}{c} (d_2'')^T V_{10} - (d_2')^T C_0 \\ (d_1'')^T V_{20} - (d_1')^T C_0 \end{array} \right].$$

Thus, a calculation of the cokriging approximation requires storing another form of transformed data, but only two vector inner-products for an actual estimate.

This approximation implies that to first order it is the extent to which $C\Gamma_i$ differ from C_0 that determines whether it is worthwhile to cokrige. If

$$C\Gamma_1 = C_0 \mathrm{and} \mathrm{C}\Gamma_2 = \mathrm{C}_0,$$

that is, if

$$CK_1^{-1}V_{10} = C_0 \mathrm{and} \mathrm{CK}_2^{-1}\mathrm{V}_{20} = \mathrm{C}_0,$$

then cokriging may provide no improvement. While it will be interesting to consider under what conditions these hold, we have not yet done so.

One very interesting idea that now arises is to combine the results of Xie's method of coregionalization with this linear approximation: the point of his method is to make the cross-variogram terms small, which may put the diagonalized data squarely in line for this approximation. Therefore, we propose that one

- transform to diagonalized variables;
- model the new variables;
- cokrige with linear approximation; and
- transform back to the original variables.

This approach is examined in one part of the Nitrate study of Chapter Six.

The Elemental Coregionalization-Like Case

A striking result of this procedure is that cokriging variables modelled as the elemental constituent of the coregionalization case gives the same result as kriging. Start with the form of the variogram matrix in the case of a one-structure "coregionalization" (there are quotes around coregionalization because this is "trivially" coregionalized: there is only a single structure):

$$V(h) = \gamma(h) \left[\begin{array}{cc} a & c \\ c & b \end{array} \right] \equiv \gamma(h) V,$$

where the matrix V on the right hand side is nonnegative definite, and γ is a standard variogram model (conditionally negative definite function). Variables with this type of model are said to be "intrinsically coregionalized" [38]. Matheron [57] called this a case of "intrinsic correlation", and also showed in [62] that cokriging reduces to kriging.

Note what this form of the variogram matrix implies about the corhogram, introduced in the previous chapter: it has a constant value, independent of h:

$$\rho(h) = \frac{c}{\sqrt{ab}}.$$

Demonstrating that cokriging is equivalent to kriging in this case means that a flat corhogram may indicate that there is no sense in estimating the variables jointly.

The form of Myers's equations in this special case is

$$\begin{bmatrix} 0 & \gamma(x_1-x_2)V & \gamma(x_1-x_3)V & \cdots & \gamma(x_1-x_N)V & F_1 \\ \gamma(x_2-x_1)V & 0 & \gamma(x_2-x_3)V & \cdots & \gamma(x_2-x_N)V & F_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(x_N-x_1)V & \gamma(x_N-x_2)V & \gamma(x_N-x_3)V & \cdots & 0 & F_N \\ F_1 & F_2 & F_3 & \cdots & F_N & 0 \end{bmatrix},$$

which one can permute to

$$\begin{bmatrix} a\begin{bmatrix} K & F \\ F^T & 0 \end{bmatrix} & c\begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix} \\ c\begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix} & b\begin{bmatrix} K & F \\ F^T & 0 \end{bmatrix} \end{bmatrix},$$

while the swapped form of the equations is

$$\begin{bmatrix} aK & cK & aF & 0 \\ cK & bK & 0 & bF \\ aF^T & 0 & 0 & 0 \\ 0 & bF^T & 0 & 0 \end{bmatrix}. \tag{4.2.15}$$

Let

$$\Delta \equiv ab - c^2$$

To invert (4.2.15), apply the following operations on the left:

$$\begin{bmatrix} K^{-1} & 0 & 0 & 0 \\ 0 & K^{-1} & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\begin{bmatrix} \frac{b}{\Delta}I & \frac{-c}{\Delta}I & 0 & 0 \\ \frac{-c}{\Delta}I & \frac{a}{\Delta}I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ -aF^T & 0 & I & 0 \\ 0 & -bF^T & 0 & I \end{bmatrix}$$

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & \frac{-\Delta}{ab}(F^TK^{-1}F)^{-1} & 0 \\ 0 & 0 & 0 & \frac{-\Delta}{ab}(F^TK^{-1}F)^{-1} \end{bmatrix}$$

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & \frac{b}{\Delta}I & \frac{c}{\Delta}I \\ 0 & 0 & \frac{c}{\Delta}I & \frac{a}{\Delta}I \end{bmatrix}$$

$$\begin{bmatrix} I & 0 & \frac{-ab}{\Delta}K^{-1}F & \frac{cb}{\Delta}K^{-1}F \\ 0 & I & \frac{ca}{\Delta}K^{-1}F & \frac{-ab}{\Delta}K^{-1}F \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

which gives the identity matrix; then the inverse is

$$\begin{bmatrix} \frac{b}{\Delta}(K^{-1}-M) & \frac{-c}{\Delta}(K^{-1}-M) & \frac{1}{a}K^{-1}FD & 0 \\ \frac{-c}{\Delta}(K^{-1}-M) & \frac{a}{\Delta}(K^{-1}-M) & 0 & \frac{1}{b}K^{-1}FD \\ \frac{1}{a}(K^{-1}FD)^T & 0 & \frac{-1}{a}D & \frac{-c}{ab}D \\ 0 & \frac{1}{b}(K^{-1}FD)^T & \frac{-c}{ab}D & \frac{-1}{b}D, \end{bmatrix}$$

where

$$D = D^T \equiv (F^TK^{-1}F)^{-1}$$

and

$$M = M^T \equiv K^{-1}FDF^TK^{-1}$$

(both are nonnegative definite, at least).

Now consider the estimates, which is where this case gets interesting (or rather, uninteresting!):

$$\begin{bmatrix} \Gamma_{c1} & \gamma_2 \\ \gamma_1 & \Gamma_{c1} \\ \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{b}{\Delta}(K^{-1}-M) & \frac{-c}{\Delta}(K^{-1}-M) & \frac{1}{a}K^{-1}FD & 0 \\ \frac{-c}{\Delta}(K^{-1}-M) & \frac{a}{\Delta}(K^{-1}-M) & 0 & \frac{1}{b}K^{-1}FD \\ \frac{1}{a}(K^{-1}FD)^T & 0 & \frac{-1}{a}D & \frac{-c}{ab}D \\ 0 & \frac{1}{b}(K^{-1}FD)^T & \frac{-c}{ab}D & \frac{-1}{b}D \end{bmatrix} \begin{bmatrix} aK_0 & cK_0 \\ cK_0 & bK_0 \\ aF_0 & 0 \\ 0 & bF_0. \end{bmatrix}$$

This reduces to exactly the kriging weights: e.g.,

$$\begin{bmatrix} \Gamma_{c1} & \gamma_2 \end{bmatrix} = \begin{bmatrix} \frac{ab-c^2}{\Delta}(K^{-1}-M)K_0 + K^{-1}FDF_0 & \frac{-bc}{\Delta}(K^{-1}-M)(K_0-K_0) \end{bmatrix}$$

becomes

$$\begin{aligned} \begin{bmatrix} \Gamma_{c1} & \gamma_2 \end{bmatrix} &= \begin{bmatrix} (K^{-1}-M)K_0 + K^{-1}FDF_0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} K^{-1}((I-FDF^TK^{-1})K_0 + FDF_0) & 0 \end{bmatrix} \\ &= \begin{bmatrix} K^{-1}K_0 - K^{-1}FDF^TK^{-1}K_0 + K^{-1}FDF_0 & 0. \end{bmatrix} \end{aligned} \tag{4.2.16}$$

Note that the first term in the vector right-hand side: it is the kriging weight, which is seen by solving just one block of the cokriging system:

$$\Gamma_{c1} = K^{-1}K_0 - K^{-1}F\mu,$$

where

$$\mu = -D(F_0 - FK^{-1}K_0),$$

which gives a result identical to the first element of (4.2.16).

So: there is absolutely no change in the estimates by cokriging in this case, as the weights do not change. That is especially interesting and important because one method proposed for finding a valid model for the cross-variogram of two variables is to use a model which is a nested combination of models of the variograms: if the variograms have the same models (type and sill), however, the situation reduces directly to this case, and one sees immediately cokriging need not be used at all.

One can reach the same conclusion (with a lot less calculation!) via an argument about the form of the variogram matrix function. Recall (equation 3.2.7) that the variogram estimator can be written as

$$V^*(\underline{h}) = \frac{1}{2N_h}D^T(\underline{h})D(\underline{h}),$$

where D is the data set of paired differences. In this simple case that means that

$$\frac{1}{2N_h}D^T(\underline{h})D(\underline{h}) = \gamma(\underline{h}) \begin{bmatrix} a & c \\ c & b \end{bmatrix} = \gamma(\underline{h})Q\Lambda Q^T.$$

This says that by merely transforming the pair difference data via

$$D' = DQ$$

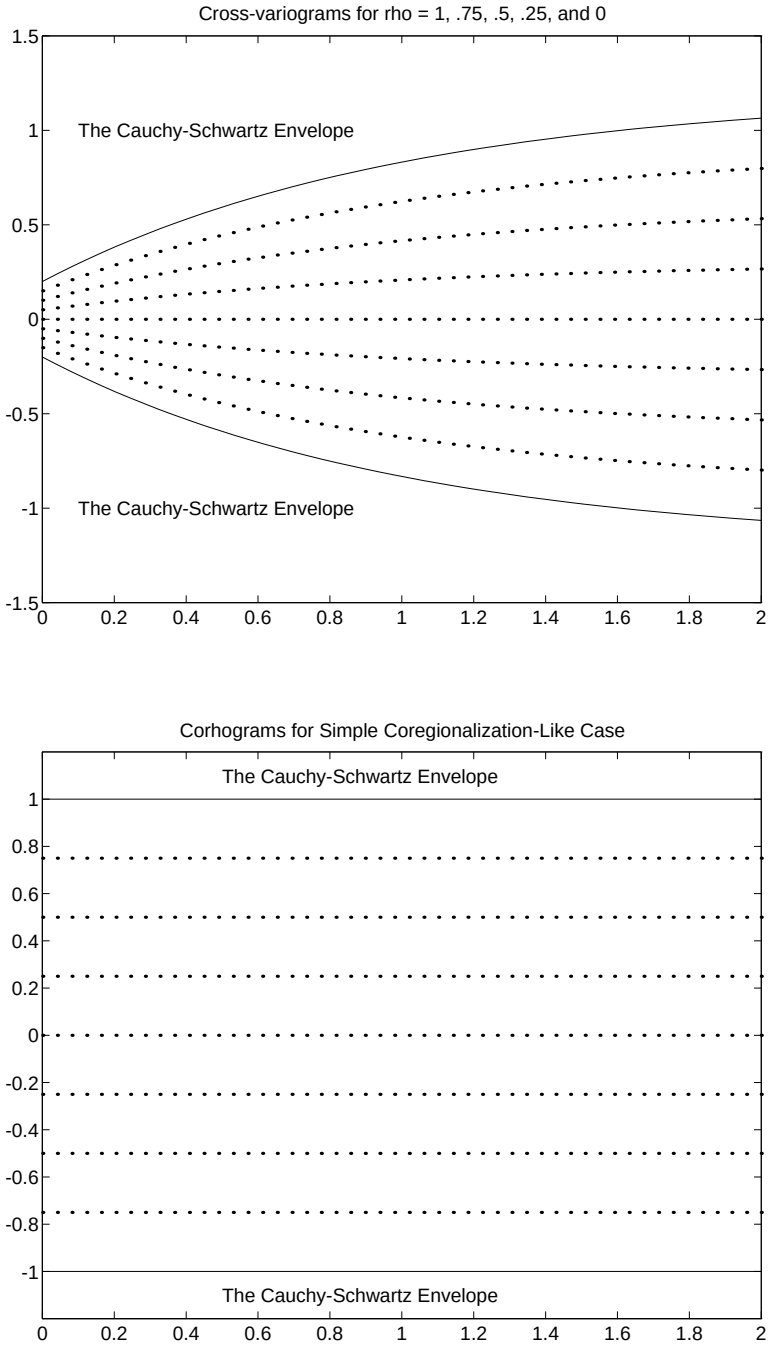


FIGURE 4.4. The dual ways of showing the information contained in the cross-variogram: against the product of the variograms, or scaled into the corhogram.

(which is equivalent to the same transformation on the original data, as the D are just linear combinations of the original data vectors), the sample variogram matrix will have been diagonalized: that is, the sample variogram matrix for the transformed data will have the form

$$V'^{*}(\underline{h}) = \gamma(\underline{h})\Lambda.$$

The corhogram for the simple coregionalization-like case is a constant: $\rho(h) = \frac{c}{ab}$. Figure (4.4) shows both the cross-variograms and and corhograms in this case, where the variogram is an exponential with nugget. It is obviously easier to focus on the corhograms, as they are simply constant.

The plots of the sample cross-variograms (and the models as well) must be within the Cauchy-Schwartz envelope to ensure invertibility, and so it is natural to represent it, too, on a picture of the cross-variogram model. This special case therefore presents a class of Cauchy-Schwartz envelope-filling cross-variograms which lead to absolutely no improvement over separate kriging. This motivates the question "What kinds of cross-variograms do suggest that cokriging will lead to an improvement over kriging results?" This remains a topic for further research.

We do not yet even know if there is any improvement in the case where $\rho(h)$ is a constant, but the variogram is not the same for all variables, as it was in the elemental coregionalization-like case. In this case,

$$V(\underline{h}) = \begin{bmatrix} \gamma_1(\underline{h}) & c\sqrt{\gamma_1(\underline{h})\gamma_2(\underline{h})} \\ c\sqrt{\gamma_1(\underline{h})\gamma_2(\underline{h})} & \gamma_2(\underline{h}) \end{bmatrix}. \tag{4.2.17}$$

The difference lies in the fact that the variogram matrix function is no longer so simply diagonalizable. Even so, it may still be written in the simplified form

$$V(\underline{h}) = \begin{bmatrix} \sqrt{\gamma_1(\underline{h})} & 0 \\ 0 & \sqrt{\gamma_2(\underline{h})} \end{bmatrix} \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix} \begin{bmatrix} \sqrt{\gamma_1(\underline{h})} & 0 \\ 0 & \sqrt{\gamma_2(\underline{h})} \end{bmatrix}.$$

This suggests a possible transformation of the paired difference data via the inverse square-root of the respective variogram inverses; again, we have not pursued this transformation.

Wackernagel suggests, in [93], that the corhogram is not the statistic that one should inspect, but rather what he called the "autokrigeability coefficients"

$$ac_{ij} = \frac{\gamma_{ij}}{\gamma_i}$$

and

$$ac_{ji} = \frac{\gamma_{ij}}{\gamma_j}.$$

This is based, however, on his attempt to demonstrate intrinsic coregionalization, since this implies that cokriging reduces to kriging. It is unknown, however, whether intrinsic coregionalization is a necessary condition for "autokrigeability" or not.

Oddly enough, Some authors [38] report differences in estimates in the case of an intrinsically coregionalized cokriging, citing similar claims in the Summer 1992 issue of *Geostatistics: An Interdisciplinary Geostatistics Newsletter*.

Generalization

The new formulation of the cokriging equations generalizes, but not elegantly. For example, in the three variable case, one may permute as before and multiply through by the kriging system matrix inverses to get

$$\begin{bmatrix} I & A_{12} & A_{13} \\ A_{21} & I & A_{23} \\ A_{31} & A_{32} & I \end{bmatrix}$$

Multiplying through by the inverse in the first two variables, as given above,

$$\begin{bmatrix} B_1 & -B_1A_{12} & 0 \\ -B_2A_{21} & B_2 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} I & A_{12} & A_{13} \\ A_{21} & I & A_{23} \\ A_{31} & A_{32} & I \end{bmatrix} = \begin{bmatrix} I & 0 & \alpha_1 \\ 0 & I & \alpha_2 \\ A_{31} & A_{32} & I \end{bmatrix}$$

followed by

$$\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -A_{31} & -A_{32} & I \end{bmatrix} \begin{bmatrix} I & 0 & \alpha_1 \\ 0 & I & \alpha_2 \\ A_{31} & A_{32} & I \end{bmatrix} = \begin{bmatrix} I & 0 & \alpha_1 \\ 0 & I & \alpha_2 \\ 0 & 0 & I - A_{31}\alpha_1 - A_{32}\alpha_2 \end{bmatrix}$$

so that one must invert the matrix $I - A_{31}\alpha_1 - A_{32}\alpha_2$, which is again $(N+p)\times(N+p)$.

Induction on this process implies that in each case one gets the solution of a single chain $(1, \cdots, p-1, p\text{-way})$ of cokriging equations, from the kriging case all the way up to the p-way case, each of which will usually be of interest.

Discussion

Below we list the positive features of this new formulation:

- simultaneous kriging estimates,
- a chain of (sub-)cokriging estimates,
- ease of comparison of different cross-variogram models,
- smaller systems of equations, and
- better conditioned matrices.

We have not yet realized the goal described in the introduction, of a method for the solution of a p-way cokriging system which gives the results of all $p-1, p-2, \cdots, 1$-way cokriging sub-systems: however, a single chain of such cokriging solutions is obtained.

If the goal were, say, the estimation of the concentration of nitrates, and there were k other variables which one suspected might help improve the estimates of nitrate via cokriging, one could order them as $n_1, n_2, \ldots, n_k$ and cokrige with this method so as to get the results of

- kriging for nitrate (as well as kriging for each n_i);
- cokriging for nitrate with n_1;
- cokriging for nitrate with n_1, and n_2;
- $\vdots$
- and cokriging for nitrate with $n_1, n_2, \ldots,$ and n_k.

One topic for the near future will be to rework the results of Dubrule [23], who discovered a very clever way of cross-validating the results of universal cokriging using components of the inverse. As the inverse of Myers's system is no longer computed, Dubrule's results must be rewritten in terms of the components of the matrices used in this formulation.

A Word on Kriging Matrix Conditioning

We wish to make one further point, concerning the solution of cokriging equations in general: that it is very important to take the matrix solver into account, as it is perhaps the most important single element in the cokriging process. Carr and Myers [12] discussed different equation solvers for cokriging programs, and decided (at that point) on Gaussian elimination. In their first cokriging code, Carr, Myers, and Glass [13] used a slower, iterative, algorithm which minimized memory use. The program "cokrige", which was developed by the Geostatistics Group of the Mathematics Department, University of Arizona, for adaptation into the Geo-EAS pantheon of programs (but never formally approved) incorporated Gaussian elimination.

Early in the Nitrate study we encountered trouble when generating maps with "cokrige": it seems that we were using "too many variables" and/or "too many sites", which led to estimates which were obviously very poor (e.g., estimates orders of magnitudes higher or lower than any data values); yet there was no indication from the program of any problem. Gaussian elimination was improperly chosen as the solution algorithm, at least in that case, because of the danger posed by both the size and the conditioning of the matrices to be inverted. The modelling process is still poorly understood, and the risk of creating large ill-conditioned matrices is sufficiently high that we were inspired to write a new program, choosing another algorithm, and a safer algorithm, for the matrix inversion: the SVD, in double precision.

McCarn and Carr [64] compare gaussian elimination, LU decomposition, and, to a lesser extent, the SVD, in the computation of kriging weights, as well as the effect

Gaussian Elimination	LU Decomposition	SV Decomposition
ops $= \frac{2N^3}{3}$	$\frac{N^3}{3} \leq$ ops $\leq \frac{2N^3}{3}$	ops $> N^3$

TABLE 4.1. Operation counts for different equation solvers (from McCarn and Carr [64])

of numerical precision used and the advantages of iterative improvement. They give the number of operations for the three methods (Table (4.2.2)). They also discuss the value of using only a small number of neighbors, to reduce round-off error, suggesting 10-20 neighbors for local neighborhoods.

They note that the SVD gives results identical to those using gaussian elimination or LU decomposition in the case of ordinary kriging, but state cryptically that "for universal kriging...there is a large difference in the solution yielded by SVD from that yielded by either Gauss elimination or LU decomposition." They do not go on to explain why the methods gave different results, or tell which "solution" is better.

One possibility is that the functions used to model the drift were not scaled properly. Note that the two parts of the cokriging matrix in (4.2.11),

$$\begin{bmatrix} V & F \\ F^T & 0 \end{bmatrix},$$

are independent: scaling the variables related to V does not affect F, and *vice versa*. If rows and columns corresponding to F, say, are allowed to get much larger than the V portion of the matrix, the condition number will increase artificially (in the sense that scaling would have prevented any problems). This could happen if the functions used were simple monomials (like xy), and the geographical coordinates were orders of magnitude larger than the variogram values contained in V. Programmers must check that the kriging system is balanced, before a solution is attempted. Software used in this dissertation scales the variables, so that the drift functions and the variogram matrix values are on the same order.

This scaling problem is the same as that found by O'Dowd [71], when he reported that the condition number of the ordinary kriging system went up with a linear increase in the sills of the variogram models. This is simply a result of having a column (and row) of fixed values (ones) in the F portion of (4.2.11), while the V portion is scaled linearly. Poor conditioning in this case is not a fundamental characteristic of the kriging system, as it can be removed by scaling.

One advantage of using the SVD as a solver is that the condition number of the coefficient matrix shows up immediately as the ratio of the largest and smallest singular values: if A is $N \times N$, then

$$\text{condition(A)} = \begin{cases} \frac{\lambda_1}{\lambda_N} & , \lambda_N \neq 0; \\ \infty & , \lambda_N = 0 \end{cases}$$

This number should be reported, especially when it is high, since it serves as a handy diagnostic to indicate whether the results will be useful or not. If the coefficient matrix is non-invertible to machine precision, then the option should be given to proceed with the pseudo-inverse (which is obtained from the SVD, and leads to a least-squares solution for the projection of the right-hand side onto the residual column space of the matrix).

In fairness to the developers of the algorithm which failed, and failed to warn the user, at the time that the program was developed storage and speed considerations were much more important than they are in today's workstation environment; and the Gaussian method requires less memory, and is faster. But once again, we see that with a change in computational power it may be necessary to rethink some of the early procedures.

Chapter 5

Kernels and Kriging: In Search of a Compromise

5.1 Introduction and Motivation

Concerns about the large linear systems involved in unique-neighborhood (i.e., global) kriging and the instabilities in their solutions has motivated a search for better methods. It is certainly an unhappy fact that the more data available for interpolation, the more dangerous the method may become (as will be seen shortly): this is fundamentally contrary to what one expects from an interpolation scheme.

In the course of studying the solutions of the kriging system of equations (given by (4.2.11)), we discovered that the data weights obtained as the solution had strong spatial properties: they certainly satisfied the intuitive sense that greater weight should be given to close neighbors, with weight falling off as distances from the estimation site become large. The form of the kriging weights was striking, however (Figures (5.1) and (5.2) for one-dimensional kriging weights, and (5.3) for two-dimensional weights): the resemblance that these (typical) weight distributions bear to weights given by kernels led us to begin seek equivalent kernels appropriate for a variety of variogram models.

As noted in the last chapter, kernel methods are generally faster and more stable than kriging. Thus, identifying kernels that give good approximations to the kriging weights would allow approximations of the results of kriging in some cases, leading to an increase in speed and stability at the cost of a loss of some of kriging's optimality. The most serious objection to kernel estimators is that they are unmotivated: why a factor of "2" (for example) in inverse distance weighting schemes? The use of a kernel associated with a variogram model, would be a significant improvement in the use of these techniques.

Kriging and kernels have been seen as estimation opponents in some work: Yakowitz and Szidarovszky [99] compared kriging (with correct and misspecified variogram models) with a gaussian smoothing kernel with bandwidth corrections. Using several test data sets, they concluded that kriging gave better estimates when the variogram model was correctly specified, but that the kernel did a better job when the variogram model was inappropriate.

Moreover, as noted in Carr and Myers [11], kriging on gridded data with local neighborhoods is actually equivalent to estimating with a kernel under some conditions: the weights are computed once (for those points away from the boundary) and

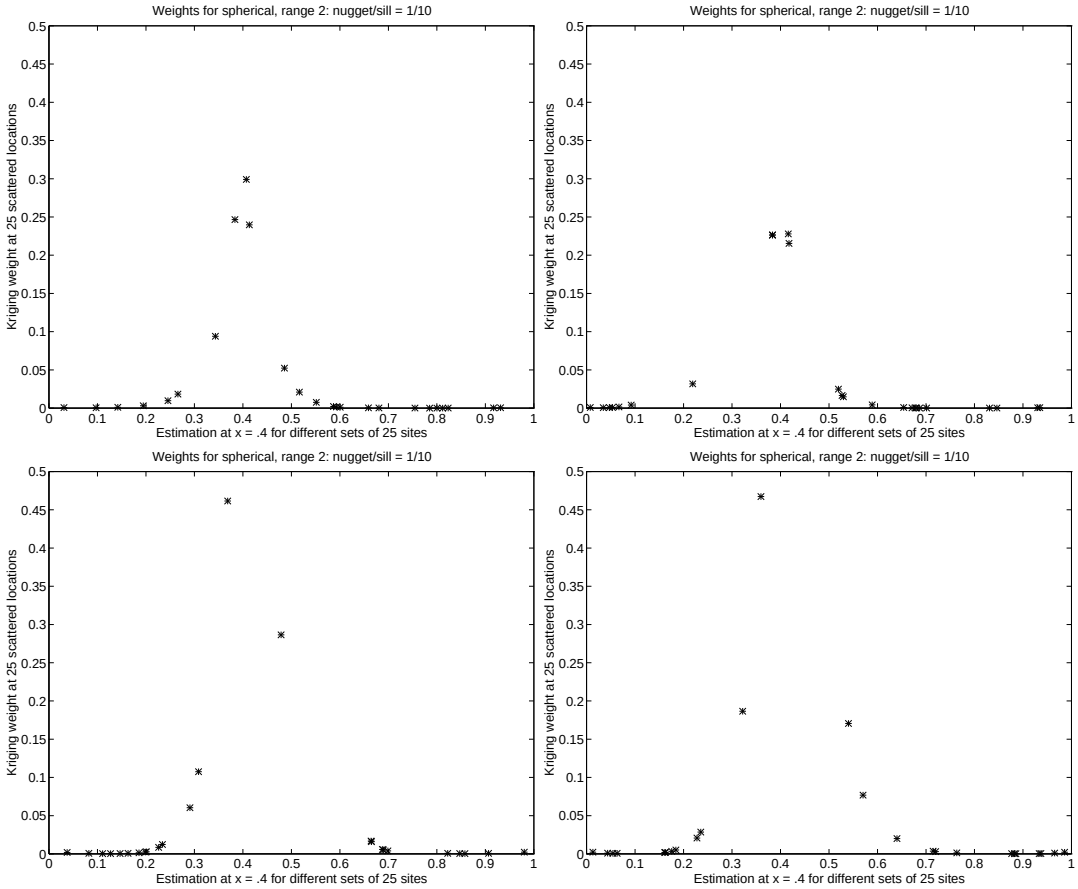


FIGURE 5.1. Four kriging weight patterns in the one-dimensional case, using 25 scattered data locations on the interval [0,1]. Estimation at x=.4 with four different sets of locations.

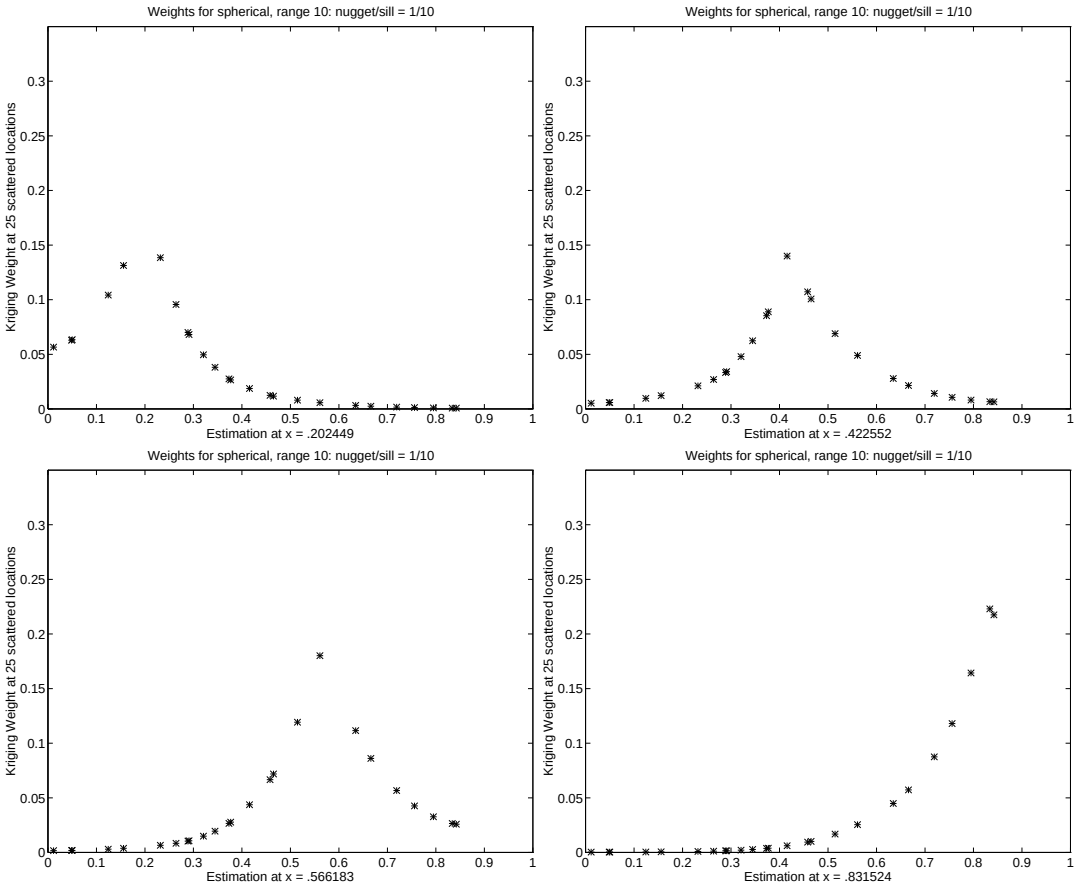


FIGURE 5.2. Four kriging weight patterns in the one-dimensional case, using 25 scattered data locations on the interval [0,1]. Estimation using fixed set of data locations, at four different points on the unit interval.

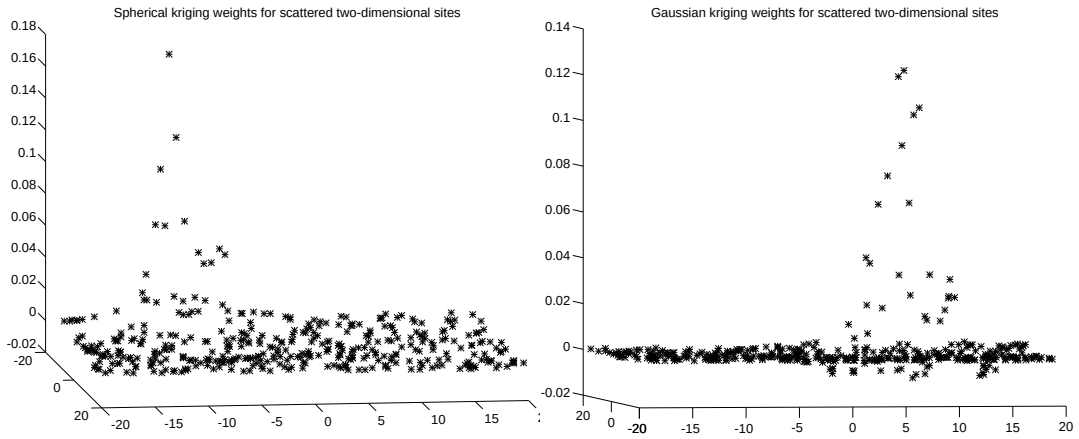


FIGURE 5.3. Kriging weights for two different models in the two-dimensional case, for scattered sites. The resemblance these (typical) weight distributions bear to weights given by kernels suggested that there might be equivalent kernels appropriate for a variety of variogram models.

the weights obtained are used as a kernel for all points in the interior. This relies on the data locations always being in the same pattern, and the estimation location being at the same position with respect to that pattern every time.

It is known that spline interpolation is a special case of kriging [96]. The spline minimizes the functional

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - g(t_i))^2 + \lambda \int_0^1 g''(t)^2 dt,$$

where N (typically noisy) observations y_i occur on the interval $[0,1]$. Furthermore, extensive work by a number of investigators has shown that, for the one-dimensional smoothing spline, there is a well-defined kernel which can be used in place of the cubic splines. Silverman [82] showed that the form of the weight function in smoothing spline estimation is

$$G(s,t) \approx \frac{1}{f(t)}\frac{1}{h(t)}\kappa(\frac{s-t}{h(t)}), \tag{5.1.1}$$

where $f(t)$ is the design point density function, $h(t) = \lambda^{\frac{1}{4}} f(t)^{-\frac{1}{4}}$, λ is the spline smoothing parameter, and the kernel κ is defined by

$$\kappa(u) \equiv \frac{1}{2}\exp(-\frac{|u|}{\sqrt{2}})\sin(\frac{|u|}{\sqrt{2}} + \frac{\pi}{4}). \tag{5.1.2}$$

This kernel leads to an estimate at site s of

$$g^*(s) = \frac{1}{n}\sum_{i=1}^{n} G(s,t_i)y_i.$$

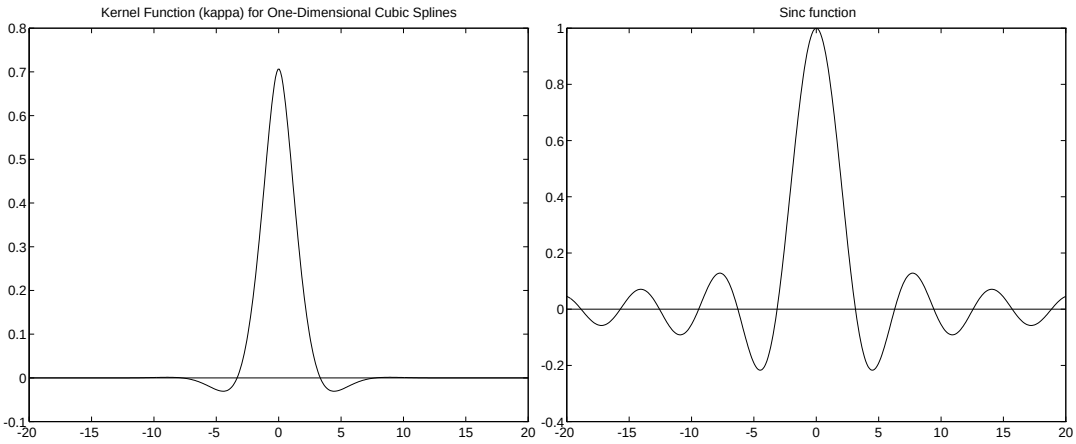


FIGURE 5.4. Silverman's kernel function for the smoothing spline (left) looks like an attenuated sinc function (5.1.3), right.

He showed that the kernel performed poorly near the boundary, and developed a correction using points reflected out beyond the boundary. Messer [65] examined bounds on the differences of the two, and concluded that the approximation is excellent in general: good enough that one could use it in place of the spline. We say that the kernel κ (5.1.2) is associated with the smoothing spline model.

Note that the weight function is essentially the kernel κ multiplied by certain corrections related to bandwidth, and λ (which is also related to smoothing). Silverman's κ looks something like a sinc function (Figure (5.4)); that is,

$$w(x) \approx \frac{sin(x)}{x}. \tag{5.1.3}$$

The sinc function is a popular choice as a smoothing kernel for image analysis.

Note also that the sum of the weights is not divided out as it was in the initial description of kernel methods (equation (4.1.1)). Chu and Marron [14] discuss the comparative value of using T versus some integral over a domain surrounding s (as Silverman has). They conclude that from an analysis standpoint, the integral is easier; but that results they get using the two different methods suggest that using T is best.

The smoothing spline does not coincide with kriging as presented in the previous chapter, but rather with a modified version of kriging equations:

$$\begin{bmatrix} V + \sigma^2 I & F \\ F^T & 0 \end{bmatrix} \begin{bmatrix} \Gamma \\ \mu \end{bmatrix} = \begin{bmatrix} V_0 \\ F_0 \end{bmatrix}.$$

The difference lies in the addition of $\sigma^2 I$ to the variogram matrix V: this results in estimation, rather than interpolation, and corresponds to an assumption that the noise in the data is uncorrelated white noise of variance σ^2.

The inspiration provided by the form of the weight vector solutions of the kriging equations, by Silverman's result in the case of the smoothing spline, and also by obvious cases where the kriging system for certain models give results equivalent to those of kernels, provoked the search for a way of approximating the results of kriging by kernels. The dual form of the kriging equations show that the variogram model defines a kernel estimator, only using the set of transformed data (which must be obtained from the large linear system in a global kriging scheme (equation (4.2.7))): that is,

$$y^*(s) = \sum_{i=1}^{n} \gamma(s, t_i)\hat{y}_i + \mu.$$

To what extent can a variogram model be said to generate a kernel estimator for the data, rather than the transformed data? That is, the extent to which variogram γ can be associated with kernel κ_γ such that

$$y^*(s) = \sum_{i=1}^{n} \kappa_\gamma(s, t_i)y_i.$$

Note that the words "kernel" and "weight function" are being used interchangeably now: since the bandwidth corrections like those which Silverman made will not be discussed, no effort to distinguish between the two will be made. As refinements to the kernels discussed herein develop in the future, however, the distinction will have to be made.

A variogram model determines the coefficient matrix of a kriging system (equation (4.2.11), for example). (In the following, the ordinary kriging system will generally be used for figures and to make derivations.) One inverts the coefficient matrix to obtain a weight vector (and lagrange multiplier); the weights are multiplied with the data to produce an estimate (4.2.6). If a function κ exists, which always reproduces the kriging weights yet which is only a function of the relative positions of the estimation site x_0 and the data locations, then the equivalence is exact. In general, however, one does not expect exact equivalence, but only to approximate the weights sufficiently well using the kernel that one can make use of it when time is more important than optimal accuracy. Such a kernel will be called an "apparent (or effective) kernel function", and is obviously not unique.

It seems that the data weight vectors obtained from the kriging equations for a variety of models (when represented in the coordinate space) have the appearance of a kernel: that is, have a form suggestive of the existence of some underlying kernel function of the sort Silverman discovered for the smoothing spline. It is in this sense that a kernel is (or may be) associated with the variogram. Kernels appropriate for each variogram considered are then described.

There are not characterizations as nice as (5.1.2) for variogram models in general: that is, there are not nice analytical expressions for any models other than the most

trivial ones. However, we have been able to experimentally examine the forms of the apparent kernels for the standard models, and communicate the results in this chapter. We also demonstrate one attack on the kernel problem, via a trade off between the matrix equations and integral equations.

Oddly enough, our search for kernel approximations to kriging began in the multivariate case: that is, in the case of cokriging. Initial kernels centered squarely on variogram matrix model. One experiment entailed using the following multivariate kernel estimator:

$$z^*(x_0) = (\sum_{i=1}^{N} \Gamma^{-1}(x_i - x_0))^{-1} \sum_{i=1}^{N} \Gamma^{-1}(x_i - x_0) z_i; \tag{5.1.4}$$

and, in view of the variogram matrix results from coregionalization, i.e. that for variables with constant corhograms the variogram matrix takes the form

$$\Gamma(h) = \text{diag}(\Gamma(\text{h}))^{-\frac{1}{2}} \text{Cdiag}(\Gamma(\text{h}))^{-\frac{1}{2}}$$

with C a constant positive definite matrix (4.2.17), the square root of the variogram matrix was also used as a kernel. Results with these kernels were unimpressive, however: cross-validation showed that they did not even stand up well against inverse square distance weighting. There is also the potential for non-invertibility of the model matrices and their sums in (5.1.4), and these methods are almost as *ad hoc* as the inverse distance weighting methods.

Kernels also ignore the sampling pattern, which is one of the motivations for using the kriging methodology: kriging uses the sample pattern information to deduce when weights should be down- or up-graded because of redundancy (or lack thereof). Kriging takes account of the sampling by comparing each data site with every other data site in a large matrix, which must be inverted to obtain the estimates. As will be shown, large ill-conditioned matrices may lead to poor results.

The focus in the following discussion will be on the univariate case, i.e., kriging, and mostly in one-dimension. We will show that the kriging equations also determine weights which appear to be, at least to a fairly good approximation, representative of some apparent kernel function, whose shape is not necessarily what one would predict from the variogram model, or its inverse. The kernels exhibit characteristics consistent with the kernel that Silverman found for the splines, and other well-known and intuitive functions.

We have not yet returned (in a more serious way) to the equally important problem of multivariate kernels, and kernel approximations to cokriging.

5.2 Shadow Effect? What Shadow Effect?

The claim was once made, in a meeting of The Geostatistics group, University of Arizona, that the one-dimensional kriging problem possesses the property that

points beyond the very nearest are effectively "shadowed", or "screened" (see Journel and Huijbregts[1] [48]) by closer data sites, and hence do not contribute much to the estimation of a given point. The implication was that while this is true in one-d, it is qualitatively different from what one finds in higher dimensions. From observations of actual one- and two-dimensional kriging weights, we present several dissenting conclusions and discoveries.

The first is that the notion of a shadow effect is more distracting than enlightening: while there may be some "shadowing", particularly when there is no nugget, this is merely a restatement of the obvious intuitive notion that nearer points get more weight than far points, and "near" in the shadowing cases is "very near", while "much" is "very much". The shadow effect is not an important separate phenomenon, but rather a simple limiting case.

The second discovery, more empirical than theoretical, is that the kriging system appears to give rise to what seems to be a fixed kernel function: that is, that the variogram determines a kernel estimator to fairly good approximation.

The simplest case is the nugget model: its kernel is obviously just a constant (for all locations away from the data sites): at the sites, the kernel becomes a delta function. One can verify that the kriging equations in the case where $x_0 \neq x_i$ yield constant weights $\underline{w} = \frac{1}{N}\underline{1}$, and $\mu = \frac{n}{N}$: i.e., that

$$\begin{bmatrix} n(\underline{1}\underline{1}^T - I) & \underline{1} \\ \underline{1}^T & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{N}\underline{1} \\ \frac{n}{N} \end{bmatrix} = \begin{bmatrix} n\underline{1} \\ 1 \end{bmatrix}.$$

The case $x_0 = x_i$ is equally obvious. In fact, as the ratio of nugget to total sill goes to 1, the kernel of a variogram model becomes a constant function irrespective of model type.

Another simple case, and one which may have helped inspire the shadow effect notion, corresponds to the linear model in one dimension. As the dual form of the kriging equations for this model shows, estimates obtained from kriging with a linear model are obtained by simply "connecting-the-dots" between data values at data locations. The linear variogram therefore gives rise to the following kernel (in the absence of a nugget):

$$\kappa(x) = \frac{(x - x_i)z_i + (x_{i+1} - x)z_{i+1}}{x_{i+1} - x_i},$$

where x_i and x_{i+1} are the two closest neighbors, "shadowing" out all the others. This is the essence of the shadow effect: that only the very nearest values are needed, as others are screened out.

The third conclusion is that the behavior of the kernel function associated with a model in the neighborhood of the location of the estimation site is related to,

[1] "The close data ... screen the influence of the more distant data....", p. 312

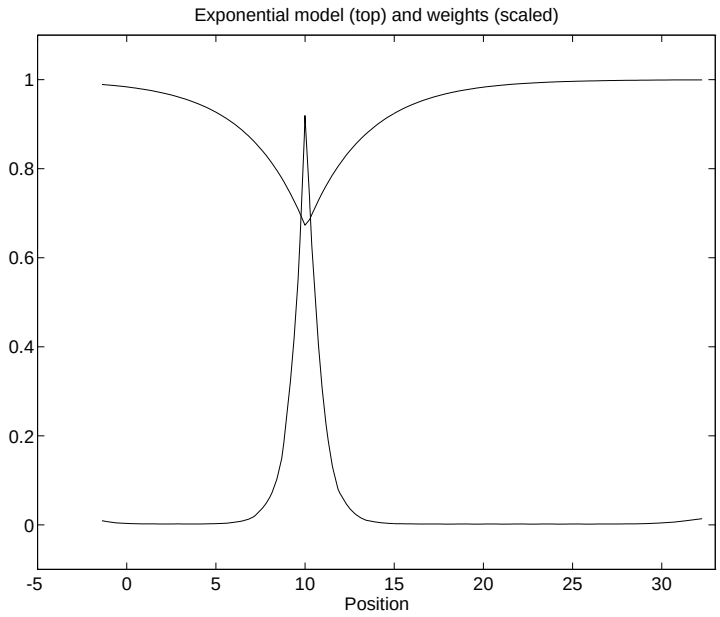


FIGURE 5.5. The model and kernel seem to mimic each other in this exponential variogram interpolation pattern. The model above, and the kriging weights below, are referenced to an estimation site around 10.

and perhaps determined by, the concavity of the variogram: those models which are concave down at the outset have weights which tend to fall off rapidly, whereas those which are concave up tend to spread the weight around (see Figure (5.5) for an example). This makes sense from the standpoint of the differentiability of the estimator: the kriging estimator is as differentiable as the variogram model; if the true variogram model is infinitely differentiable, then the underlying phenomenon must be smooth, in which case it makes sense to give high weight to many nearby and even fairly distant neighbors when estimating.

We present numerous examples to illustrate these points, beginning with one-dimensional problems, then considering the two-dimensional case.

5.2.1 Variogram Models That Are Concave Up

■ Example 1: The Cosine Model

One variogram model which starts out concave up is the periodic cosine model. The cosine model is not strictly conditionally negative definite, and so is not considered one of the standard models. It can be used in conjunction with other models, however, in nested models, and is overlooked as a model, especially when periodic phenomena are studied (i.e. time series of environmental variables). Data sampled from the cosine function realization gives rise to a cosine variogram, as seen by taking

the (spatial) limit

$$\gamma(h) = \lim_{h \to 0} \frac{1}{2(R-h)} \int_0^{R-h} [cos(x+h) - cos(x)]^2 dx = 1 - cos(h).$$

As was pointed out previously (equation (4.2.6)), the kriging estimate can be written as

$$z^*(x) = \left[\begin{array}{cc} \underline{z}^T & 0 \end{array} \right] K^{-1} \underline{v}(x) = \underline{w}^T \underline{v}(x),$$

where K represents the kriging coefficient matrix, $\underline{v}(x)$ represents the RHS of the kriging system, and $\underline{w}$ represents the solution (weight) vector

$$\underline{w} = K^{-1^T} \left[\begin{array}{c} \underline{z} \\ 0 \end{array} \right],$$

in which case the emphasis is on the variogram function (the constant weights w multiply the variogram values; the variogram serves as a kernel), or as

$$z^*(x) = \left[\begin{array}{cc} \underline{z}^T & 0 \end{array} \right] K^{-1} \underline{v}(x) = \left[\begin{array}{cc} \underline{z}^T & 0 \end{array} \right] \underline{w}(x) = \underline{w}^T(x) \left[\begin{array}{c} \underline{z} \\ 0 \end{array} \right],$$

where

$$\underline{w}(x) = K^{-1} \underline{v}(x), \tag{5.2.5}$$

in which case the weights w are functions of x, but weight the data values z directly.

From the standpoint of computational efficiency, the first case above is better: the estimator is a function (the variogram model) which one merely computes relative to N data sites, forming a vector, and multiplying with the constant weights w; but for purposes of intuition, the second might be preferred. The magnitude of the function value $w(x_i)$ tells (to some extent, anyway) the amount of confidence a particular site x_i merits, while the sign tells whether x_i is correlated positively or negatively with x, the location of interest. It is the second form of the weights, weights of data values, that we seek to relate to kernel estimators.

Two sets of one-dimensional "design points" $\{x_i\}$ were generated, in two different ways: in one, the x_i were generated randomly and uniformly; in the second, they were generated as a set of shifted grids. The data values were given by $y_i = cos(x_i)$, the theoretical variogram obtained above was used as the input to a kriging program, and the estimates and weights were computed for a grid (using ordinary kriging).

Figure (5.6) shows the estimates of the cosine function at a spot for which there is little data: the goal was to estimate the function at a peak, after removing nearby data points. If there were a shadow effect, then a missing peak could never be reconstructed by its neighbors.

As one can see, however, the "missing hump" in the data was adequately recomputed, from which one can conclude that the cosine model does not exhibit the

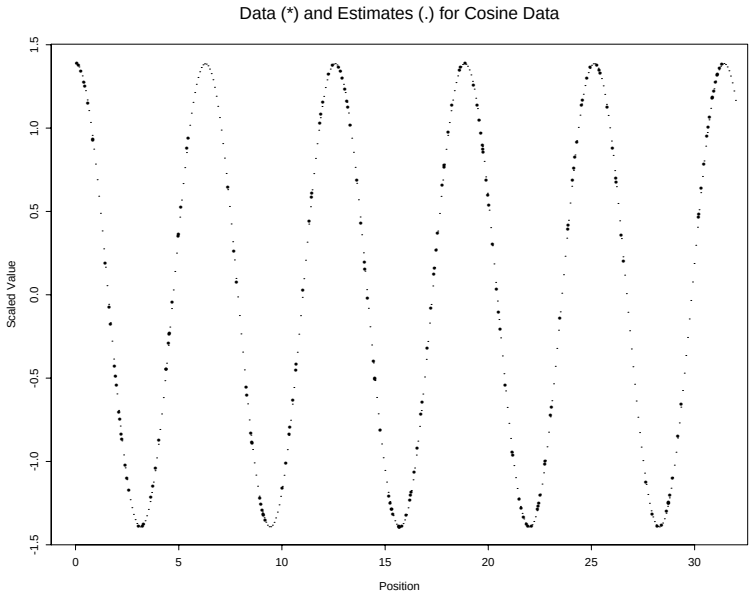


FIGURE 5.6. Cosine reconstructed from scattered samples: note in particular the hump on the left side, which was well reconstructed in spite of the lack of elevated neighbors.

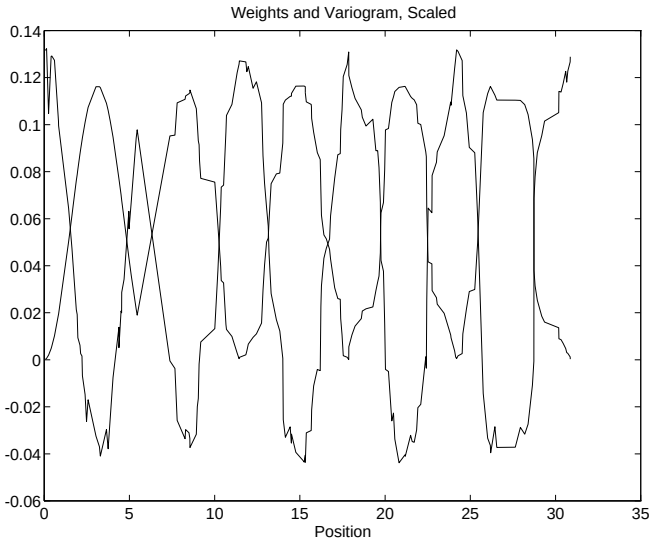


FIGURE 5.7. Cosine data weights and variogram: one and the same?

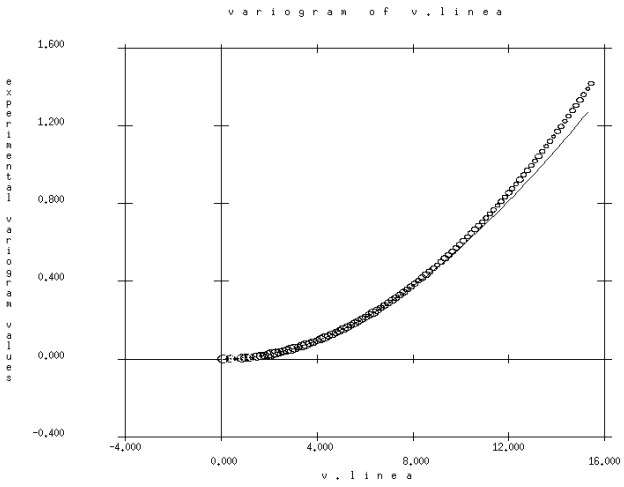


FIGURE 5.8. The Variogram of data set linear.dat modelled by a long-range gaussian (long with respect to the pair distances).

shadow effect in general. Figure (5.7) shows that the data weights are also periodic, and are inversely and well-correlated (-0.99386) with the variogram. That is, when the variogram is low, the weights are high; while if the variogram is high, the weights are low.

The cosine variogram is unusual, in that it is not strictly conditionally negative definite. But the same phenomenon is illustrated (albeit not so dramatically) by other variogram models.

The cosine model has another curious property: kriging weights obtained using the cosine model proved to be remarkably stable, in spite of an increasing nugget. The weights barely change as the nugget goes from zero to 200% of the sill of the cosine.

■ Example 2: The Gaussian Model

The gaussian model is another which fails to show a shadow effect. Data sampled from a linear function gives rise to quadratic growth, which can be well-modelled by a gaussian variogram, as shown in Figure (5.8). A linear function was sampled, its variogram modelled with a gaussian, and the same procedure was carried out on the data set linear.dat.

In order to explore how changes in model parameters changed the apparent kernel function underlying a gaussian model, the nugget was varied relative to the sill, the range was varied, and the estimate was made at different locations among the data sites (recall that Silverman's G (5.1.1) contains information about the density of

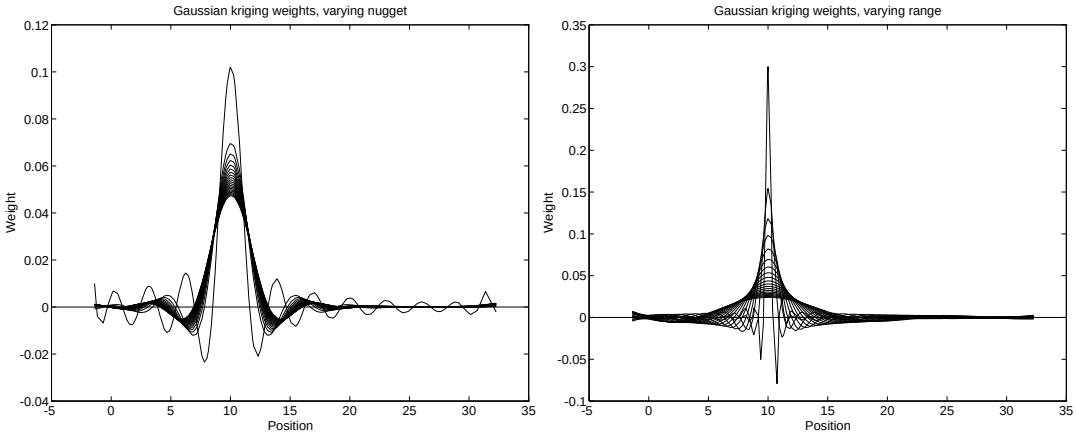


FIGURE 5.9. The weights for the gaussian model suggest a kernel function which resembles the sinc. Left: nugget variation; the highest weight drops steadily as the nugget percentage increases. Right: range variation; the weights steadily spread out as the range increases.

design points, as this is also important when deciding how to weight a region using a kernel). Changes in the form of the apparent kernel function were examined as these parameters varied.

The kriging weights obtained using the gaussian model, even at zero nugget, show no tendency to shadow each other out (Figure (5.9)). The effect of varying the nugget is to dampen the oscillations that occur, and to spread out the first peak so as to give larger weights to more neighbors.

Note the oscillation attenuation that occurs when a nugget is added to a gaussian (Figure (5.9)). The nugget damps those, and spreads out the central peak of the gaussian. This is similar to the effect seen in the second plot, for the change in range. Note also that the apparent kernel function is reminiscent of a sinc function. While this sort of function does falls off, it is far from negligible at the second hump.

As the position at which the estimate is taken varied (and hence the distribution of points in the neighborhood: Figure (5.10)), notice first how very similar the form of the weights is, and then, in particular, how the weights are distorted near the boundary. The latter is much like the boundary distortion that Silverman corrected in the spline kernel case.

One note about the smoothing spline model: the shape of Silverman's kernel function κ suggests that its "equivalent variogram" would be concave up. There is no variogram to plot in this case: the cubics are generalized covariances, rather than variograms. The estimator in the spline case is of the same form as the kriging estimator, justifying calling it an example of kriging (again, from Watson [96]):

$$f(x) = \underline{d}^T G^{-1} g(x),$$

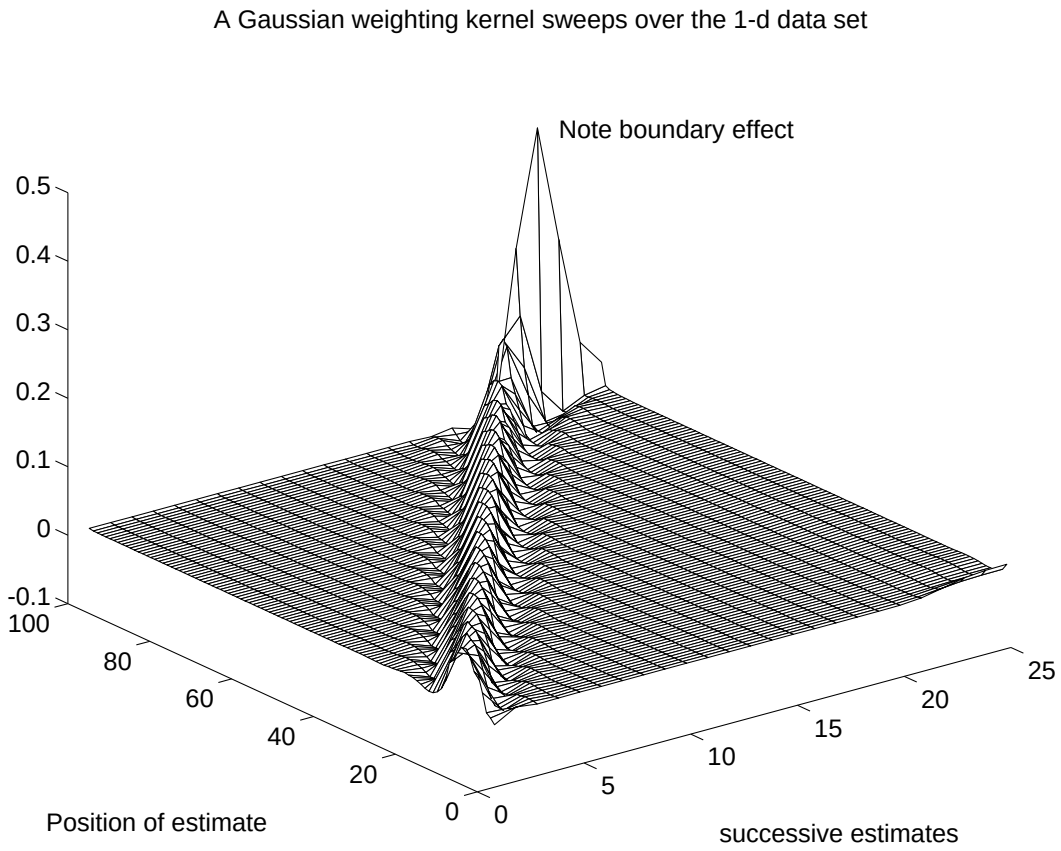


FIGURE 5.10. Notice the boundary effect, which is very similar to that found by Silverman, which he corrected using reflection.

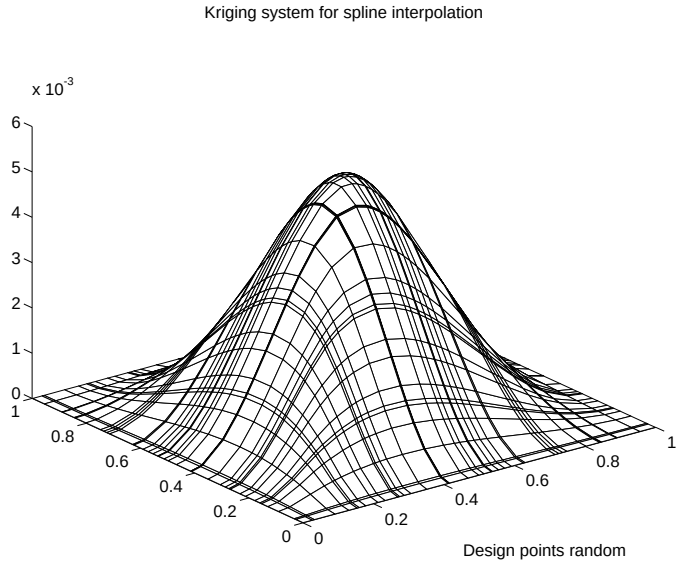


FIGURE 5.11. The portion of the spline coefficient matrix corresponding to the variogram portion of the kriging system.

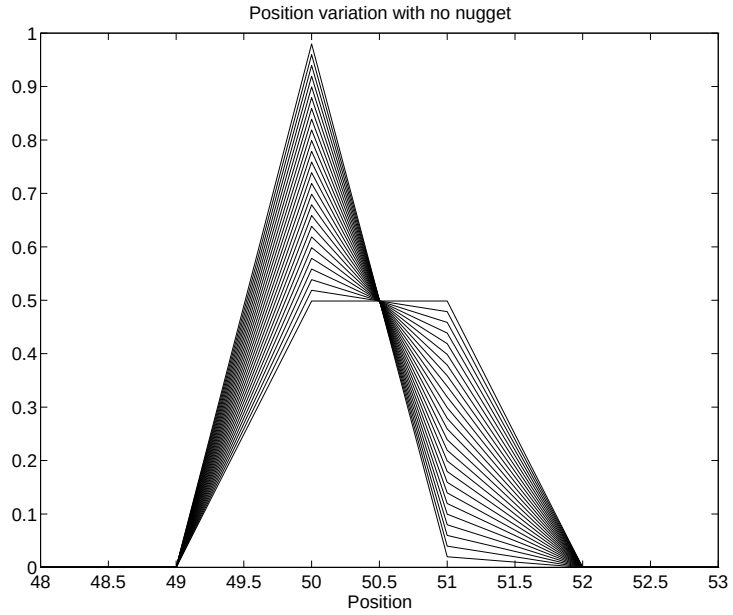


FIGURE 5.12. The kriging weights in this exponential model without nugget are effectively non-negative on only the two closest neighbors, at 50 and 51. Shown are the weights as estimates occur at a succession of values from 50 to 50.5.

where the matrix G is given component-wise by cubic spline terms of the form

$$g(x; x_i) = \frac{1}{6}H(x - x_i)(x - x_i)^3 - \frac{1}{6}x^3(1 - x_i)^2(1 + 2x_i) + \frac{1}{2}x^2(1 - x_i)^2 x_i,$$

and H is the heaviside function. It is interesting to see what the kriging matrix looks like for the spline functions (Figure (5.11), with 40 design points randomly distributed on the interval [0,1]).

5.2.2 Variogram Models That Are Concave Down

The data weights obtained using an exponential model without a nugget show the shadow effect. In fact, they appear to behave the same as the weights in the linear model case, as seen in Figure (5.12). Figure (5.13) shows the dramatic change that the weights undergo with the addition of even a small nugget (the first change represents a nugget/sill ratio of .1).

Figure (5.5) compares the model to the weights. Certainly the variogram model resembles the form of the weights, which suggests the reasoning behind efforts to relate the two (as in 5.1.4).

The results for the case of the spherical model are similar enough that they do not bear repeating. However, the spherical model leads to more high frequency oscillations

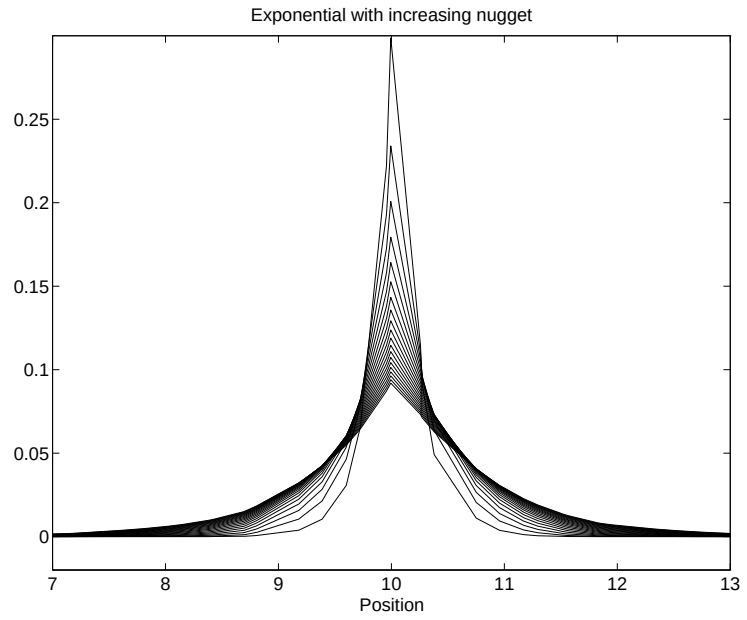


FIGURE 5.13. Once a nugget is added, the exponential weights begins to look resemble a pointy but smooth kernel.

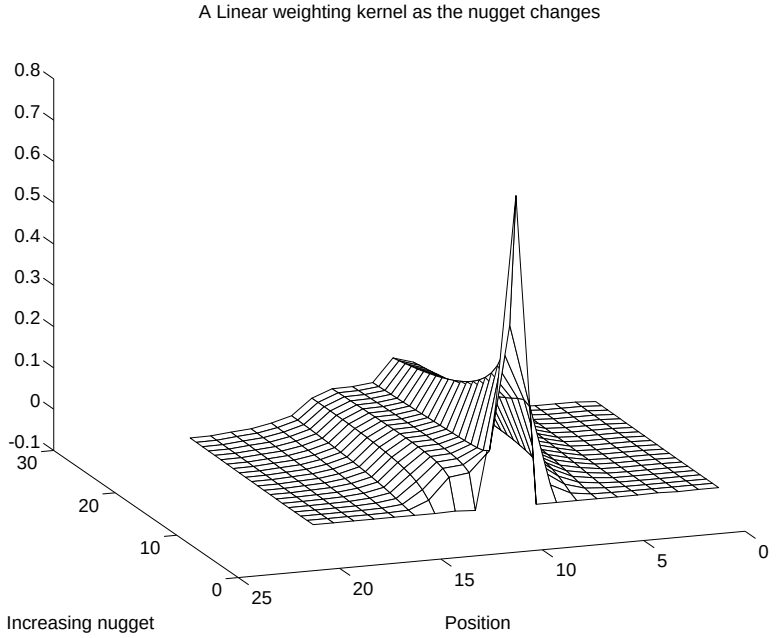


FIGURE 5.14. Kriging weights for the linear model go quickly from shadow effect to smooth kernel as the nugget is increased. The one-dimensional weight distributions are stacked by increasing nugget.

in the weights, perhaps because of the "artificial" range condition (the spherical is only once differentiable at its range).

5.3 Two-Dimensional Case

The two-dimensional case was examined, in order to examine the question of whether the two-d case is qualitatively different from the one-d case. The forms of the apparent kernel functions argue that there seems to be no qualitative difference. Again, a variety of sampling schemes for the "data" were used. The results were not remarkably different for the two cases, however, so the plots shown correspond to the gridded data (which make for nicer plots).

The spherical model showed the shadow effect when used without a nugget (Figure (5.15)). These figures were generated for gridded data in the plane With the nugget, the weights take the form of the so-called "witch's hat", as one can see, of the form

$$\kappa(x) = \exp(-|\mathrm{x} - \mathrm{x}_0|)\cos(|\mathrm{x} - \mathrm{x}_0|).$$

Even in two dimensions, the spherical without a nugget gives large weights only to sites very near the estimation site. The same is true for the linear and exponential models without nuggets.

Figure (5.16) shows the bizarre behavior of the weights when used in spherical extrapolation. This is typical of all models when used for extrapolation, both concave up and down. That's an incredible pattern, which only the convolution of a smooth function with the inverse of a smooth matrix could concoct: it seems likely that the kernel function will be difficult to describe in this case!

The gaussian model at zero nugget again shows no shadow effect; and again it resembles a sinc function, whose width is a function of both the size of the nugget and the range of the gaussian. Just as in the one-dimensional case, the two-dimensional case demonstrates the wavy behavior of the non-nugget model, and demonstrates as well the attenuation of the oscillations once a nugget term is added.

5.4 Discussion

We speculate that the shadow effect may have loomed large in some peoples' minds because it does occur in some models, particularly the linear model. It quickly dissipates, though, as soon as even the smallest amount of nugget is added. This is important, because many models described in the literature include some nugget term (because variograms of real data seldom seem to approach the origin). The nugget plays the crucial role, announcing to the scheme that the variogram model suggests unreliability of neighbors, thus pushing all other weights up in the vicinity while reducing the nearest.

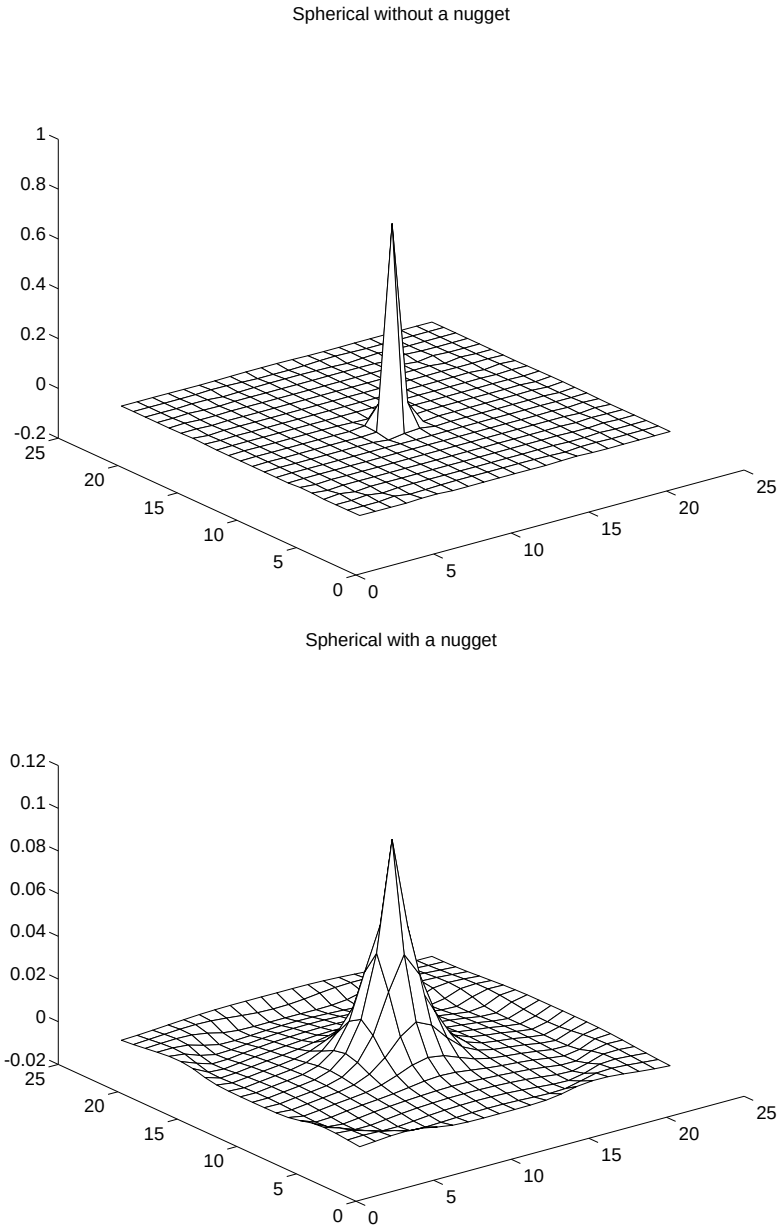


FIGURE 5.15. Kriging weights suggest a "witch's hat" kernel function in the case of a spherical model, with (below) and without (above) a nugget. The shadow effect seems to appear in the weight pattern above, without nugget.

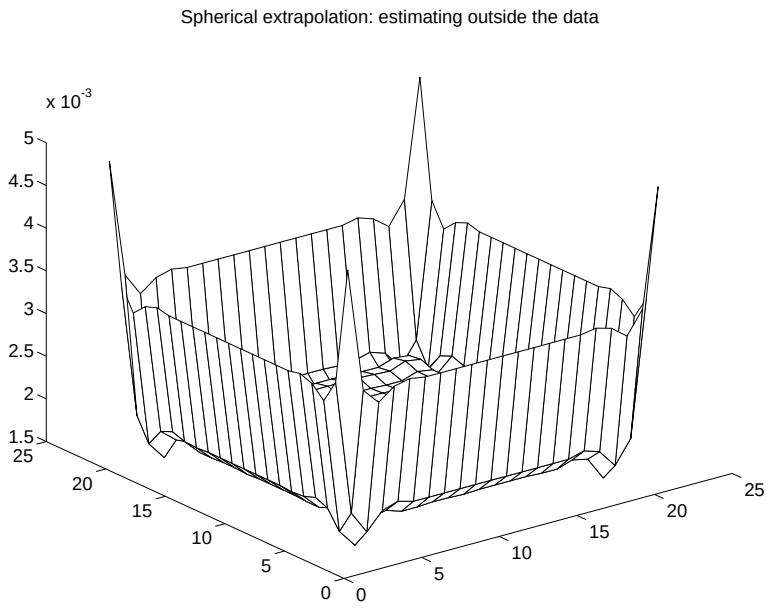


FIGURE 5.16. This sort of extrapolatory behavior is typical for all models.

This effect is as pronounced in the two-dimensional case as it is in the one-dimensional case (Figure (5.9)), thus belying the belief that there is some qualitative difference between the one-dimensional and two-dimensional cases. It is not obvious, however, how those kriging systems corresponding to models which show a two-dimensional shadow effect (e.g. the spherical, linear, and exponential) determine the number of neighbors and which of the neighbors which will receive large weights (the one-d case is much simpler, as there are definitively two nearest neighbors, which divide all the weight (Figure (5.12))).

Although not much attention has been paid to the range in these figures, the effect is as expected: the increasing range tends to spread out the apparent kernel functions (provided that no shadow effect is displayed: otherwise, as in the case of the linear, there is no change).

One surprising effect of increasing the nugget/sill ratio is the width augmentation, which is similar to the effect achieved by increasing the range. This may be due to renormalization, which keeps the sum of the weights at 1.

In sum: the shadow effect is simply another way of saying that the "width" of the apparent kernel function determined by the variogram model is very small; the only standard models which showed it are concave down, and then lose the shadowing property as soon as a nugget is added.

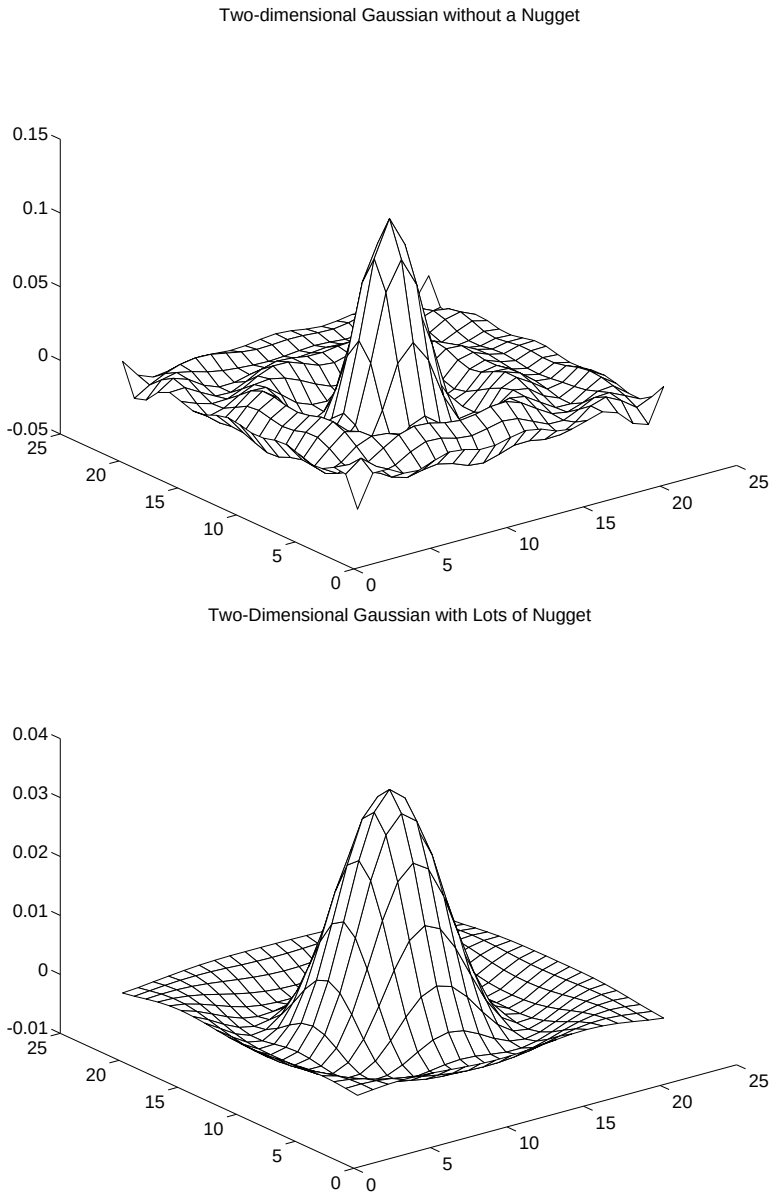


FIGURE 5.17. Kriging weights for a gaussian model, both without (top) and with (bottom) a nugget.

5.5 Analytical and Experimental Results

The continuous version of the ordinary kriging equations can be written as a system of integrals of the form

$$\int_{-1}^{1} V(x-y)w(y)d\mu(y) + \phi(x_0) = v(x-x_0), \tag{5.5.6}$$

or

$$\int_{-1}^{1} V(x-y)w(y)d\mu(y) = v(x-x_0) - \phi(x_0),$$

where the point measure $d\mu(y)$ ranges over the support S, assumed to lie in the interval $[-1,1]$, with the side condition that

$$\int_{-1}^{1} w(y)d\mu(y) = 1. \tag{5.5.7}$$

Suppose that the point at which the estimate is desired, x_0, is not in the support (if it were, then the estimate would be the value $z(x_0)$: alternatively, $w(y)$ would be the delta function $\delta(y-x_0)$).

The LHS of the matrix system (5.5.6) is a discrete sampling of the continuous symmetric Hilbert-Schmidt kernel [84] of the same form (shown in Figure (5.18)). It is best represented in a distorted frame, with actual distances between data points preserved in inter-column spacing (Figure (2.2)). As the number of data points increases, the matrix tends to look more like the continuous version V. As Preisendorfer discusses in [76], this may imply that the singular vectors of V as a matrix will approach the singular vectors of the continuous H-S kernel. Such an observation gave rise to the study of empirical orthogonal functions, which play an important role in meteorology and other disciplines.

Notice that there are now two kernels in the discussion: it is essential to distinguish between them. There is first of all the apparent kernel function associated with a variogram model, which will serve as an approximating function to the weight (solution) vector w of the kriging system; and there is secondly the H-S kernel represented (discretely) by the matrix $V(x-y)$.

The goal is as follows: to obtain the approximating weight kernel by solving the integral equation featuring the H-S kernel V, only using continuous data; that is, "invert the continuous matrix", and use the solutions as a kernel. The well-defined function obtained by the continuous inversion is then compared to the weights found in finite kriging systems.

In the example to follow, the expansion of the inverse is given in an infinite basis of eigenfunctions, but the function which corresponds to that expansion has not been isolated. Ultimately, that is the goal: to get a nice form for the the "pseudo-solution", the apparent kernel function, which we can use to get a good approximation to the

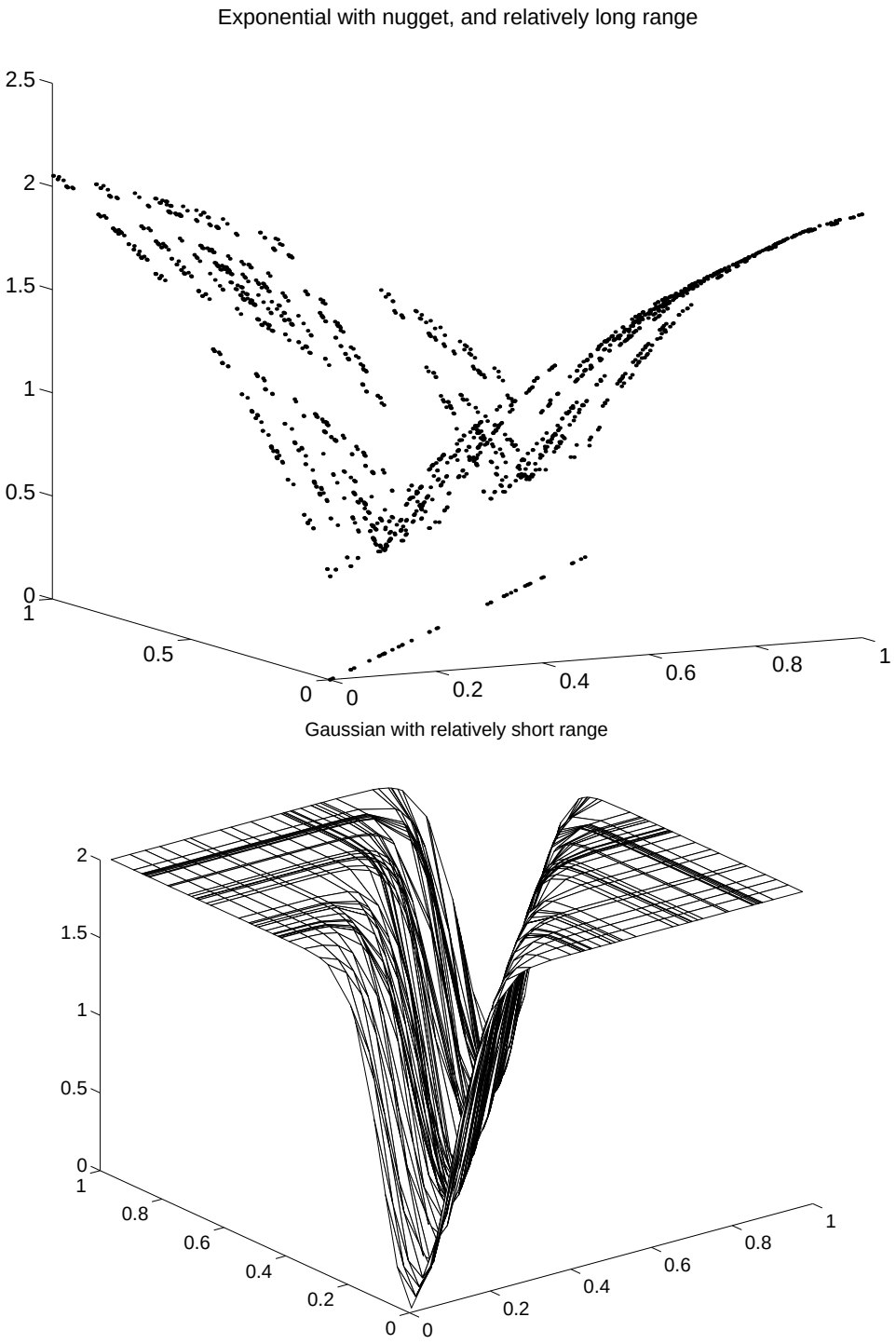


FIGURE 5.18. The ordinary kriging matrices (minus the row and column of ones) for two standard models for one-dimensional scattered data sets. Top, exponential with nugget; bottom, gaussian, without nugget.

kriging weights, as Silverman did. In the meantime, the solution obtained works fairly well, where only a finite part of the infinite Fourier series obtained from the operator is used.

The details are carried out with the exponential model (with nugget), given by

$$v(x-y) = s\left(1-\exp(-|\mathrm{x}-\mathrm{y}|)\right) + n\left(1-\delta(x-y)\right).$$

where s denotes the sill and n the value of the nugget. Some insight into this model is provided by Cressie, in [16], where he shows that the kernel in the case of extrapolation on gridded data is given by complete shadowing, using only the nearest neighbor. The range has been left out for simplicity in the following discussion, although it is easily reintroduced at the end. The integral equation (5.5.6) with this model is

$$\int_{-1}^{1} \left[s\left(1-\exp(-|\mathrm{x}-\mathrm{y}|)\right) + n(1-\delta(x-y))\right] w(y)dy = v(x-x_0) - \phi(x_0),$$

which, taking advantage of the constraint (5.5.7), yields

$$\int_{-1}^{1} \exp(-|\mathrm{x}-\mathrm{y}|)\mathrm{w}(\mathrm{y})\mathrm{dy} + \frac{\mathrm{n}}{\mathrm{s}}\mathrm{w}(\mathrm{x}) = \exp(-|\mathrm{x}-\mathrm{x}_0|) + \frac{\phi(\mathrm{x}_0)}{\mathrm{s}}. \tag{5.5.8}$$

Considering the continuous case, with arbitrary RHS, this is a Fredholm integral equation of the second or first kind (depending on whether there is, or is not, a nugget). Notice that it is the presense of the nugget term which distinguishes the two cases.

Stakgold treats part of this example in [84] (pp. 365-366): he shows that the exponential kernel is a Green's function, so that the eigenvalues and eigenfunctions of the integral operator with kernel $\exp(-|\mathrm{x}-\mathrm{y}|)$ on this interval are related to those of the differential equation

$$u'' = \theta u,$$

with boundary conditions

$$u'(1) + u(1) = u'(-1) - u(-1) = 0,$$

where $\theta \equiv 1 - \frac{2}{\lambda}$, and λ is an eigenvalue of the integral equation. The general solutions are

$$A\exp(\sqrt{\theta}) + \mathrm{B}\exp(-\sqrt{\theta}),$$

but, as one can verify, there are no positive eigenvalues that satisfy the boundary conditions (i.e. there are no real exponential solutions). There is, however, a countably infinite set of negative eigenvalues and corresponding eigenfunctions. Setting $\rho = \sqrt{-\theta}$, the orthonormal eigenfunctions are

$$c_i(x) = \frac{\cos(\rho_{1\mathrm{i}}\mathrm{x})}{\sqrt{1+\sin^2(\rho_{1\mathrm{i}})}}$$

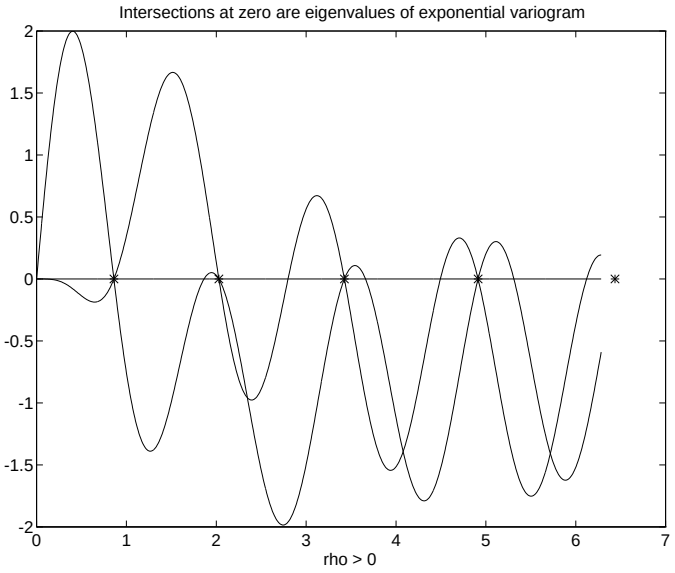


FIGURE 5.19. Joint zeros (origin excepted) of these functions give the eigenvalues for the differential equation coinciding with the case of the exponential variogram. Values are converging on integral multiples of $\frac{\pi}{2}$.

$$\text{and} \tag{5.5.9}$$

$$s_i(x) = \frac{\sin(\rho_{2\mathrm{i}}\mathrm{x})}{\sqrt{1+\cos^2(\rho_{2\mathrm{i}})}},$$

where $\rho_{1i} = \cot(\rho_{1i})$, and $\rho_{2i} = -\tan(\rho_{2i})$, with $\rho_{2i} \neq 0$. Then the eigenvalues of the differential equation are $\theta = -\rho^2$, whose values can only be approximated numerically (as roots of the transcendental equations for ρ given above). Figure (5.19) shows the locations of all (positive) ρ of the differential equation simultaneously, as roots of a single function

$$r(\rho) \equiv \tan(4\rho) - \frac{4\rho(\rho^2-1)}{1-6\rho^2+\rho^4}.$$

The first eigenvalue belongs with the cosine, with alternation afterwards.

The integral equation has corresponding simple eigenvalues

$$\lambda_{\cdot i} = \frac{2}{1+\rho_{\cdot i}^2} = \begin{cases} 2\sin^2(\rho_{1\mathrm{i}}) \\ 2\cos^2(\rho_{2\mathrm{i}}) \end{cases},$$

which are all positive and have zero as a limit point (but zero is not an eigenvalue). As one may simply check, the eigenvalues of the differential equation are tending to integral multiples of $\frac{\pi}{2}$.

The set of eigenfunctions (5.5.9) forms a basis for $L_2(-1,1)$ with which one can solve the inhomogeneous Fredholm equation (5.5.8), as the kernel is a symmetric

compact operator [84]. Expanding w and the exponential in the right-hand side in the basis of the eigenfunctions of the kernel (re-numbering, and calling them q_i),

$$w(x) = \sum_{i=1}^{\infty} w_i q_i(x),$$

and

$$\exp(-|\mathrm{x} - \mathrm{x}_0|) = \sum_{\mathrm{i}=1}^{\infty} \lambda_\mathrm{i} \mathrm{q_i}(\mathrm{x}_0)\mathrm{q_i}(\mathrm{x}).$$

Notice that the eigenfunctions have been used to express $\exp(-|\mathrm{x} - \mathrm{x}_0|)$, a "column element" of the kernel, as an outer-product of eigenfunctions.

Let

$$1_i \equiv \int_{-1}^{1} q_i(y)dy = \begin{cases} \sin(\rho_{1\mathrm{i}})\tan(\rho_{1\mathrm{i}}) \\ 0, \ \rho_{2i}. \end{cases}$$

This notation indicates that these are the coordinates of the expansion of 1, the "canonical constant", in the basis; equivalently, 1_i is the inner product of 1 and eigenfunction q_i.

We now seek conditions on the coefficients of w such that equality is maintained in equation (5.5.8). Rewriting the inhomogeneous equation and solving term by term gives

$$\lambda_i w_i + \frac{n}{s} w_i = \lambda_i q_i(x_0) + \frac{\phi(x_0)}{s} 1_i.$$

This implies that

$$w_i = \frac{\lambda_i q_i(x_0) + \frac{\phi(x_0)}{s} 1_i}{\lambda_i + \mu}, \tag{5.5.10}$$

with $\mu = \frac{n}{s}$, and so

$$w(x; x_0) = \sum_{i=1}^{\infty} \frac{\lambda_i q_i(x_0) + \frac{\phi(x_0)}{s} 1_i}{\lambda_i + \mu} q_i(x).$$

There is a condition on the value of $\phi(x_0)$: it was chosen to ensure the constraint (5.5.7), which means that if w is a solution of the integral equation satisfying the constraint, then

$$\frac{\phi(x_0)}{s} = \left(\sum_{1}^{\infty} \frac{1_i^2}{\lambda_i + \mu} \right)^{-1} \left(1 - \sum_{1}^{\infty} \frac{q_i(x_0)\lambda_i 1_i}{\lambda_i + \mu} \right). \tag{5.5.11}$$

This expression is obtained by summing the weights obtained in (5.5.10) to 1, and solving for $\phi(x_0)$.

Approximating the infinite sum (5.5.11) with a finite sum yields an approximation to the desired kernel. However, doing so introduces a null-space upon eliminating all components of frequency higher than a certain n. Therefore the solution should reflect this, by adding in a term f_0 such that

$$f_0(x) \equiv \exp(-|\mathrm{x} - \mathrm{x}_0|) - \sum_{\mathrm{i}=1}^{\mathrm{n}} \lambda_\mathrm{i} \mathrm{q}_\mathrm{i}(\mathrm{x}_0)\mathrm{q}_\mathrm{i}(\mathrm{x}).$$

This also has the effect of altering the form of ϕ: that given above (5.5.11) was for the infinite series. w and phi are now given by

$$w(x; x_0) = \frac{1}{\mu}\left[\exp(-|\mathrm{x} - \mathrm{x}_0|) - \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{\lambda_\mathrm{i}^2 \mathrm{q}_\mathrm{i}(\mathrm{x}_0)}{\lambda_\mathrm{i} + \mu}\mathrm{q}_\mathrm{i}(\mathrm{x}) + \frac{\phi(\mathrm{x}_0)}{\mathrm{s}}\left(1 - \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{\lambda_\mathrm{i} 1_\mathrm{i}}{\lambda_\mathrm{i} + \mu}\mathrm{q}_\mathrm{i}(\mathrm{x})\right)\right]$$

and (5.5.12)

$$\frac{\phi(x_0)}{s} = \left(n - \sum_{1}^{n} \frac{\lambda_i 1_i^2}{\lambda_i + \mu}\right)^{-1}\left(\mu - \langle \exp(-|\mathrm{x} - \mathrm{x}_0|), 1\rangle + \sum_{1}^{\mathrm{n}} \frac{\mathrm{q}_\mathrm{i}(\mathrm{x}_0)\lambda_\mathrm{i}^2 1_\mathrm{i}}{\lambda_\mathrm{i} + \mu}\right),$$

where the expression of w has been rewritten to get faster convergence ([84]). However, application of the kernel showed that adding the null-space term proved risky, in the sense of leading to poor approximation of the weights: adding nothing led to a good approximation to the rank-n pseudo-solution anyway (that is, the solution obtained from the matrix equations using only the first n singular vectors), and it may be a better idea to go that route in the general case (with non-uniform design points), for safety's sake: it corresponds to a "least-squares" solution, with the right-hand side replaced by its projection onto the span of the eigenfunctions.

Following are the results of applying this method to the case of the exponential model (Figures (5.20) and (5.22)). The procedure was as follows:

- Eigenvalues corresponding to the first 120 eigenfunctions were found (numerically), using the symbolic manipulator Maple. No more eigenfunctions were used than the rank of the matrix system, however (i.e. a maximum of N for an $N \times N$ system).

- Design points were generated. In one case, 100 points were chosen randomly, according to the uniform distribution, to lie on the interval $[-1, 1]$. Approximated and true kriging weight distributions were produced for 20 additional positions for comparison. In the second case, 500 points were chosen to lie uniformly on the interval $[-1, 1]$.

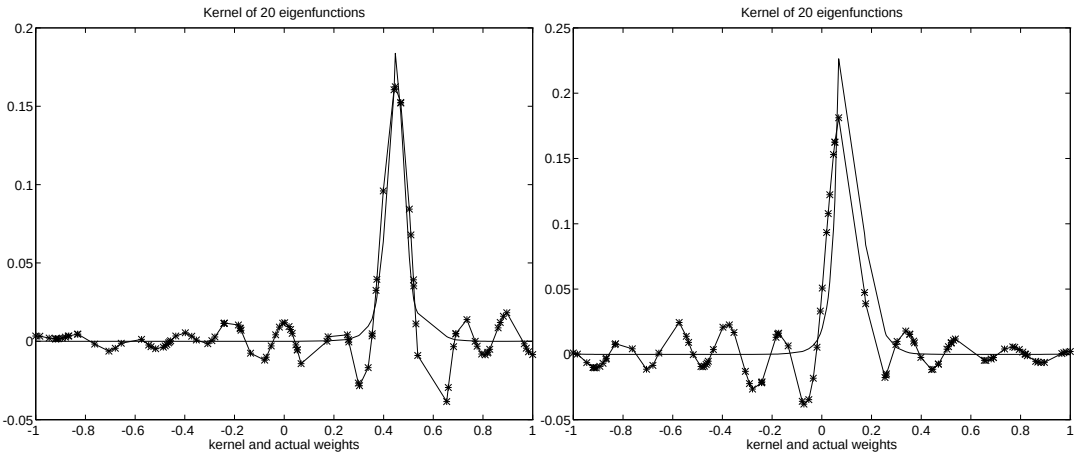


FIGURE 5.20. The best and worst looking weight distributions from a set of 20 random points, on an interval with 100 design points randomly dispersed. Ratio of nugget to sill: .15. The actual weight distributions are smooth and decline monotonically away from the point at which the estimate is desired. (There is not much difference, but the one on the right was considered worst of the twenty.)

- The continuous eigenfunctions were orthogonalized with respect to each other (as vectors) against the random design pattern, i.e.

$$\underline{q}_j = \underline{q}_j - (\sum_{i=1}^{j-1} \underline{q}_j \underline{q}_i)\underline{q}_i,$$

  where the vectors are formed by evaluating the eigenfunctions at the data locations: $q_i = q(x_i)$, and afterwards renormalized. This was done successively, starting with the eigenfunction corresponding to the largest eigenvalue. This seemed logical, and definitely improved performance. Since the eigenfunctions were altered from their original form, an option was given to use a linear spline interpolant to $q_i(x_0)$. The figures were generated without using the spline interpolant, however, although using the spline improved appearance. The eigenfunctions were then re-orthogonalized.

- The re-orthogonalized eigenfunctions were multiplied by the variogram matrix, to give the eigenvalues appropriate for the distribution of points. I.e.,

$$\Lambda = \mathrm{diag}(\mathrm{Q}^{\mathrm{T}}\mathrm{VQ}).$$

  This resulted in slight changes from the analytic eigenvalues, due perhaps to the alteration of the eigenfunctions during re-normalization, and to the erratic scattering of the points. It seemed to have the effect of smoothing out extreme values far from the peak.

- One important note: in treating the matrix problem as an integral equation, or vice versa, there is a scale problem: the eigenfunctions of the integral equation take values which are right around 1, whereas the components of the singular vectors of the matrix problem decrease in value as $\frac{1}{\sqrt{N}}$; simultaneously, the singular values of the matrix increase as N.

- Finally, there was a somewhat arbitrary choice, which dictated how well the weights were reproduced: it was necessary to choose a number of basis functions. One might want to do so on the basis of the spacing of the design points. In particular, one wants to avoid the natural aliasing that will occur if too many are used on too coarse a design pattern (high frequencies will alias to look like lower frequencies). The choices for these examples were made empirically. That would obviously not be possible in real applications, however, as one will not compute the kriging weights first in order to decide whether or not one would rather approximate them! On the other hand, it may be that experimentation with standard models and patterns will lead to empirical rules.

How good an approximation, or estimate, can be obtained with this kernel? In fact, it is only the weights that one can analyze, whereas one is ultimately interested in how much effect the approximation will have on the estimate of the quantity of interest: a strange data distribution can give an arbitrarily large difference, no matter how small an actual difference the approximation makes in the weights. Results indicated that for carefully chosen values of the number of basis functions, the weight distributions looked fairly similar and varied by only a small percentage in the area of most importance (around the peak).

Notice that for the case of the unequally spaced design points (Figure (5.20)), there was not much difference between the best and the worst approximate weight distributions of the twenty samples. (Again, good weight distributions were obtained only by neglecting to add in the null-space term). The theoretical values ($q_i(x_0)$), rather than the splined singular vectors for the right-hand side were used).

In the second example case, of 500 design points the null-space term was used (that is, equations (5.5.12)), and Figure (5.21) shows the result of using 40 eigenfunctions. Included in that figure are the rank-40 pseudo-solution (obtained by using only the first 40 singular values of the inverse), and the kernel estimator neglecting the null-space term. The kernel approximation is much better than the rank-40 matrix solution (which is almost identical to the kernel estimator without the null-space term). Without the addition of the null-space term, an approximation to the weights which compares very favorably with the rank-40 approximation the actual kriging matrix was obtained. The low-rank weights summed almost exactly to 1: thus, normalization problems do not account for the lower peaks at the maximum.

A close-up look at the weights when using 120 pseudo-eigenfunctions (5.22) gives

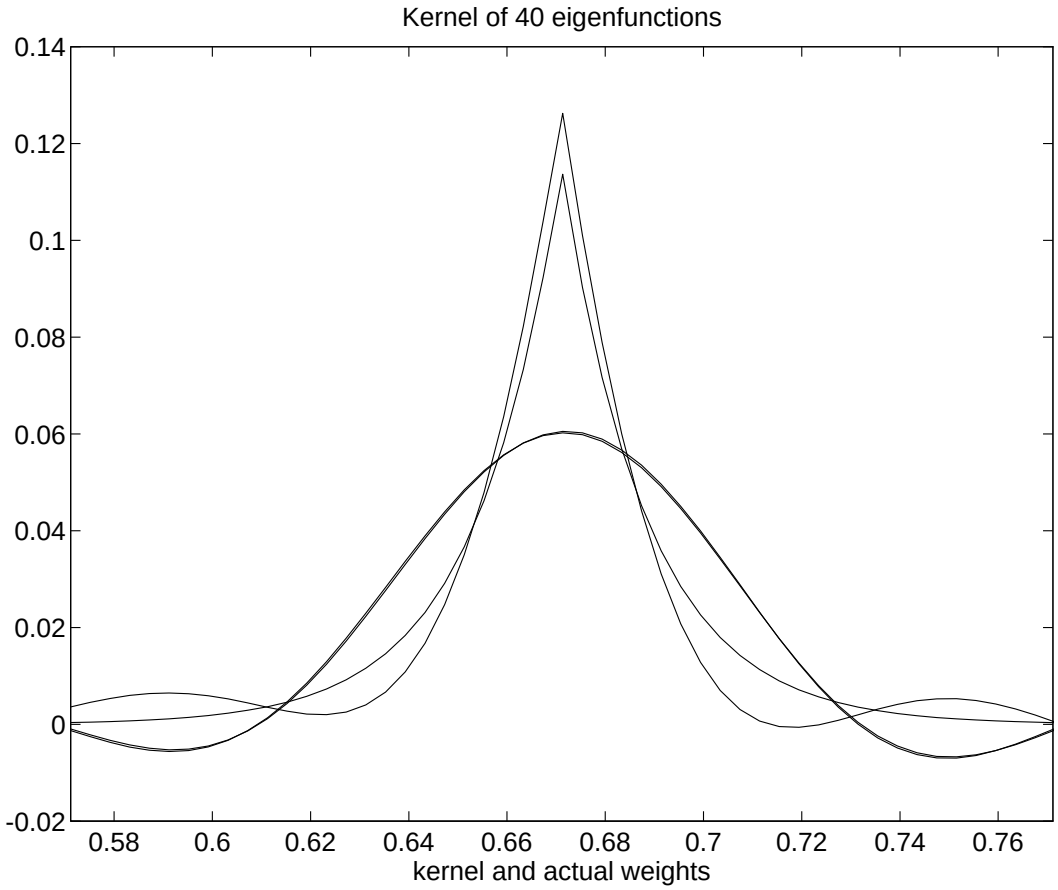


FIGURE 5.21. 40 eigenfunctions, rather than the 500 singular vectors obtained from the kriging system, generate the weights for an approximation to kriging. The kernel is slightly higher at the peak, and oscillates about the true weights away from the peak. Also depicted are the rank-40 pseudo-solution, and the kernel solution without the addition of the null-space term: these last two are essentially identical.

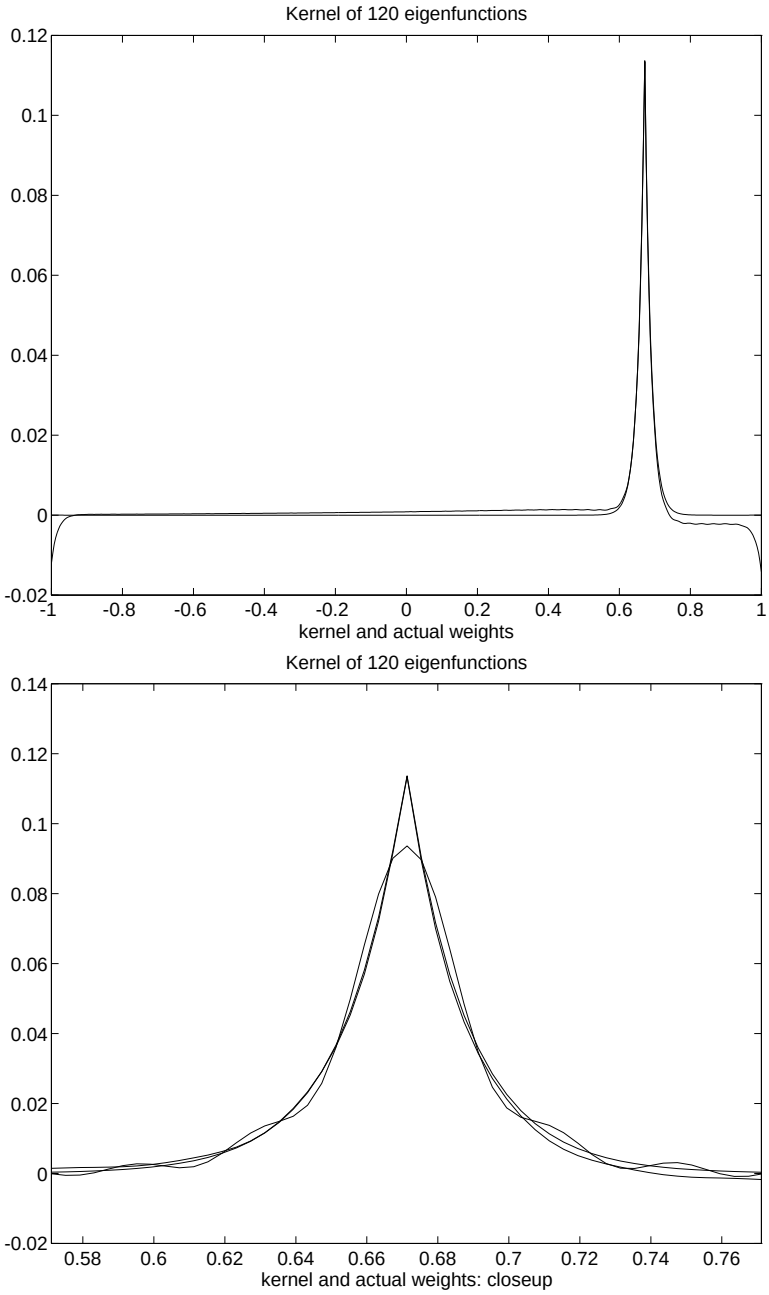


FIGURE 5.22. Above: all weights, in the midst of 500 design points; the kernel dips below the actual weights at right, and is above (for awhile) at left, before dropping below. Below: a close-up view of that weight distributions in a neighborhood of the point at which the estimate is computed. Also included are the rank-120 approximation using the 120-pseudo-inverse, which oscillates, and falls short of the peak. The kernel solution is nearly indistinguishable from that of the true distribution, slightly above at left and below at right. Obviously, a pretty good fit!

a feeling for how well the kernel can perform. The "rank-120 pseudo-inverse" approximation to the weights has been included as well: it has the lowest hump at its maximum, and oscillates.

In the 500 point case, Matlab required 47.3 cpu-seconds on a Sun 4 to compute and orthogonalize 120 "pseudo eigenfunctions", and the pseudo-singular values. The matrix inversion of a 500 element matrix required 77.8 cpu-seconds, while inversion via the SVD required 490.6 cpu-seconds.

There is still much work to do before a kernel formulation substitutes adequately for kriging. The next step will be to determine the function which corresponds to the series of orthogonal functions of the exponential: a kernel which involves such a large number of terms is perhaps too computationally demanding to be very useful.

Silverman discusses the features a kernel estimator should possess, in particular the dependence of the kernel on the distribution of the design points (i.e., the bandwidth), and the amount of smoothing (in the kriging case, how much nugget) that one is doing with it. In future work, these factors should be taken into account as well.

5.6 From Here to Infinity

In this section, the development proceeds in the opposite direction: starting with the kriging equations, we link back up to the infinite-dimensional case. This also provides another opportunity to show the use of the kernel idea in another concrete example.

The ordinary kriging equations can be written as

$$\begin{bmatrix} V & \underline{1} \\ \underline{1}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{w} \\ \mu \end{bmatrix} = \begin{bmatrix} Q\Lambda Q^T & \underline{1} \\ \underline{1}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{w} \\ \mu \end{bmatrix} = \begin{bmatrix} \underline{v}(x_0) \\ 1 \end{bmatrix}$$

Recall how the "fast interpolation" process of the SVD may be used to express the variogram RHS $\underline{v}(x - x_0)$ in terms of the basis of the SVD of V:

$$\underline{v}(x_0) = Q\Lambda\Lambda^{-1}Q^T\underline{v}(x_0) = Q\Lambda\underline{q}(x_0),$$

where $v(x_0)$ is now expressed in terms of the basis of singular vectors of V. $\underline{q}(x_0)$ could be considered as in "interpolated section" of the matrix Q (equation (2.2.7)). In fact, letting x_0 range over all permissible values would reconstitute the kriging method's interpolating function for the singular vectors (recall that any interpolator of the singular vectors reproduces an interpolator for the matrix). This seems really backwards: using the kriging system to interpolate, or induce an interpolation, of the singular vectors in the kriging system; but it is really just equivalent to saying that one can treat x_0 as a variable and solve for any value it takes.

One could think of the fast interpolation process as converting the matrix problem into an integral equation problem: a compact H-S kernel results from using the interpolated outer-products of the SVD of the matrix, i.e.

$$k(x,y) = \sum_{i=1}^{N} \lambda_i q_i(x) q_i(y).$$

As Stakgold notes in [84], this is a standard way for creating example operators with desirable properties.

One now solves the system using the Singular Value Decomposition:

$$\begin{bmatrix} I & Q\Lambda^{-1}Q^T\underline{1} \\ \underline{1}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{w} \\ \mu \end{bmatrix} = \begin{bmatrix} Q\underline{q}(x_0) \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} I & \underline{\alpha} \\ \underline{0}^T & -\underline{1}^T\underline{\alpha} \end{bmatrix} \begin{bmatrix} \underline{w} \\ \mu \end{bmatrix} = \begin{bmatrix} \underline{\beta} \\ 1 - \underline{1}^T\underline{\beta} \end{bmatrix},$$

where $\alpha \equiv Q\Lambda^{-1}Q^T\underline{1}$ and $\beta \equiv Q\underline{q}(x_0)$. One can solve directly, to yield

$$\mu = \frac{1 - \underline{1}^T\underline{\beta}}{-\underline{1}^T\underline{\alpha}} = \frac{1 - \underline{1}^T Q\underline{q}(x_0)}{-\underline{1}^T Q\Lambda^{-1}Q^T\underline{1}}$$

and

$$\underline{w} = \underline{\beta} - \mu\underline{\alpha} = Q\left[\underline{q}(x_0) - \mu\Lambda^{-1}Q^T\underline{1}\right].$$

Proceeding to the infinite dimensional case, one replaces the matrix inner-products by integrals, and obtains the kernel function corresponding to a given variogram model:

$$\mu = \frac{1 - \sum_0^\infty \int q_i(x)q_i(x_0)dx}{-\sum_0^\infty \int\int q_i(x)\lambda_i^{-1}q_i(y)dxdy} = \frac{1 - \sum_0^\infty q_i(x_0)\int q_i(x)dx}{-\sum_0^\infty \lambda_i^{-1}\left(\int q_i(x)dx\right)^2}$$

$$= \frac{1 - \sum_0^\infty q_i(x_0)1_i}{-\sum_0^\infty \lambda_i^{-1}1_i^2} \tag{5.6.13}$$

and

$$w(y) = \sum_0^\infty q_i(y)q_i(x_0) - \mu\sum_0^\infty q_i(y)\lambda_i^{-1}\int q_i(x)dx$$

$$= \sum_0^\infty \left[q_i(x_0) - \mu\lambda_i^{-1}1_i\right] q_i(y) \tag{5.6.14}$$

Note that this is exactly the form obtained from the integral equation (5.5.10), when the nugget is zero. This form is more general, as it applies to all models equally, but requires calculating the singular vectors of the variogram including nugget, rather than allowing its separation via the Fredholm form, as above.

■ Example: The Cosine Model

This isotropic variogram model has the form

$$\gamma(h) = 1 - \cos(\mathrm{h}),$$

as seen previously, so it can be expressed as an outer-product of functions like so:

$$\gamma(x-y) = 1 - \cos(\mathrm{x})\cos(\mathrm{y}) - \sin(\mathrm{x})\sin(\mathrm{y}). \tag{5.6.15}$$

Consider a domain defined as an integral multiple (L, say) of the interval $[-\pi, \pi]$, and normalize the eigenfunctions:

$$\gamma(x-y) = 2\pi L \frac{1}{\sqrt{2\pi L}}\frac{1}{\sqrt{2\pi L}} - \pi L \frac{\cos(\mathrm{x})}{\sqrt{\pi L}}\frac{\cos(\mathrm{y})}{\sqrt{\pi L}} - \pi L \frac{\sin(\mathrm{x})}{\sqrt{\pi L}}\frac{\sin(\mathrm{y})}{\sqrt{\pi L}}$$

$$\begin{cases} q_0(x) = \frac{1}{\sqrt{2\pi L}}; \\ q_1(x) = \frac{cos(x)}{\sqrt{\pi L}}; \\ q_2(x) = \frac{sin(x)}{\sqrt{\pi L}}; \end{cases}$$

$$\begin{cases} \lambda_0 = 2\pi L; \\ \lambda_1 = -\pi L; \\ \lambda_2 = -\pi L. \end{cases}$$

Expanding the weight kernel in this basis, leads (from (5.6.13) and (5.6.14)) to

$$\mu = \frac{1 - \sum_0^2 q_i(x_0) \int q_i(x)dx}{-\sum_0^2 \lambda_i^{-1} \left(\int q_i(x)dx\right)^2}, \quad \text{and}$$

$$w(y) = \sum_0^2 \left[q_i(x_0) - \mu\lambda_i^{-1} \int q_i(x)dx \right] q_i(y).$$

As the interval of interest is an integral number of periods, this is easily solved to give the weight kernel:

$$\mu = 0,$$

and

$$w(y) = \sum_0^2 q_i(x_0)q_i(y) = \frac{1}{2\pi L} + \cos(\mathrm{y} - \mathrm{x}_0)$$

This is valid when data values are at all points on the interval $[-L\pi, L\pi]$, in which case one notices that the sum (or rather the integral) of the weights will be unity. Now we compare the kriging weights with the kernel weights for a set of discrete data locations, either gridded or scattered.

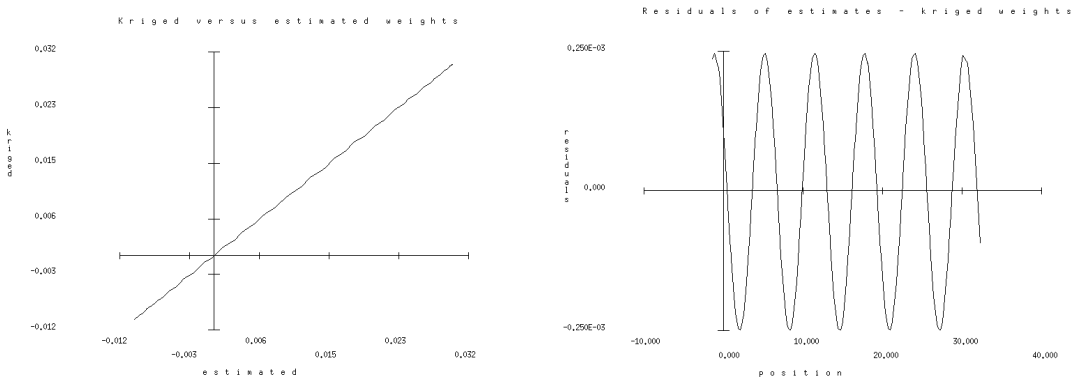


FIGURE 5.23. Kriging weights versus the cosine kernel weights. Variation is systematic, but small, when considering scattered rather than gridded locations.

The approximated kernel function gave excellent results, in the sense that the weights were essentially identical (Figure (5.23)) for gridded data (100 points) on a fine mesh (fine compared with the period of the cosine model), and also with scattered locations generated according to a uniform distribution.

In the latter case the same experiment was tried with a data set composed of 200 points, on the interval $[-2, 33]$. Figure (5.23) shows that there is more variation in the weights, and, furthermore, that the variation is systematic. There seems to have been a phase shift which was unaccounted for. Even so, the variation is only a few percent of the actual weights, which means that, in general, there should be little change in the estimates.

In the derivation of the kernel function, it was assumed that the interval of interest was an integral multiple of periods. If this assumption had not been made, one would not have had orthogonality between the functions of (5.6.15), which made the analysis simpler. Even so, in the event that the extent of the data is large compared to the period, the functions of (5.6.15) will still be good approximations to the singular vectors, μ will tend to zero as $\frac{1}{L}$, and hence the weights will approach the same kernel function as $\frac{1}{L^2}$. The far greater leap was in using the weights obtained for continuous data on a set of discrete data locations.

5.7 Discussion

When should one consider using the kernels discussed in this chapter in real interpolation and estimation problems? The answer is probably that, at this point, one should not be using them at all. There are many issues still unaddressed, including rigorous demonstrations that passage to the limits in cases such as (5.6.13) is permissible, bandwidth corrections, border effects, etc. However, some preliminary

information is available, based on this study: rigorous demonstration or no, good approximations to the kriging weights using kernels have been obtained in several cases, which indicates that kernels may one day have a broader and better motivated place in geostatistical analysis (especially where the data set is large and well-distributed in space).

One issue is the degree to which the set of eigenfunctions, orthogonal for a continuous set of data, is still orthogonal (or "essentially so") and even linearly independent for a discrete set of data points. The intuitive notion is that, if the data locations are essentially randomly dispersed over the line, with a fairly uniform distribution, then kernel replacement will probably be okay; if, on the other hand, data is clumped and poorly distributed, then kriging should be considered. Kriging has the ability to decluster data, averaging clusters, which kernels clearly do not have (although the design point adjustments should address that issue to some extent).

In the derivation above, the orthogonality of the matrix Q implied that $Q^TQ = I$. If the eigenfunctions of the kernel associated with the particular variogram are used, instead of the eigenvectors of the SVD, then one won't generally have that identity, and will be forced to renormalize the eigenfunctions, maintaining the total proportion of each eigenfunction by adjusting the eigenvalue appropriately. Thus, another heuristic argument is that the kernel should be appropriate when $Q^TQ \approx I$ for the adjusted eigenfunctions (i.e. if the eigenvectors so obtained are still essentially orthogonal to each other).

It may also be that the matrix Λ will not be invertible: this is the case for the cosine model, for instance. In such cases, one may use the pseudo-inverse. The consequences of using the pseudo-inverse may be similar to those found in the exponential model, however: that, when the matrix should be full rank but may be so poorly conditioned that the pseudo-inverse is a necessity, then the lower-rank approximation to the weight distribution (e.g. figure (5.22)) will give a much smoother estimating kernel.

Chapter 6

Case Study: Nitrate Pollution in the Phoenix Area

6.1 Introduction

We present herein some results from a recent study [52] of nitrate pollution in an area around Phoenix, Arizona. The study was funded by the Arizona Department of Environmental Quality and the United States Geological Survey.

The database consisted of a large set of well water sample analyses (approximately 700) for a period extending over 15 years in time, and a pair of land-use maps created at two times in that period. The 700 plus well samples do not represent 700 wells, since some wells were sampled more than once, at different times. 34 wells were sampled in three separate time periods, which were given roughly as 1975-1977, 1980-1985, and 1986-1990. The goals of the geostatistical portion of the study included

- arriving at the best method for unbiased estimation of nitrate concentration, and
- mapping nitrates, and changes in nitrates, over the time span of the data.

The data came from many sources [1]; most of the data were hand-entered, from hand-written paper forms, and many human errors (both on the original forms and in the entry process) were discovered by inspecting the data visually and statistically. While much effort went into checking and repairing the data, it is not certain that all such errors were eliminated. Furthermore, in most cases no information about the reliability of the lab work performed was available. The quality of well data is essentially unknown, but quite possibly poor.

Nitrate was the variable of principal interest, although many additional variables had been sampled at the sites. These include specific conductivity, ph, water temperature, bicarbonate, carbonate, hardness, calcium, magnesium, sodium, potassium, sulfate, chloride, fluoride, silica, dissolved solids, phosphorus, aluminum, arsenic, barium, boron, cadmium, chromium, copper, iron, lead, lithium, manganese, nickel, silver, selenium, strontium, and zinc. Many of these variables were actually analyzed at only a few sites; some variables were simply missing from an analysis. Because so many agencies provided data for the study, the data collected were not uniform at

[1] The Salt River Project; the Cities of Phoenix, Glendale, and Sun City; the Roosevelt Irrigation District, Buckeye Irrigation Company, Metropolitan Water District, Sunnyboy Water company, and other smaller companies.

sites: some agencies tested for many more variables than other agencies, which led to an uneven distribution of numbers of sites for a given variable.

Some variables were reported as "non-detects" in samples, which means that the concentration was not detectable by the instruments used to measure them. These were replaced by the stated non-detect level: i.e., if the form stated "$N < .06$", then $N = .06$ was used. Thus, while very small anyway, these values were exaggerated. One of the consequences of looking at data collected over a long time span is that the detection levels change as equipment improves; this led to the unfortunate consequence that a non-detect was replaced by one value in one data set and by another value in another data set.

Other options for dealing with non-detects include replacing them by some fraction of the detection level in existence at the time (or lab), or attempting to bootstrap a distribution for the non-detects [37, 80].

There were neither the resources nor the authority to collect new data. Sampling groundwater at a site without an existing well is an expensive and lengthy procedure. Some well owners were not receptive to having their water tested, which made random sampling impossible. Additional distortion of the true picture of nitrate concentrations in the area was also introduced by the practice of shutting off drinking water wells which exceed the allowable limits of nitrate concentration (removing them from future sampling).

Some variables were reported in various forms: for example nitrates were reported as N, as NO_3, and as $NO_2 + NO_3$ dissolved. Conversion factors were used, increasing the potential for errors.

Analyses can be checked for "ionic balance": if the proportion of anions to cations was deemed to be in error, then the results of the analysis were discarded. Many analyses were thrown out on this basis, suggesting a high degree of error in the analyses in general.

The quality of the land-use maps was also questionable: the earlier map was obtained by actual ground survey, and is probably fairly accurate; the more recent map was digitized, using the first map and areal photos as a guide. Some glaring errors were detected in the second map, based on known features of the landscape. While these errors were detected and corrected, the extent of undetected error in the maps is still essentially unknown.

Several procedures were used for choosing a set of variables to use for multivariate analysis and interpolation. The most obvious choice used those variables which were sampled frequently across all times. That eliminated the heavy metals particularly, and some other variables. Another example is depth-to-perforation: it was eliminated because there was no assurance that the wells were perforated in only a single layer. The condition of the wells was, for the most part, unknown, and the water may be from multiple layers.

Thus, there were a number of interesting features of the study:

- the temporal aspect of the data set, which permitted a study of how the area changed on a relatively long time scale, but also made for problems such as changes in lab work over time;
- the data quality issues, because the data came from many sources, with varying degrees of consistency and reporting standards;
- the use of varying detection limits over time, as methods of detection improved; and
- hidden sources of data set distortion, such as the fact that wells which exceed pollution limits were shut off, and so eliminated from further study, etc.

In the end, all those "interesting features" impeded the most important part of the study, as it was impossible to characterize the uncertainty in the data, thus precluding successful characterization of the uncertainty in the maps. The comparison between interpolation methods was unaffected by this uncertainty, as the comparison was carried out without the imposition of assumptions on the data quality (other than the assumption implied by using an exact interpolator).

Multiple interpolation methods were compared, including kernel estimators, such as inverse square weighting; radial basis functions (a technique essentially like kriging [70], but without pretense of statistical assumptions); kriging and cokriging; and a variety of black-box routines, from the public domain package GRASS (Geographical Resources Analysis Support System), from ARC/INFO, S-Plus, and other commercial software packages. With the exception of the inverse square methods (used in some commercial software) comparative cross-validation results for the black-box routines are not included.

The multiquadric radial basis function was used, a popular choice in the radial basis function literature. This function is chosen for use in the dual formulation (4.2.7), but usually without reference to the variogram. The multiquadric is given by

$$g(|\underline{h}|) = \sqrt{|\underline{h}|^2 + \mu^2},$$

where μ is either guessed or chosen by optimizing cross-validation statistics. Note that as $|\underline{h}|$ gets large, this approximates a linear variogram model.

6.2 Cross-Validation Results

In order to compare interpolation methods, a set of criteria is needed. These were based on a "leave-one-out" procedure called **cross-validation**: that is, a datum $z(x_i)$ is removed from the data set, an estimate $z^*(x_i)$ is then made at x_i, and $z(x_i)$ and $z^*(x_i)$ are compared. This is done for all i, to give measures of "goodness-of-fit". Thus cross-validation addresses the question "How well is the particular method estimating

the actual data values from the remainder of the data?" This provides some idea of how much uncertainty there will be away from the data locations.

The following cross-validation statistics were used (see [69]):

- $\overline{z^* - z} \equiv \frac{1}{N}\sum_{i=1}^{N}(z^*(x_i) - z(x_i))$ should be close to zero;
- $\overline{(z^* - z)^2}$ should be small;
- $\rho(z^*, z)$, the sample correlation coefficient, should be close to 1;
- when comparing kriging and cokriging, the normalized estimation error, given by

$$u(x) = \frac{z^*(x) - z(x)}{\sigma(x)} \tag{6.2.1}$$

  at each site x, where $\sigma(x)$ is the (co)kriging standard deviation, should be have a mean-square close to 1, i.e. ideally

$$\frac{1}{N}\sum_{i=1}^{N}\frac{[z^*(x_i) - z(x_i)]^2}{\sigma^2(x_i)} = 1;$$

- the estimates $z^*(x_i)$ should have statistics similar to (but not necessarily the same as) those of the true values, $z(x_i)$: that is, the means, standard deviations, extremes, skewness, etc. correspond "to a reasonable degree".

Tables (6.2), (6.2), and (6.2) contain the results of the comparison of several techniques, and a variety of cokrigings. Because of the conditioning and matrix modelling problems, only two-variable cokriging was considered. Each additional variable useful for nitrates was modelled and cokriged with nitrate, and cross-validation statistics were compared. (Models for the variograms and cross-variograms of the variables used in cokriging are found in the appendix.) The cokriging results were obtained with a fortran program based on the kriging program of the public domain package Geo-EAS. Because estimation variances are not available for techniques other than kriging and cokriging, the statistics relating to the normalized estimation variable described in (6.2.1) have not been included.

Although tests showed that universal cokriging gave better results than ordinary cokriging, ordinary cokriging cross-validation statistics are included for that variable giving the best universal cokriging results. Ordinary cokriging was done twice for that variable, for 10 and 20 neighboring sites (limits suggested by McCarn and Carr [64]).

One interesting experimental test involved cokriging nitrates with a second variable generated as a linear combination of some of the other variables. There is one

obvious way of doing this: use linear regression to form the best surrogate for nitrate from the other variables, by solving the least-squares regression problem

$$X\underline{b} = \underline{y},$$

where $\underline{y}$ was the vector of nitrate concentrations, and X was the matrix of other variables used.

The second way in which to do this was by using principal component analysis as a regression technique. PCA was carried out on the matrix $[X|\underline{y}]$, with nitrates ($\underline{y}$) in the last (p^{th}) column. Let

$$S_{N\times p} = (I - \frac{1}{N}\underline{1}\underline{1}^T)[X|\underline{y}]\mathrm{diag}(\Sigma)^{-\frac{1}{2}}$$

be the shifted and scaled data. Then $S = U\Lambda Q^T \equiv AQ^T$ by the SVD, and the best rank-one approximation to the data is

$$S\underline{q}_1 = \underline{a}_1. \tag{6.2.2}$$

Dropping the subscript on the principal singular vectors, the PCA approximation to (scaled) nitrates is

$$s_{ip} = \frac{1}{q_p}(a_i - \sum_{j=1}^{p-1} s_{ij}q_j),\ q_p \neq 0.$$

This was tested because of its even-handed treatment of the variables: that is, PCA gives just one single best linear regression equation (6.2.2) for the p variables, whereas linear regression gives p different regression equations. For example, one may not merely invert the linear regression equation for y on x, $y = ax+b$, to get the regression equation for x on y [56, 97]: it will not necessarily be $x = (y-b)/a$. Linear regression does not give the best-fit line to the point cloud (x, y) in terms of orthogonal projection from the points to the line, but rather the best-fit line to either $(x, ax+b)$ or $(ay+b, y)$. PCA, using the SVD, provides the optimal line in the former sense: PCA gives the best fitting line to the (centered) data cloud, which can then be inverted for any case.

Surrogate variables, one obtained by linear regression and the other from PCA, were modelled and cokriged against nitrate as well. The results were especially good using the linear regression variable: it had the best cross-validation statistics for the 1977 data set. However, as this was considered an experimental procedure, the linear regression variable was not used to generate the final map: magnesium, giving the next-best results, was used instead. Poor cross-validation statistics were obtained using the linear regression variable for the 1988 data. The PCA variable gave reasonable results for all three data sets, but never the best; on the other hand, it never gave bad cross-validation statistics.

The "modern methods" did fairly well in general. Cokriging gave smaller values for the kriging variance than did kriging, and tended to do quite a bit better in terms

method	$\overline{z^* - z}$	$\overline{(z^* - z)^2}$	$\rho(z^*, z)$	$\sigma(z^*)$	$\overline{z^*}$	z^*_{min}	z^*_{max}
true nitrate	0	0	1	0.9817	10.03	7.248	11.83
n/lr	0.0272	0.3887	0.7970	0.9789	10.01	7.341	12.05
n/mg	0.0091	0.4318	0.7413	0.7157	10.02	7.891	11.92
(n/mg,o,10)	0.0158	0.6069	0.6083	0.6532	10.02	8.491	11.38
(n/mg,o,20)	-0.0018	0.6210	0.5932	0.5705	10.03	8.738	11.30
n/cl	0.0097	0.4411	0.7348	0.7050	10.02	8.182	11.72
n/pca	0.0067	0.5470	0.6560	0.6091	10.03	8.839	11.69
krige	0.0110	0.6248	0.5900	0.5918	10.02	8.705	11.76
(krige,o,10)	0.0182	0.6096	0.6059	0.6494	10.01	8.485	11.37
(krige,o,20)	-0.0026	0.6238	0.5908	0.5676	10.04	8.711	11.32
$(\frac{1}{d^2}, 20)$	0.0411	0.6540	0.5827	0.7124	9.992	8.072	11.39
rbf	0.0343	0.7702	0.5403	0.8325	9.998	7.695	11.97
$(\frac{1}{d^2}, 4)$	0.0756	0.8232	0.5318	0.8841	9.957	7.446	11.64
n/ca	-0.2228	24.75	0.1618	5.048	10.26	-0.0208	39.78

TABLE 6.1. Cross-Validation results for each method, circa 1977. Row notation: n/lr means nitrate universally cokriged with the linear regression variable, in a global neighborhood; n/mg, with magnesium; n/cl, with chloride; n/pca, with the principal component variable; (n/mg,o,10) means ordinary cokriging with 10 neighbors; n/ca, with calcium; "krige" means universally kriged nitrate in a global neighborhood; (krige,o,10) means ordinary kriging, with 10 neighbors; for the inverse square distance cases, the second number in the pair again represents the number of neighbors used; rbf means radial basis function, which was a multiquadric. Order is roughly best-to-worst, with the ordinary cokriging results following those for the best universal variable.

of the $\overline{(z^* - z)^2}$ statistic. It sometimes happened, however, that several methods had good statistics, each best on one. For example, in the 1988 table (6.2) pca regression had the smallest mean error, while linear regression the lowest mean-square error. This raises the question: "When the two variables give very similar cross-validation statistics, but different maps, which map should one choose?"

One possibility in the case of a tie would be to examine the "cross-variance" maps for the cokriging cases: that is, plots of the cokriging variances of the two variables. Although no definitive rule about them exists, there seems to be some correlation between those pairs of variables which do well in cokriging cross-validation and the appearance of these plots. For example, one reason that the magnesium variable was chosen over the linear regression variable in the 1977 data set was that its cross-variance map did not have the extreme values that the linear regression map did.

method	$\overline{z^* - z}$	$\overline{(z^* - z)^2}$	$\rho(z^*, z)$	$\sigma(z^*)$	$\overline{z^*}$	z^*_{min}	z^*_{max}
true nitrate	0	0	1	1.000	10.000	5.396	11.61
n/ca	0.0031	0.3273	0.8196	0.8340	9.997	6.183	11.34
(n/ca,o,10)	-0.0319	0.5644	0.6592	0.6615	10.03	7.666	11.21
(n/ca,o,20)	-0.0296	0.5414	0.6782	0.6288	10.03	7.791	11.07
n/cl	0.0061	0.3751	0.7904	0.8247	9.994	6.042	11.29
n/pca	0.0024	0.4175	0.7653	0.6969	9.998	5.837	11.15
krige	0.0041	0.5191	0.6921	0.6868	9.996	5.662	10.99
(krige,o,10)	-0.0290	0.5580	0.6640	0.6747	10.03	7.262	11.22
(krige,o,20)	-0.0265	0.5363	0.6811	0.6426	10.03	7.268	11.08
rbf	0.0057	0.5861	0.6613	0.8212	9.994	6.643	11.54
$(\frac{1}{d^2}, 10)$	-0.0236	0.5769	0.6521	0.7133	10.02	7.431	11.27
$(\frac{1}{d^2}, 4)$	-0.0142	0.6406	0.6250	0.8079	10.01	7.281	11.44
$(\frac{1}{d^2}, all)$	-0.0574	0.6525	0.6059	0.4690	10.06	8.759	11.01
n/lr	0.0062	1.015	0.5026	1.023	9.994	5.259	12.53
n/mg	0.4244	14.54	0.4206	4.108	9.576	-26.27	28.78

TABLE 6.2. Cross-Validation results for each method, circa 1985

method	$\overline{z^* - z}$	$\overline{(z^* - z)^2}$	$\rho(z^*, z)$	$\sigma(z^*)$	$\overline{z^*}$	z^*_{min}	z^*_{max}
true	0	0	1	1.003	9.993	7.341	11.84
n/mg	0.0026	0.4316	0.7568	0.8182	9.990	7.051	11.27
(n/mg,o,10)	-0.0160	0.5248	0.6925	0.7490	10.01	8.025	11.22
(n/mg,o,20)	-0.0178	0.5174	0.6960	0.7157	10.01	7.990	11.23
n/lr	-0.0036	0.4263	0.7584	0.7836	9.996	7.932	11.34
n/ca	-0.0032	0.4620	0.7363	0.7945	9.996	8.059	11.25
n/pca	-0.0004	0.4712	0.7279	0.7240	9.993	8.103	11.12
krige	-0.0015	0.5196	0.6944	0.7207	9.994	7.979	11.09
(krige,o,10)	-0.0182	0.5200	0.6957	0.7476	10.01	8.067	11.26
(krige,o,20)	-0.0213	0.5156	0.6972	0.7119	10.01	8.038	11.20
rbf	0.0038	0.5645	0.6794	0.8397	9.989	7.555	11.52
$(\frac{1}{d^2}, 10)$	0.0026	0.5550	0.6736	0.7627	9.990	8.082	11.28
$(\frac{1}{d^2}, 4)$	-0.0061	0.6009	0.6595	0.8480	9.999	7.587	11.59
n/cl	-0.7513	110.2	0.0173	1.690	10.01	1.956	19.48

TABLE 6.3. Cross-Validation results for each method, circa 1988

Certainly, if there is a negative kriging variance, then this will show up instantly, indicating a serious problem (probably a modelling problem, giving rise to an ill-conditioned or invalid cokriging matrix); and if some kriging variances are inordinately different or large, this too should be obvious. This is simply a qualitative assessment, but it seems not to have been described in the literature. The cross-variance maps are included in the appendix (Figures (**??**), (**??**), and (**??**)). At any rate, visual inspection of the variance maps is simple, and may yield some insight into the results.

The cross-validation results, in particular the mean-square error, indicate that kriging generally does about as well as any of the paired cokriging results, and it did well consistently. This is perhaps the most important conclusion of the mapping: that kriging, while not optimal in the sense of giving the best cross-validation statistics, is perhaps the cheapest, fastest way to obtain near-optimal results (they may not be the best results obtainable, but they may be good enough).

It is interesting to note that in each period, one pair of cokriged variables gave terrible cross-validation statistics: nitrate-chloride in 1988; nitrate-magnesium in 1985; and nitrate-calcium in 1977. Magnesium was chosen as the best cokriging variable with nitrates (in 1977 and 1988) and calcium in 1985. These are two variables which gave poor results in one other (seemingly very similar) case. The failures were either the result of bad cross-variogram modelling or matrix condition problems, as negative cokriging variances occurred. As Figure (6.1) shows, however, the Cauchy-Schwartz condition was not violated, which appears to implicate the matrix condition or sampling pattern problems. Note that, as mentioned in Chapter Three, the corhograms of the winners are piled up around the origin (magnesium in 1977 and 1988, and calcium in 1985).

One can also speculate about whether this was due to changes in the area chemistry, or whether it was simply a matter of poor modelling. Figure (6.2) shows the comparison of the sample variograms for nitrate and its cross-variograms with the additional variables. Considering the cross-variograms of nitrate and magnesium, for example, it seems that the 1985 case is midway between the 1977 and 1988 cases. The same can be said of calcium. This argues against the belief that the difference is a result of different groundwater chemistry, although it is not conclusive by any means.

In sum: poor results were obtained in one case from variables that gave good results in another case. Thus, if one had blindly assumed that magnesium cokriged best against nitrates, because it had done well in 1977, and had used the nitrate/magnesium map again for the 1985 data set, then one would have been deceived. This demonstrates the importance of cross-validation to the map-selection process.

6.3 Overview of Mapping Results

Four different contour maps, generated with four different methods, are shown in

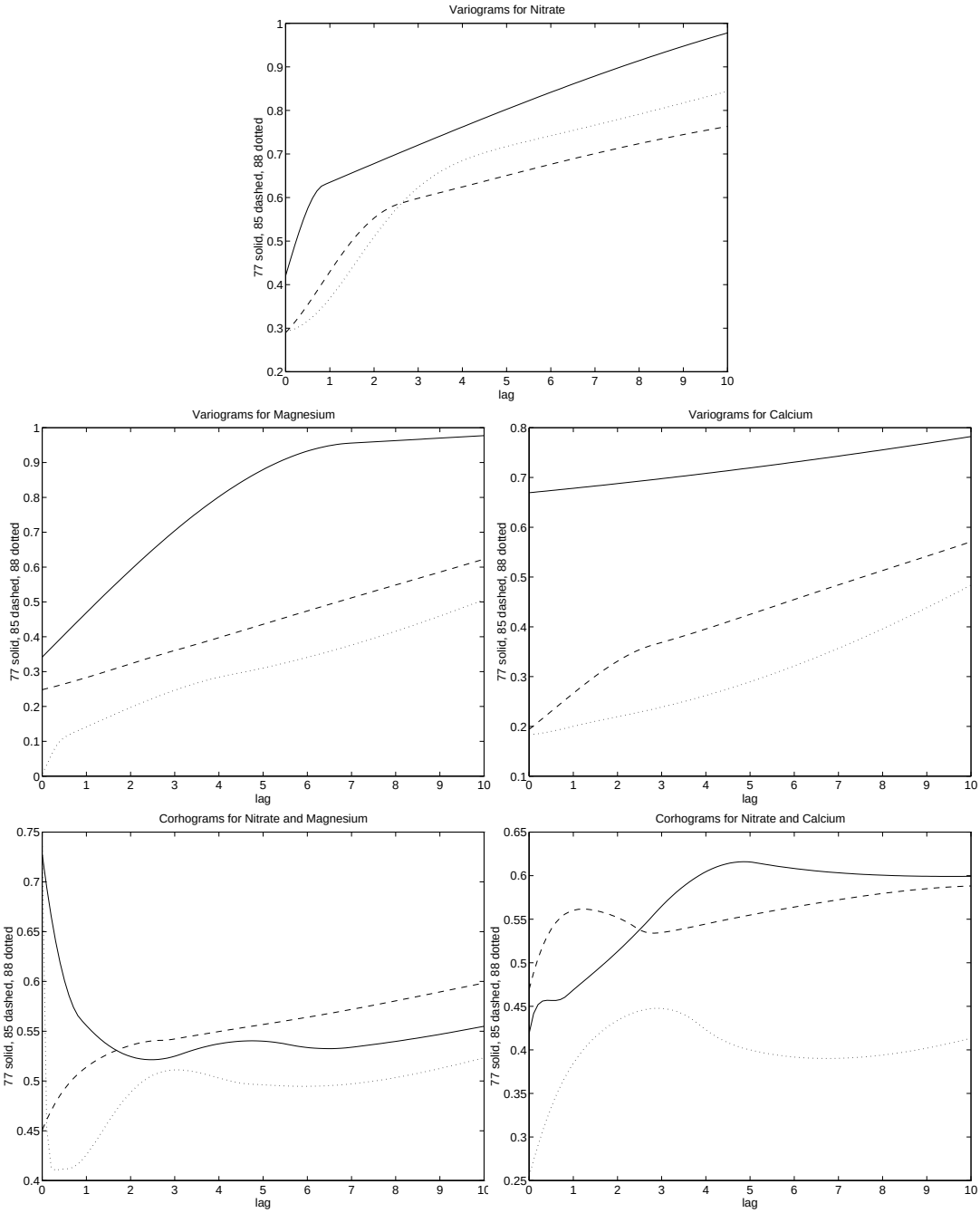


FIGURE 6.1. The three data sets give rise to three sets of Nitrate and Magnesium models, cross-variograms, and corhograms. No corhogram failed ($|\rho(h)| > 1$) for the intervals used in the matrix systems. 1977: solid lines; 1985: dashed lines; 1988: dotted lines.

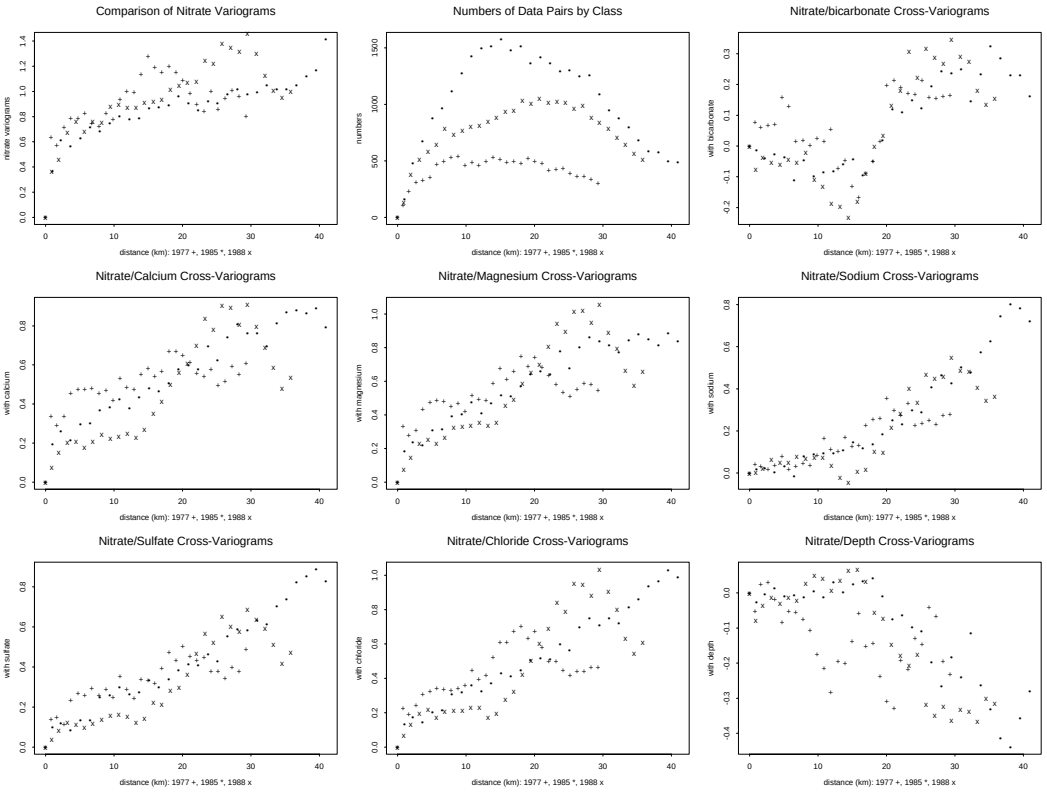


FIGURE 6.2. A comparison of the isotropic sample variograms of nitrate, and cross-variograms of nitrate and other variables of interest for the three data sets. N.B.: the variogram values of zero at zero are not necessarily realistic, but were added to force plots to include the origin, and to indicate the size of the nugget.

Figure (6.3). All were created using the same contour line values, so if all methods gave the same results, all the maps would look exactly the same: there are obvious differences in their appearance. Clearly criteria are needed, such as the cross-validation procedures just described, to help select the best one.

The cross-validation statistics indicated that universal cokriging produced the best maps. We used a second-order polynomial drift, on log-transformed values (an assumption of log-normality of groundwater chemical data is common [2]), and produced maps in the back-transformed nitrate concentrations for three different (somewhat well-characterized) periods. The success of cokriging is in accord with the results reported by others, for example Pan et al. [72].

Our final recommendation concerning mapping techniques was somewhat different, however: the difficulties inherent in cokriging, including the modelling steps, and the sizes and condition numbers of the matrices involved, led us to suggest that kriging is the preferred method in many cases. This is especially true for those unskilled in the art/science of cross-variogram modelling.

This recommendation could change in the future, as improvements in cokriging methodology reduce the size of the coefficient matrices to invert (e.g. the new cokriging algorithm, described herein), and as modelling becomes easier (e.g. linear coregionalization); the linear approximations described in Chapter Four may serve as a starting point for the decision as to which variables to use in the study. At present, however, cokriging is not yet easy enough and stable enough to use unconditionally.

6.4 Diagonalizing the Data

Xie [98] did not actually model the variograms of the diagonal elements he obtained. We do so in this section, using the models in the kriging (rather than cokriging) process: original variable estimates are then reconstituted from the estimates of the transformed variables, using the inverse linear transformation. This is analogous to factorial kriging [79, 28], and, like factorial kriging, is appropriate provided that all variables are of equal importance.

Factorial kriging is a clever scheme, related to the notion of a coregionalization: one exchanges the original variables in a study for linear combinations of variables, which often come from an application of PCA to the data matrix. The new variables (linear combinations of the original variables) so derived are empirically uncorrelated:

$$(I - \frac{1}{N}\underline{1}\underline{1}^T)X\mathrm{diag}(\Sigma^{-\frac{1}{2}}) = \mathrm{U}\Lambda\mathrm{V}^{\mathrm{T}};$$

kriging is carried out on the variables in U, then estimates for X, X^*, are given by

$$X^* = U^*\Lambda V^T\mathrm{diag}(\Sigma^{\frac{1}{2}}) + \underline{1}\overline{\mathrm{x}}^{\mathrm{T}}.$$

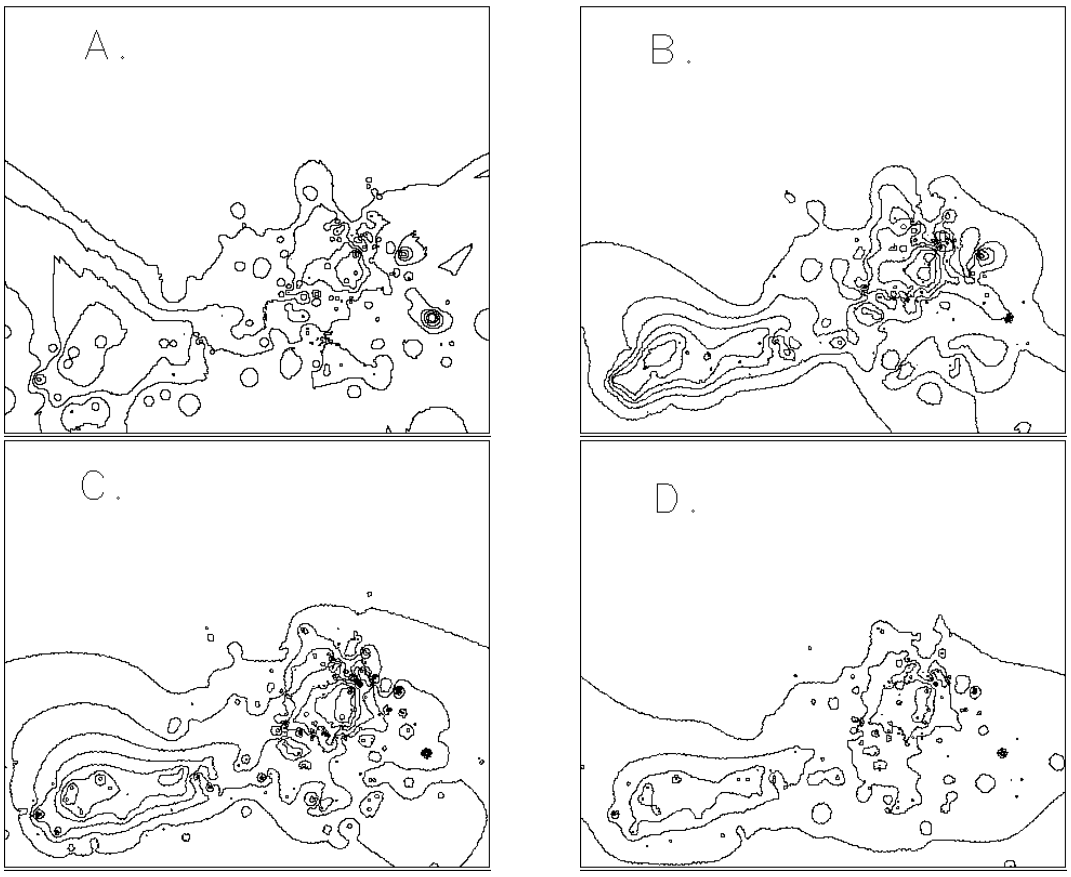


FIGURE 6.3. Four methods, four maps: A. inverse distance squared; B. radial basis function (multiquadric); C. kriging; D. cokriging. There is significant variation in these maps: how is a manager to choose? These maps, of nitrate concentrations, were produced for the same area, for the same period (around 1985). The same contour levels were used in all maps, although they are not marked, as the goal of this figure is to simply point out the obvious differences in the maps.

PCA gives empirically uncorrelated variables, but only at lag zero. (As shown earlier, Xie's method and the TSVD method reduce the correlation over all lags, by minimizing the cross-variograms). The linear combinations are kriged, and the results transformed back to the original variables.

The data Xie used in his dissertation came from the Nitrate study: 171 data locations for the three variables bicarbonate, calcium, and magnesium, for a period around 1977. Sample variograms and cross-variograms for these variables were computed using the automated procedures described elsewhere in this dissertation. Nine variograms were computed for the raw data R (three for the variables themselves, and six for their sums and differences); Myers's method (equation (3.4.14)) was used to obtain the three cross-variograms. We proceeded similarly for the diagonalized data D, which was obtained from the original data by the transformation matrix B of Xie (3.5.18):

$$D = RB.$$

Once the diagonalized data were kriged, the grid was re-transformed, via

$$R^* = D^* B^T.$$

In two of the eighteen cases, variogram models were altered manually, then recomputed using the least-squares method. Inspection showed that the automated technique had failed to model a short-range effect. A guess was made as to the missing model, a portion of the nugget was replaced by the sill of the guess, and the automated technique was used again, to get a good visual fit. The Cauchy-Schwartz condition was satisfied in all pair-wise cases.

Ordinary cokriging results were obtained using a double-precision fortran code tested against published results and routines ([55, 13]). Figures (6.4), (6.5), and (6.6) show the contour maps for kriged, transformed/kriged, and the winning cokriging (if any). In the case of bicarbonate, it appears that the transformed, kriged, and retransformed map is closer to the cokriged map, whereas in the case of calcium the kriged map looks slightly more similar. For both of these variables, the cross-validation statistics of the kriged results were essentially identical (the variances of the diagonalized data could not be retransformed).

Cokriging with all three variables produced negative variances, which meant a serious problem, and cross-validation statistics are invalid (Table (6.4)). Plots showed that the cokriged map had dramatic highs and lows outside of the kernel of the data. This may be an indicator that the size of the matrix $((3*171)\times(3*171))$ was a factor in giving unstable results. On the other hand, it may be that the models involving magnesium led to singular variogram model matrices: recall that a corhogram of 1 implies non-invertibility of the variogram matrix, as

$$V = \begin{bmatrix} \sqrt{\gamma_1(h)} & 0 \\ 0 & \sqrt{\gamma_2(h)} \end{bmatrix} \begin{bmatrix} 1 & \rho(h) \\ \rho(h) & 1 \end{bmatrix} \begin{bmatrix} \sqrt{\gamma_1(h)} & 0 \\ 0 & \sqrt{\gamma_2(h)} \end{bmatrix},$$

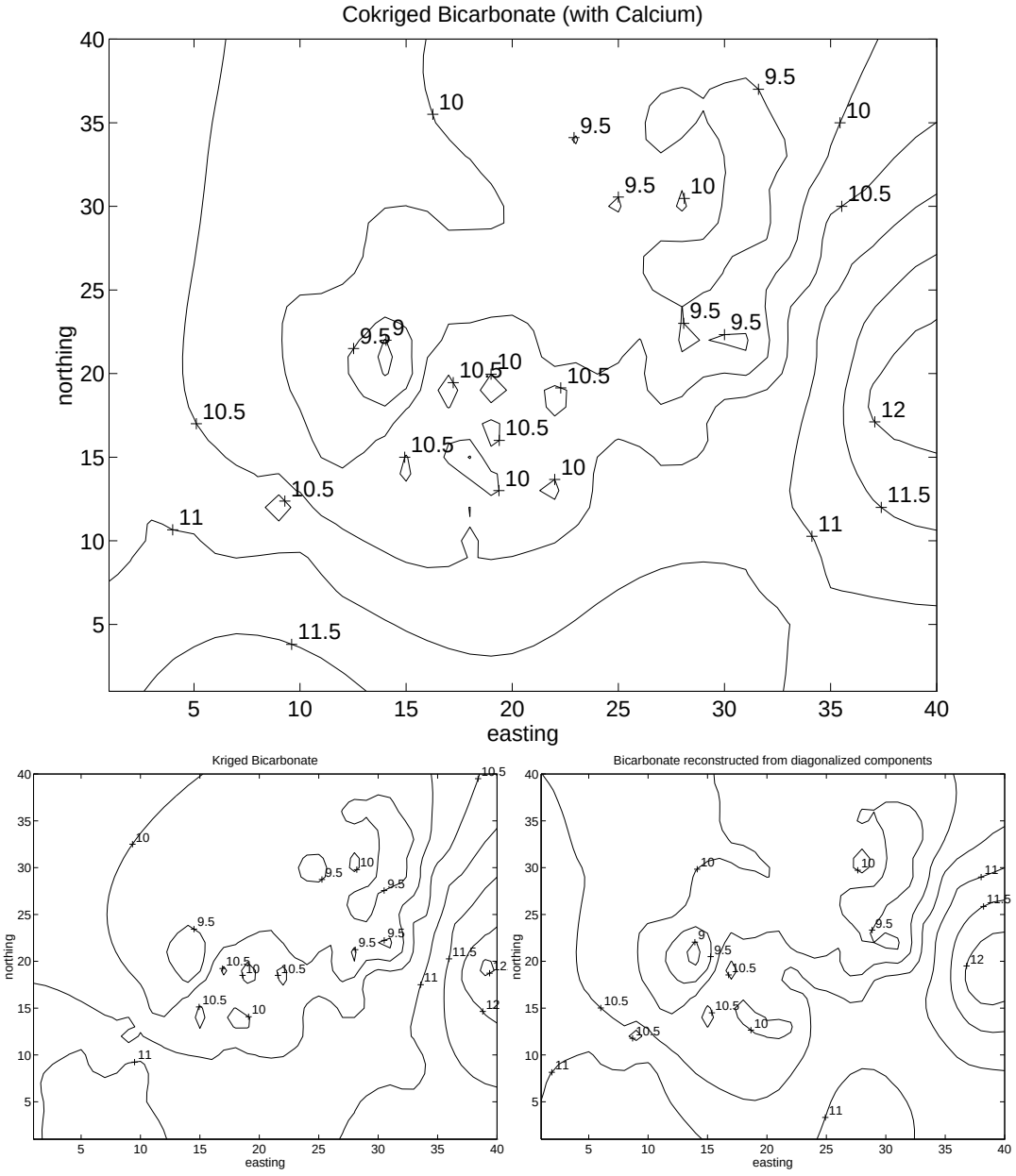


FIGURE 6.4. Bicarbonate contours, for cokriging and kriging of the raw data, and kriging of the transformed data, retransformed to the original. Results were contoured to the same intervals.

method	$\overline{z^* - z}$	$\overline{(z^* - z)^2}$	$\overline{\left(\frac{z^*(h)-z(h)}{\sigma(h)}\right)^2}$	$\rho(z^*, z)$	$\rho(z^*, \frac{z^*(h)-z(h)}{\sigma(h)})$	$\overline{\sigma(h)^2}$
ideal:	0	0	1	1	0	0
kr, bi	0.0113	0.5325	0.9238	0.6760	0.0523	0.5827
kr,trans	0.0087	0.5325	N/A	0.6750	N/A	N/A
co (12)*	0.0058	0.5224	1.0060	0.6824	0.0065	0.5200
co (13)	0.0060	0.5585	1.1004	0.6572	-0.0730	0.5034
co (all)	0.0050	0.7323	1.4915	0.5595	-0.2877	0.4948
kr, ca	0.0104	0.4128	0.7636	0.7603	0.0585	0.5544
kr,trans	0.0076	0.4152	N/A	0.7582	N/A	N/A
co (12)*	0.0033	0.4116	0.8346	0.7599	0.0112	0.4986
co (23)	0.0585	2.0629	5.0059	0.4292	-0.6783	0.1619
co (all)	invalid	invalid	invalid	invalid	invalid	invalid
kr, mg*	0.0018	0.4466	0.7627	0.7411	0.0640	0.6037
kr,trans	0.0080	0.4610	N/A	0.7309	N/A	N/A
co (13)	-0.0031	0.5037	0.9395	0.7009	-0.0527	0.5251
co (23)	-0.0324	0.6248	2.8418	0.7828	-0.5721	0.1528
co (all)	0.0018	0.5229	3.0511	0.7828	-0.4790	0.1510

TABLE 6.4. Cross-Validation statistics for the data Xie used in his dissertation [98], using kriging, cokriging, and all sub-cokrigings. Starred results were judged the best of their group. "kr,trans" results were from kriging the diagonalized (transformed) variables, then linearly retransforming; variance-related cross-validation could not be retransformed so easily.

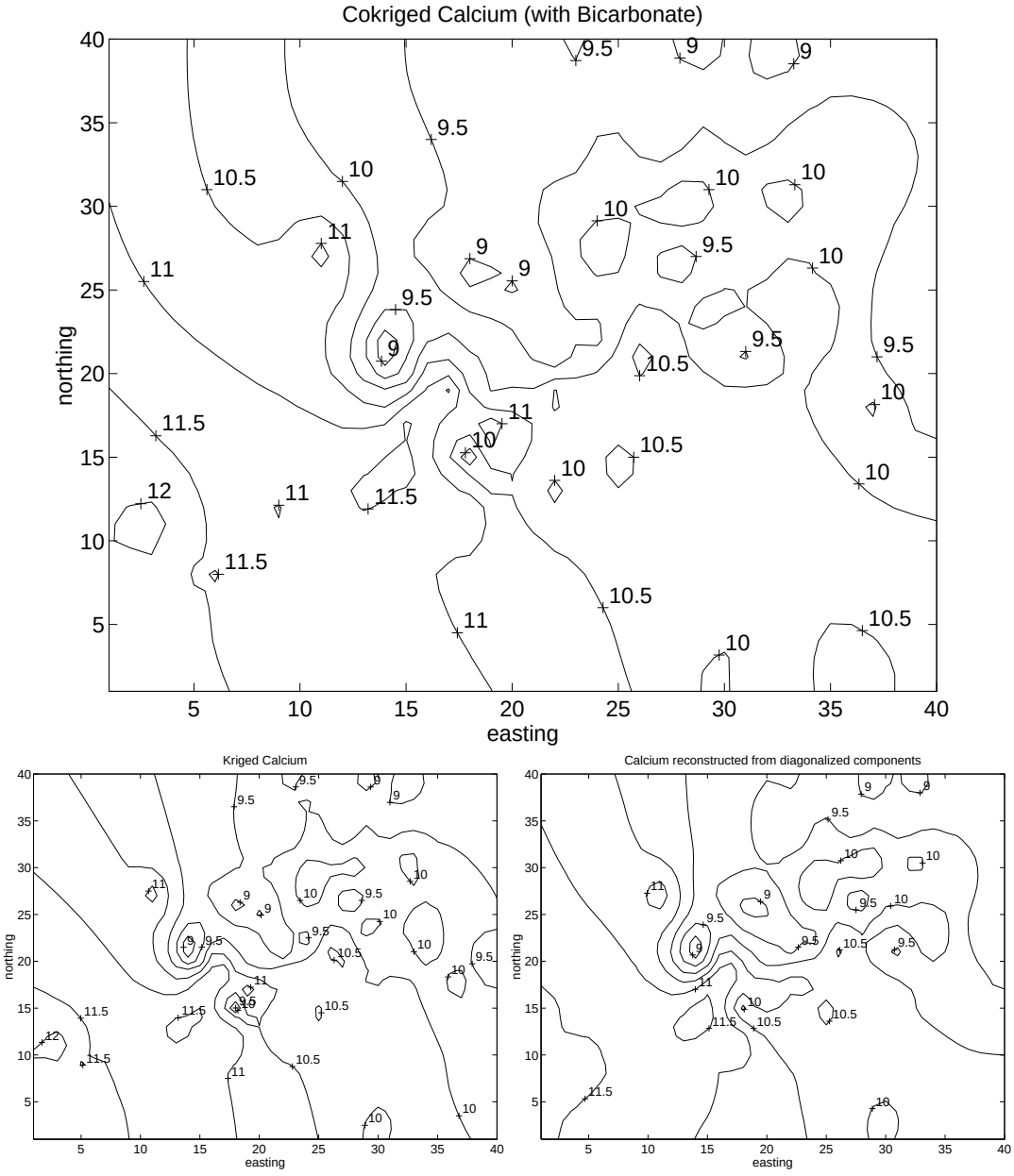


FIGURE 6.5. Calcium contours, for cokriging and kriging of the raw data, and kriging of the transformed data, retransformed to the original. Results were contoured to the same intervals.

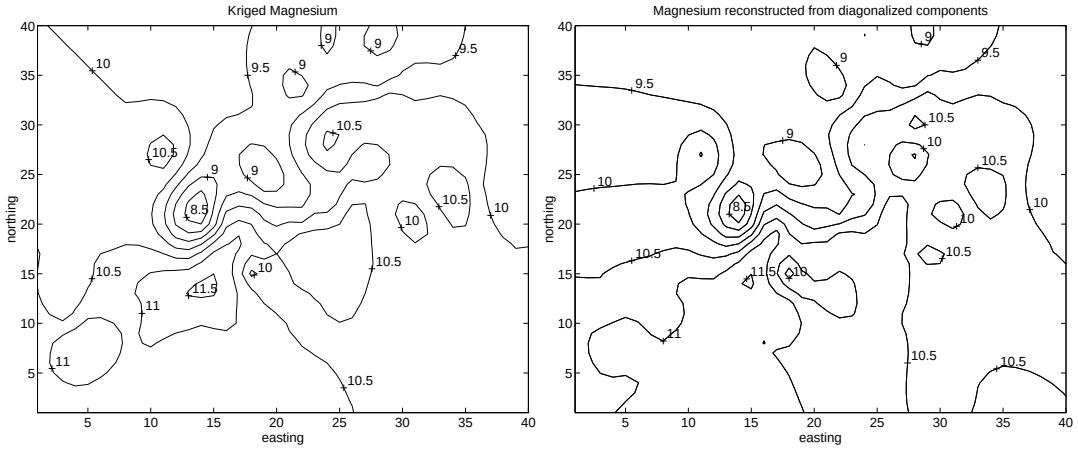


FIGURE 6.6. Magnesium contours, for kriging of the raw data, and kriging of the transformed data, retransformed to the original. Results were contoured to the same intervals. Kriging beat cokriging, and raw kriging did better than did kriging transformed data.

which means that if $\rho(h) = 1$ the matrix V is rank-one. The corhogram of magnesium and calcium was high (≈ 1) at the origin, and, as seen in Table (6.4), the cokriging of those two also gave poor cross-validation statistics.

6.5 Linear Approximation to Cokriging

The results obtained from the linear approximation to cokriging of the diagonalized variables 1 and 2 were disappointing. In this case, the norms for the matrices A_1A_2 and A_2A_1 of (4.2.14) were rather high, 0.3270 and 0.3757; these values may therefore serve as a starting point for the discussion of "how high is too high" to make the linear approximation. The variation from kriging was radically higher for the linear approximation than for cokriging, and, as one can see, the contour map for the approximated variable is a poor approximation to the cokriging map.

The condition numbers of the kriging matrices were 562.3 and 356.6, while the two additional matrices to invert (M_1 and M_2) had condition numbers of 1.5 and 1.6. The condition number of the large Myers system matrix was 565.6, scarcely higher than the kriging matrix of variable 1. The kriging matrices need to be inverted anyway, however, and therefore, in place of inverting the 344×344 matrix of condition number 565.6, it is only necessary to invert two additional matrices of size 172×172, with condition numbers of little more than 1.

6.6 TSVD of Common Sites

As described earlier in the chapter on Variogram Analysis, one can also use the

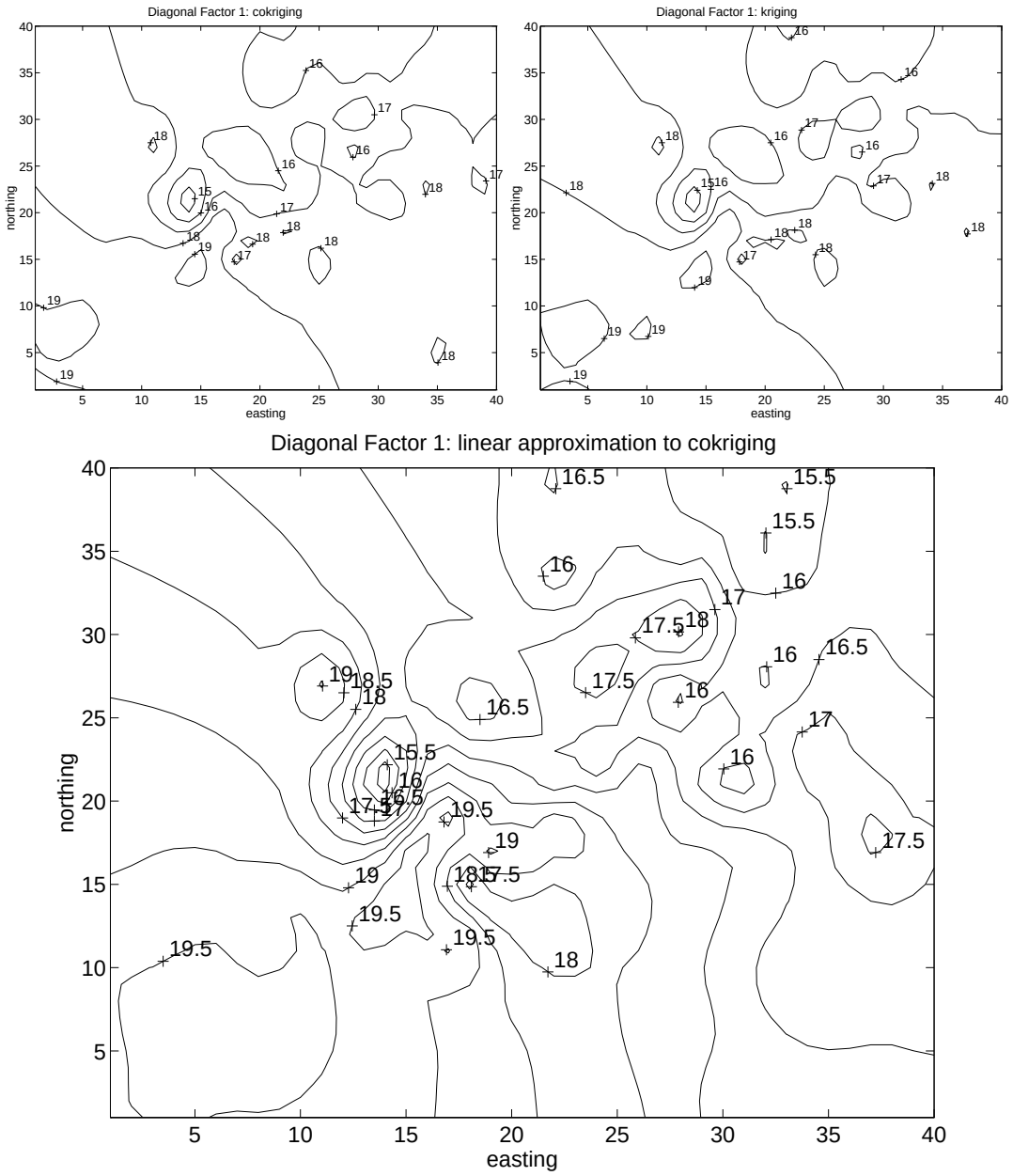


FIGURE 6.7. Maps obtained using the new cokriging method (described in the Chapter on kriging), kriging, and the linear approximation to cokriging. The linear approximation failed to approximate the cokriging map well, but this result may simply indicate that the the norms of the matrices related to the cross-variogram were too large.

TSVD as a multivariate data analysis technique. The data set for the Nitrate project was broken into 3 periods; for those periods, there were 34 sites common to all three sets, for which 8 non-geographical variables are reported: bicarbonate, calcium, magnesium, sodium, sulfate, chloride, nitrates, and depth-to-water. This constitutes a $3 \times 8 \times 34$ three-tensor.

The Matlab procedure *unsymsort.m* (found in the appendix) was used to decompose this unsymmetric tensor. For comparison purposes, a decomposition using the SVD was also generated (on the two smallest dimensions separately) and results were compared. In other words, two matrices were created from the three-tensor: one was 3×3, and the other 8×8:

$$P_{3\times 3} \equiv \langle T_{ikl}, T_{jkl} \rangle,$$

and

$$Q_{8\times 8} \equiv \langle T_{kil}, T_{kjl} \rangle.$$

From these were obtained two sets of singular vectors, which together serve as a basis for the two space of 3×8 matrices. These basis matrices, pairs $p_i \otimes q_j$, then multiplied the tensor, generating a set of vectors r_{ij} in the Z-dimension, and a set λ_{ij} of "coordinates" for the decomposition:

$$\langle p_i \otimes q_j, T \rangle = \lambda_{ij} r_{ij}.$$

The results are compared in Figure (6.8).

The successive SVD method is faster, retains all the information, and is easier to characterize, as the decomposition information is contained in the sets of vectors and scalars

$$\{p_i\}_{i=\{1,\cdots,p\}},\ \{q_j\}_{j=\{1,\cdots,q\}},\ \{r_{ij}\},\ \text{and}\{\lambda_{\text{ij}}\},$$

whereas the TSVD may require additional p and q information: its decomposition information may require storage in sets

$$\{p_{ij}\}_{i=\{1,\cdots,p\},j=\{1,\cdots,q\}},\ \{q_{ij}\}_{i=\{1,\cdots,p\},j=\{1,\cdots,q\}},\ \{r_{ij}\},\ \text{and}\{\lambda_{\text{ij}}\}.$$

However, the low-rank approximation offered by the TSVD was better than that of the SVD method, judging by the percentage of representation

$$\frac{\sum_{i=1}^{rank}(\lambda_i^2 - \mu_i^2)}{\sum_{i=1}^{rank} \mu_i^2}. \tag{6.6.3}$$

where the λ_i are the singular values, and the μ_i correspond to the rank-one tensors given by the SVD method.

As algorithms are improved, speed should also; and it may be that, in the general case, the TSVD will be able to achieve much more than the meager 2% advantage it had at rank eight. The TSVD showed better results at certain ranks in tests with

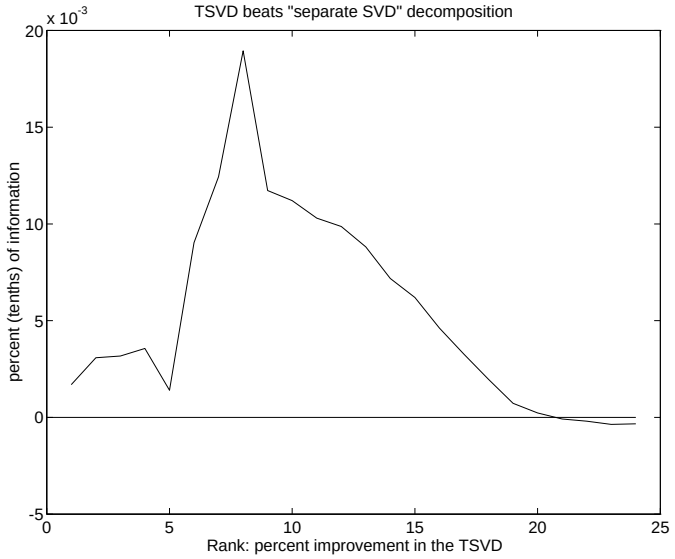


FIGURE 6.8. Comparison of the total "diagonal" representations by a separate SVD, and by the TSVD. The TSVD does better at representing the information, up to rank 21, but does not quite capture all the information in the original tensor (accounting for the dip at the end).

random outer-product tensors (e.g. tensors formed as the sum of four random outer-products); generally much better than 2%, and it also tend to get better "recovery" (not so much loss at the last ranks). Perhaps the strong correlation structure of the Nitrate study data has an especially adverse effect on the power method.

Results using the same procedure on a set of such outer-product tensors, with the same size as the Nitrate tensor (i.e., $3 \times 8 \times 34$), are now described. The tensors generated were sums of four outer-products (with four weights from a uniform distribution on [0,1]:

$$T = \sum_{i=1}^{4} w_i p_i \otimes q_i \otimes r_i,$$

where p_i, q_i, and r_i are random unit vectors). Figure (6.9) shows the histogram of the improvements TSVD achieved over the SVD method for the 100 runs; and in Table (6.6), the summary statistics are gathered. Two items are depicted: the maximum improvement offered by the TSVD (for any rank, for a particular tensor), and the minimum (which were all negative, implying that the SVD method outperformed the TSVD method at some point, invariably at the tail end, as seen in Figure (6.8)). The TSVD gain is given by (6.6.3). As the TSVD failed to reconstitute the tensor, there is a slight dip at the end in Figure (6.9).

Overall the TSVD outperformed the separate SVD method on this task, at times achieving up to about 70% more information over its similarly-ranked SVD cousin.

Best Improvement	mean	max	min	σ
%	16.03	72.70	00.21	15.89
Worst Loss	mean	max	min	σ
%	-00.04	(-)00.39	(-)00.00	00.06

TABLE 6.5. Results of 100 runs, for random $3 \times 8 \times 34$ tensors (in percent).

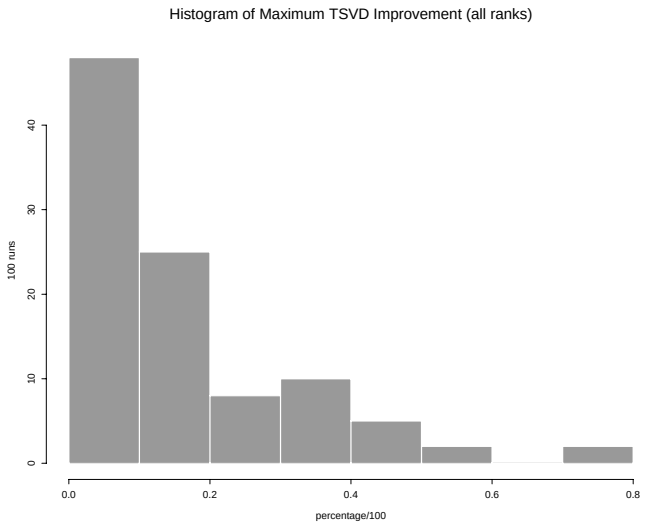


FIGURE 6.9. The TSVD maximized the representation of the tensors for some fixed rank in each case, as shown in this histogram of the improvements TSVD achieved.

The TSVD failed to account for at most half a percent of the information in the tensors, as measured by the Frobenius norm.

Chapter 7

Conclusion

The Singular Value Decomposition is a powerful tool, utilized in many techniques of modern mathematics. Its applications are extremely diverse, but the prevalence and importance of linear problems are the primary reasons that the SVD turns up so often.

As we have shown, the SVD is essential in techniques such as Principal Components Analysis and Correspondence Analysis; the solution of large, ill-conditioned linear systems; image processing, compression, and analysis; variogram analysis; rapid interpolation; and in many other areas. Its usefulness gives it a special place in statistics and geostatistics, and much of mathematics in general.

We have shown that the SVD can be generalized, from the familiar matrix case to the more general three-tensor case (the TSVD). Properties of the SVD led us to the TSVD: we merely followed parallels between the two decompositions, generalizing useful properties of the SVD (such as its rapid interpolation property). We have shown how to define the Tensor SVD, that it exists, at least in the three-dimensional case, and that the singular tensors of a three-tensor are bi-orthogonal.

Furthermore, we have shown the link which exists between the TSVD and the solution of the near-simultaneous diagonalization problem of Flury. It has been shown that the TSVD may accomplish essentially the same decomposition in problems of geostatistical importance, especially variogram modelling in the important case of the linear coregionalization model. It goes beyond the near-simultaneous diagonalization problem, however, as the TSVD decomposes arbitrary tensors, not just tensors whose layers are symmetric or even positive definite matrices.

Other applications of the TSVD in the theory of approximation and estimation have been described, as well as some of the applications which motivated us to pursue a tensor generalization of the SVD: one problem came from categorical statistics, and another from data estimation.

While a power algorithm has been presented for the calculation of the TSVD (in the three-tensor case), it is clear that we will need better algorithms before this decomposition can be put to serious use.

The three-tensor of most interest to geostatisticians is the sample variogram tensor, which is actually a stack of positive-definite matrices, each representing a sample variogram value for a given distance. We have shown how the sample variogram tensor is precisely a spatial decomposition (when properly weighted) of the sample covariance matrix. We have shown how one can begin to use this knowledge when choosing variables for a cokriging system.

Xie's algorithm and the TSVD decomposition of the variogram tensor lead to a new set of variables, linear combinations of the original set of variables, which have the property that their cross-variograms are small. One can then hope to krige these (relatively) uncorrelated variables, then retransform to the original variables, in lieu of cokriging. This idea is an extension of the factorial kriging model, but is more appropriate as it focuses on reducing the cross-variogram, rather than the correlation at lag zero.

Small cross-variograms may allow the use of a first-order approximation of the cokriging results: this approximation arose out of the formal solution of the new formulation of the cokriging equations, presented herein.

This new formulation is, in one sense, merely a permutation of Myers's system: as we have shown, the two-variable system is given in the two formulations by

$$\begin{bmatrix} V & F \\ F^T & 0 \end{bmatrix} \begin{bmatrix} \Gamma \\ \mu \end{bmatrix} = \begin{bmatrix} V^x \\ F^x \end{bmatrix} \iff \begin{bmatrix} K_{11} & C_{12} \\ C_{12} & K_{22} \end{bmatrix} \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} = \begin{bmatrix} K_1^x & C_{12}^x \\ C_{12}^x & K_2^x \end{bmatrix}.$$

The advantage of writing the system of equations in terms of the permuted system is that certain steps toward inversion are already carried out, and the solution vectors, or weights, obtained do double duty by also providing the solution to the kriging systems K_{11} and K_{22}. Certain sub-system cokriging solutions are also obtained. Furthermore, the inversion process displays connections to the Cauchy-Schwartz condition in the matrix setting, and the matrices which one must invert are smaller, and better conditioned.

From our investigations of kriging we have discovered that there is a strong relationship between kriging and kernel estimation. The spline, which we know to be a special case of kriging, has been thoroughly explored in the literature, and the kernel of the smoothing spline identified. In the case of variogram models, other than the nugget and the linear model without nugget, no such identification has yet been made.

Even so, it appears that one can at least approximate the action of a kriging system on a well-distributed set of data by a kernel, a method which has obvious advantages over the large linear systems of the kriging equations. We have shown example kernels which appear to be analogues of the standard variogram models, and examined qualitative features of others.

Finally, several methods described herein were applied to a data set consisting of well samples from an area around Phoenix, Arizona. Demonstrations were given of automated variogram and cross-variogram modelling, variogram matrix modelling via the TSVD in the case of coregionalization, the linear approximation to cokriging, TSVD diagonalization of common sites through time, and cross-validation of maps. These were given in such a way that interested readers should be able to carry out the procedures on their own, and, as an aid to those so inclined, some MATLAB routines are included in the appendix.

We hope that the results presented herein represent some small contribution toward the betterment of geostatistical analysis.

Appendix A

TSVD Calculations

A.1 Symmetric Power Method for Finding Singular Tensors

A.1.1 Main Routine

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  This is a program for calculating the symmetric singular tensors of a three-
%  tensor.  Tensors may be read in as a set of outer-products, or as a file
%  containing the tensor in patches, p x q, of which there are r. These
%  dimensions should be at the top of the file, in the first line.
%

%                          set up some of the necessary declarations:

vector = 0;
demand_orthogonality = 1;
maxiterations = 200;
normeps = 1e-6;
eps = 1e-30;
filename='../tdiss.dat';

%                          read in the tensor and dimensions:

if vector == 1
  tensor=ten_tile(u1,u1,u3,lambda,1);
else
  % it is assumed that the patches are read in as pxp layers in filename:
  % in the top row of the file are four numbers: p, p, r, and any other
  % integer.
  [tensor,p,q,r]=tensor_as_patches(filename);
end

maxtrips = p*p*p;
diffs=zeros(maxtrips,maxiterations);
pkeep=zeros(p,maxtrips);
lambdas=zeros(1,maxtrips);
rkeep=zeros(r,maxtrips);
rvec=zeros(r,1);

%                          scale the tensor:

tensornorm=norm(tensor,'fro');
tensor=tensor/tensornorm;
ntrips = 0;

%
%                          Start the power iterations:
%
for i = 1:maxtrips
%
%                          Create the four tensor:
%
  tensor2=patch_ten_mat(tensor,p);
```

```
  for l = 1:p
    ll=(l-1)*p;
    for m = 1:p
      mm=(m-1)*p;
      taid=tensor2(:,(l-1)*p+m);
      for j=1:p
        for k=1:p
          fourtensor(ll+j,mm+k)=taid'*tensor2(:,(j-1)*p+k);
        end
      end
    end
  end
%
%                           Randomize the initial guess:
%
  pvec=randn(p,1);
  pvec=normr(pvec')';
  pback=pvec;
%
%                           Iterate on the four tensor:
%
  for j = 1:maxiterations

    outer=pvec*pback'+ pback*pvec';
    pvec=pback;

    for k = 1:p
      kk = (k-1)*p+1;
      for l = k:p
        ll = (l-1)*p+1;
        prod(k,l)=sum(sum(outer.*fourtensor(kk:kk+p-1,ll:ll+p-1)));
        prod(l,k)=prod(k,l);
      end
    end

    pback=prod*pvec;
    pback=normr(pback')';

    if demand_orthogonality == 1
      for k=1:ntrips
        pback = pback-pkeep(:,k)'*pback*pkeep(:,k);
        pback=normr(pback')';
      end
    end

    diff=1-abs(pvec'*pback);
    diffs(j,i)=diff;
    if diff < eps
      break;break;
    end

  end
  fprintf('Diff: %f;\n',diff);
%
%                           Compute the r vector, and other things:
%
  iterates=j;
  pvec=pback;
%
%                           Deflate the tensor:
%
  if demand_orthogonality == 1
```

```
    for k=1:ntrips
      pvec = pvec-pkeep(:,k)'*pvec*pkeep(:,k);
      pvec=normr(pvec')';
    end
  end
  outer=pvec*pvec';

  for j = 1:r
    ll=p*(j-1);
    temp=tensor(ll+1:ll+p,1:p);
    rvec(j)=pvec'*temp*pvec;
    temp = temp - rvec(j)*outer;
    if demand_orthogonality == 1
      outerprod=(temp*pvec)*pvec';
      temp=temp - (outerprod + outerprod');
    end
    tensor(ll+1:ll+p,1:p)=temp;
  end
%
%                           Store our values:
%
  pkeep(:,i)=pvec;
  lambdas(i)=norm(rvec);
  rkeep(:,i)=rvec/lambdas(i);
%
%                           Inform user, and check for finish:
%
  ntrips = ntrips + 1;
  newnorm=norm(tensor,'fro');
  fprintf('Trip %d, newnorm %f; %d iterations.\n',i,newnorm,iterates);
  if newnorm < normeps
    break
  end

  if demand_orthogonality == 1
    if ntrips == p
      break;
    end
  end
end
%
%                           If we're demanding orthogonality, then compute
%                           the other products:
%
if demand_orthogonality == 1
  % Need to reread original tensor, as it is completely deflated:
  if vector == 1
    tensor=ten_tile(u1,u1,u3,lambda,1);
  else
    [tensor,p,q,r]=tensor_as_patches(filename);
  end
  tensor=tensor/tensornorm;
  ii = ntrips;
  qkeep=pkeep;
  for iloop = 1:ntrips-1
    for jloop = iloop+1:ntrips
      ii=ii+1;
      pkeep(:,ii)=pkeep(:,iloop);
      qkeep(:,ii)=pkeep(:,jloop);
      outer=pkeep(:,ii)*qkeep(:,ii)';
      for j = 1:r
        ll=p*(j-1);
```

```
        rvec(j)=sum(sum(outer.*tensor(ll+1:ll+p,1:p)));
        tensor(ll+1:ll+p,1:p)=tensor(ll+1:ll+p,1:p) - rvec(j)*outer;
      end
      lambdas(ii)=norm(rvec);
      rkeep(:,ii)=rvec/lambdas(ii);
      ii=ii+1;
      pkeep(:,ii)=pkeep(:,jloop);
      qkeep(:,ii)=pkeep(:,iloop);
      lambdas(ii)=lambdas(ii-1);
      rkeep(:,ii)=rkeep(:,ii-1);
    end
  end
  qkeep=qkeep(:,1:ii)
end
pkeep=pkeep(:,1:ii)
rkeep=rkeep(:,1:ii)
lambdas=lambdas(1:ii)*tensornorm
diffs=diffs(:,1:ntrips);
%
%                          Plot out some results:
%
%                  This shows how we did on convergence:
%
plot(diffs);drawnow;pause
%
%                  These are the new coordinates (as in the text):
%
plot(pkeep);drawnow;pause
%
%                  These are the new variograms:
%
plot(rkeep*diag(lambdas));drawnow;pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A.1.2 Sub-Routines

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% ten_tile.m: Converts outer-products representing the tensor into a tiled
%             matrix.
%
%                function [mat]=ten_tile(u1,u2,u3,lambda,cols)
%
function [mat]=ten_tile(u1,u2,u3,lambda,cols)

p=size(u1);
q=size(u2);
r=size(u3);
p=p(1);
q=q(1);
r=r(1);

num=size(lambda);
num=num(1);
if nargin ~= 5
  cols=r;
end
w3=u3(:,1:num)*lambda;
mat=zeros(r*p,cols*q);
%
% fill the first column: then symmetrize it:
```

```
%
l=0;
for m = 1:num
  for k = 1:r
    l=(k-1)*p;
    hold1=w3(k,m);

    for i = 1:p
      hold=u1(i,m)*hold1;
      for j = 1:q
        mat(l+i,j) = mat(l+i,j) + hold*u2(j,m);
      end
    end
  end
end
%
% Now do the rest of the columns:
%
for k = 1:r
  l=(k-1)*p;
  ll=l;
  for n = 1:cols-1
    nn = n*q;
    ll = ll - p;
    if ll < 0
      ll = (r-1)*p;
    end
    for i = 1:p
      for j = 1:q
        mat(ll+i,nn+j) = mat(l+i,j);
      end
    end
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% patch_ten_mat.m: Converts patch tensor into a matrix representing the tensor.
%
%                function [mat]=patch_ten_mat(mat,p)
%
function [mat2]=patch_ten_mat(mat,p)

matsize=size(mat);
q=matsize(2);
r=matsize(1)/p;

mat2=zeros(r,p*q);
for k = 1:r
  l=0;
  m=(k-1)*p;
  for i = 1:p
    for j = 1:q
      l=l+1;
      mat2(k,l) = mat(m+i,j);
    end
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% tensor_as_patches.m: reads in a tensor as patches.  Must have four
```

```
%               integers at the top, indicating p x q x r x anything!
%
%        function [tensor,p,q,r] = tensor_as_patches(filename)
%
function [tensor,p,q,r] = tensor_as_patches(filename)

fid=fopen(filename,'r');
vals=fscanf(fid,'%d %d %d %d',4);
p=vals(1);
q=vals(2);
r=vals(3);
vals
tensor=zeros(q,r*p);
tensor(:)=fscanf(fid,'%f ');
tensor=tensor';
```

A.2 Power Method for Unsymmetric Tensors

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                           UNSYMSORT.M
%
%    Unsymsort is a modified algorithm, which seeks to compare the SVDs of the
% various subsystems with the result obtained by restricting to the completely
% orthogonal space. In the event that the subspace shows a bigger (or at
% least almost bigger) subspace, then we use it instead.
%
%                           SET UP SOME OF THE NECESSARY DECLARATIONS:
%
%    vector decides whether we use (already defined) vectors, contained in
% matrices  u1, u2, and u3, of dimensions p, q, r (which also must be
% defined before starting); or whether we use a file containing "patches" of
% the tensor. vector = 1 => using vectors; else using a file of patches.
%
vector = 0;
%
%    maxiterations is the number of passes the program will make, before
% quitting from fatigue.
%
maxiterations = 200;
%
%    normeps will test the remaining norm in the matrix: if small, we
% consider it diagonalized.
%
normeps = 1e-6;
%
%    eps is a test for the difference in each pass of the maxiterations
% passes, as a test for convergence.
%
eps = 1e-30;
%
%    we introduce a fudge factor to allow the program to take a previously
% obtained combination of vectors over a newly found one, in case the
% difference is small: this is to avoid moving back into already eliminated
% space (a problem with power methods).
%
fudge = .97;
filename='../tdiss.dat';   % this is the tensor in Xie's dissertation.
%
%                           READ IN THE TENSOR AND DIMENSIONS:
%
```

```
if vector == 1
%
%    Again, if we're using this option, one must define the vectors and a
% diagonal matrix lambda, which contains the weights of their outer
% products, ahead of time.  p, q, and r must also be set.
%
  tensor=ten_tile(u1,u2,u3,lambda,1);
else
%
%    It is assumed that the patches are read in as pxp layers in filename:
% in the top row of the file are four numbers: p, p, r, and any other integer.
%
  [tensor,p,q,r]=tensor_as_patches(filename);
end
%
%    maxtrips should probably be the known maximum rank, although with the
% "drift" known to occur with power methods, this can be exceeded.
%
maxtrips = p*q;
%
%                           ZERO OUT THE ARRAYS WE NEED:
%
diffs=zeros(maxtrips,maxiterations);
lams=zeros(maxtrips,3);
prod=zeros(p,q);
lambdas=zeros(1,maxtrips);
pkeep=zeros(p,maxtrips);
qkeep=zeros(q,maxtrips);
rkeep=zeros(r,maxtrips);
pvec=zeros(p,1);
qvec=zeros(q,1);
rvec=zeros(r,1);
fourtensor=zeros(p*p,q*q);
%
%    Save the tensor (as a matrix) for later, to be shown in pictures:
%
gar=patch_ten_mat(tensor,p);
%
%                           SCALE THE TENSOR:
%
tensornorm=norm(tensor,'fro');
tensor=tensor/tensornorm;
%
%                           START THE POWER ITERATIONS:
%
ntrips = 0;
for i = 1:maxtrips
%
%                           CREATE THE FOUR TENSOR:
%
  tensor2=patch_ten_mat(tensor,p);
  kk = -1;
  for l = 0:q:(p-1)*q
    kk=kk+1;
    for m = 1:q
      mm=(m-1)*q;
      taid=tensor2(:,l+m);
      for j=1:p
        ll = kk*p+j;
        jj = (j-1)*q;
        for k=1:q
          fourtensor(ll,mm+k)=taid'*tensor2(:,jj+k);
```

```
      end
    end
  end
end
%
%                          RANDOMIZE THE INITIAL GUESS:
%
pvec=randn(p,1);
pvec=normr(pvec')';
qvec=randn(q,1);
qvec=normr(qvec')';
%
%                          ITERATE ON THE FOUR TENSOR:
%
for j = 1:maxiterations

  outer=pvec*qvec';
  for k = 1:p
    kk = (k-1)*p;
    for l = 1:q
      ll = (l-1)*q;
      temp = fourtensor(kk+1:kk+p,ll+1:ll+q);
      prod(k,l)=pvec'*temp*qvec;
    end
  end

  qback=pvec'*prod;
  qback=normr(qback)';
  diff=1-abs(qvec'*qback);

  pvec=prod*qvec;
  pvec=normr(pvec')';
  qvec=qback;

  diffs(j,i)=diff;
  if diff < eps
    break;break;
  end

end
iterates=j;
fprintf('Diff: %f;\n',diff);
%
%                          COMPUTE THE R VECTOR, AND OTHER THINGS:
%
for j = 1:r
  jj=p*(j-1);
  temp=tensor(jj+1:jj+p,1:q);
  rvec(j)=pvec'*temp*qvec;
end
lambdas(i)=norm(rvec);
rkeep(:,i)=rvec/lambdas(i);
qkeep(:,i)=qvec;
pkeep(:,i)=pvec;
%
%                          COMPUTE SVDS FOR ALL PREVIOUSLY OBTAINED TENSORS:
%
ikeep=0;
jkeep=0;
for k=ntrips:-1:1

  pvec=pkeep(:,k);
```

```
    mat=zeros(r,q);
    for j = 1:r
      jj=p*(j-1);
      mat(j,1:q)=pvec'*tensor(jj+1:jj+p,1:q);
    end
    [u,s,v]=svd(mat,0);
    lams(k,1)=s(1,1);

    qvec=qkeep(:,k);
    mat=zeros(r,p);
    for j = 1:r
      jj=p*(j-1);
      mat(j,1:p)=(tensor(jj+1:jj+p,1:q)*qvec)';
    end
    [u,s,v]=svd(mat,0);
    lams(k,2)=s(1,1);

    rvec=rkeep(:,k);
    mat=zeros(p,q);
    jj=p*(r-1);
    for j = 1:p
      mat(j,1:q)=rvec'*tensor(j:p:jj+j,1:q);
    end
    [u,s,v]=svd(mat,0);
    lams(k,3)=s(1,1);

  end

  if ntrips > 0
%
%    then we sort the lambdas, from the SVDs, looking for the max:
%
    [lamsort,index]=sort(lams);
%
%    We then sort the three maxima, finding out which column had the maximum
% value:
%
    [lamsmax,column]=sort(lamsort(maxtrips:maxtrips,:));
    fprintf('lambdas: new: %f; used: %f %f %f\n',lambdas(i),lamsmax)
    if lamsmax(3) > fudge*lambdas(i)
%
%    We recompute the SVD for the maximal trip, and column, and use it instead:
%
      if column(3) == 3
        rvec=rkeep(:,index(maxtrips,3));
        mat=zeros(p,q);
        jj=p*(r-1);
        for j = 1:p
          mat(j,1:q)=rvec'*tensor(j:p:jj+j,1:q);
        end
        [u,s,v]=svd(mat,0);
        pkeep(:,i)=u(:,1);
        qkeep(:,i)=v(:,1);
        rkeep(:,i)=rvec;

      elseif column(3) == 1
        pvec=pkeep(:,index(maxtrips,1));
        mat=zeros(r,q);
        for j = 1:r
          jj=p*(j-1);
          mat(j,1:q)=pvec'*tensor(jj+1:jj+p,1:q);
        end
```

```
        [u,s,v]=svd(mat,0);
        pkeep(:,i)=pvec;
        qkeep(:,i)=v(:,1);
        rkeep(:,i)=u(:,1);

      else
        qvec=qkeep(:,index(maxtrips,2));
        mat=zeros(r,p);
        for j = 1:r
          jj=p*(j-1);
          mat(j,1:p)=(tensor(jj+1:jj+p,1:q)*qvec)';
        end
        [u,s,v]=svd(mat,0);
        pkeep(:,i)=v(:,1);
        qkeep(:,i)=qvec;
        rkeep(:,i)=u(:,1);
      end
      fprintf('Winner: %d %d\n',index(maxtrips,column(3)),column(3))
    else
      fprintf('Winner: new tensor.\n')
    end
  else
    fprintf('Winner: new tensor.\n')
  end
%
%                           DEFLATE THE TENSOR AND STORE THE WINNER:
%
  pvec=pkeep(:,i)
  qvec=qkeep(:,i)
  outer=pvec*qvec';
  for j = 1:r
    jj=p*(j-1);
    temp=tensor(jj+1:jj+p,1:q);
    rvec(j)=pvec'*temp*qvec;
    tensor(jj+1:jj+p,1:q)=temp - rvec(j)*outer;
  end
  lambdas(i)=norm(rvec);
  rkeep(:,i)=rvec/lambdas(i);
%
%                           INFORM USER, AND CHECK FOR FINISH:
%
  ntrips = ntrips + 1;
  newnorm=norm(tensor,'fro');
  fprintf('Trip %d, newnorm %f; %d iterations.\n',i,newnorm,iterates);
  if newnorm < normeps
    break
  end
end
%
%    Restrict these guys to the number actually used:
%
pkeep=pkeep(:,1:i)
qkeep=qkeep(:,1:i)
rkeep=rkeep(:,1:i)
diffs=diffs(:,1:ntrips);
%
%    Rescale the lambdas, as we scaled the tensor originally:
%
lambdas=lambdas(1:i)*tensornorm
%
%                           PLOT OUT SOME RESULTS:
%
```

```
%                  THIS SHOWS HOW WE DID ON CONVERGENCE, IN ITERATING:
%
figure (1)
plot(diffs);drawnow;pause
%
%                  THESE ARE THE NEW COORDINATES (AS IN THE TEXT):
%
plot(pkeep);drawnow;pause
%
%                  THESE ARE THE NEW VARIOGRAMS, IF IT'S THAT SORT OF PROBLEM:
%
rkeep=rkeep*diag(lambdas);
plot(rkeep);drawnow;pause
rkeep=rkeep*inv(diag(lambdas));
for i=1:length(lambdas)
  bage=ten_mat(pkeep,qkeep,rkeep,diag(lambdas(1:i)));
  plot(gar,'w')
  hold on
  plot(bage,'y')
  hold off
  drawnow
  pause
end
delete(1)

\subsection{Additional Sub-Routines}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% ten_mat.m: Converts outer-products into a matrix representing the tensor.
%
%                function [mat]=ten_mat(u1,u2,u3,lambda)
%
function [mat]=ten_mat(u1,u2,u3,lambda)

length_u1=size(u1);
length_u2=size(u2);
length_u3=size(u3);

num=size(lambda);
num=num(2);

length_u1=length_u1(1);
length_u2=length_u2(1);
length_u3=length_u3(1);

w3=u3(:,1:num)*lambda;

mat=zeros(length_u3,length_u1*length_u2);
for k = 1:num
  l=0;
  for i = 1:length_u1
    for j = 1:length_u2
      l=l+1;
      mat(:,l) = mat(:,l)+u1(i,k)*u2(j,k)*w3(:,k);
    end
  end
end
```

Matlab Demo of New Cokriging Equations

The following program is a Matlab ".m" file: Matlab will run it like a script if it

is in the directory in which Matlab is running, if the user types:

```
>> condition
```

In particular note how close the linear approximation gets to the true values of the cross-weights from the cokriging scheme in the sample run. The reader can watch this deteriorate as the values of the coefficients .01 and .02 of the line

```
cv(i,j) = .02*k1(i,j) + .01*k2(i,j);
```

below are increased, resulting in a larger norm of the cross-terms matrices norm_AB and norm_BA.

```
---------------------------------------------------------------
                  Input to Matlab: condition.m
---------------------------------------------------------------
%
%       We present an example case, of ordinary cokriging, with three
% sites and a couple of standard models.
%

fprintf('\nThese are the sites we will be using:\n')

sites=rand(1,3)
x0=.2

range1=3;
range2=3;
sill1=1;
sill2=1;

for i = 1:3
  for j = 1:3
    distance=abs(sites(i)-sites(j));
    k1(i,j) = sill1*(1-exp(-(distance*log(20)/range1)^2));
    k2(i,j) = sill2*(1-exp(-distance*log(20)/range2));
    cv(i,j) = .02*k1(i,j) + .01*k2(i,j);
  end
end

v10=ones(4,1);
v20=ones(4,1);
cv0=zeros(4,1);
for i = 1:3
  distance=abs(sites(i)-x0);
  v10(i) = sill1*(1-exp(-(distance*log(20)/range1)^2));
  v20(i) = sill2*(1-exp(-distance*log(20)/range2));
  cv0(i) = .02*v10(i) + .01*v20(i);
end

fprintf('\nThis is the Myers system cokriging matrix:\n')

I=eye(2,2);
Z=zeros(2,2);
vmat12= [k1(1,2) cv(1,2);cv(1,2) k2(1,2)];
vmat13= [k1(1,3) cv(1,3);cv(1,3) k2(1,3)];
vmat23= [k1(2,3) cv(2,3);cv(2,3) k2(2,3)];

X = [
```

```
     Z vmat12 vmat13     I
vmat12      Z vmat23     I
vmat13 vmat23      Z     I
     I      I      I     Z
]

fprintf('and the permutation matrix P:\n')

P = [
1 0 0 0 0 0 0 0;
0 0 1 0 0 0 0 0;
0 0 0 0 1 0 0 0;
0 0 0 0 0 0 1 0;
0 1 0 0 0 0 0 0;
0 0 0 1 0 0 0 0;
0 0 0 0 0 1 0 0;
0 0 0 0 0 0 0 1;
]

pause
fprintf('C is the permuted matrix: C=P*X*Pt\n')

C=P*X*P'

fprintf('Here are its subcomponent matrices:\n')

K1 = [ C(1:4,1) C(1:4,2) C(1:4,3) C(1:4,4) ]

CR = [ C(5:8,1) C(5:8,2) C(5:8,3) C(5:8,4) ]

K2 = [ C(5:8,5) C(5:8,6) C(5:8,7) C(5:8,8) ]

pause
fprintf('Here are some useful matrices: (inverse of K1 and K2, etc.)\n')

K1inv=inv(K1)
K2inv=inv(K2)

A=K1inv*CR
B=K2inv*CR

pause
fprintf('Have a look at the condition numbers of the relevant matrices:\n')
fprintf('we should see that the conditions of K1 and K2 are less than C.\n')

condC=cond(C)

condK1=cond(K1)
condK2=cond(K2)

cond_IminusAB=cond(eye(4)-A*B)
norm_AB=norm(A*B)
cond_minusBA=cond(eye(4)-B*A)
norm_BA=norm(B*A)

pause
fprintf('So we invert I-AB and I-BA rather than C:\n')
M1inv=inv(eye(4)-A*B)
M2inv=inv(eye(4)-B*A)

Cinv=inv(C)
```

```
pause
fprintf('Define the RHS (made up variograms and cross-variograms at x0):\n')

RHS=[
 v10  cv0
 cv0  v20
]

pause
fprintf('\n      Now we compare results: first using the small systems:\n')

krig1=K1inv*v10;
krig2=K2inv*v20;

G1=M1inv*(K1inv*v10-A*K2inv*cv0);
g1=K2inv*cv0-B*G1;
G2=M2inv*(K2inv*v20-B*K1inv*cv0);
g2=K1inv*cv0-A*G2;

cokriging_weights=[
    G1 g2
    g1 G2
    ]

fprintf('And then using the large Myers System:\n')

cokriging_weights=Cinv*RHS

fprintf('Identical!\n')

fprintf('\n      By the way, why not have a look at the kriging weights:\n')

kriging_weights=[
    krig1 zeros(4,1)
    zeros(4,1) krig2
    ]


fprintf('\n      And here is the linear approximation to the cokriging weights:\n')

cheap_correction=[
    krig1 -K1inv*(CR*krig2 - cv0)
    -K2inv*(CR*krig1 - cv0) krig2
    ]

------------------------------------------------------------
                 Output from a run:
------------------------------------------------------------


These are the sites we will be using:

sites =

    0.0077    0.3834    0.0668


x0 =

    0.2000
```

```
This is the Myers system cokriging matrix:

X =

  Columns 1 through 7

         0         0    0.1313    0.0058    0.0035    0.0006    1.0000
         0         0    0.0058    0.3128    0.0006    0.0573         0
    0.1313    0.0058         0         0    0.0951    0.0046    1.0000
    0.0058    0.3128         0         0    0.0046    0.2710         0
    0.0035    0.0006    0.0951    0.0046         0         0    1.0000
    0.0006    0.0573    0.0046    0.2710         0         0         0
    1.0000         0    1.0000         0    1.0000         0         0
         0    1.0000         0    1.0000         0    1.0000         0

  Column 8

         0
    1.0000
         0
    1.0000
         0
    1.0000
         0
         0

and the permutation matrix P:

P =

     1     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0
     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     1     0
     0     1     0     0     0     0     0     0
     0     0     0     1     0     0     0     0
     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     1

C is the permuted matrix: C=P*X*Pt

C =

  Columns 1 through 7

         0    0.1313    0.0035    1.0000         0    0.0058    0.0006
    0.1313         0    0.0951    1.0000    0.0058         0    0.0046
    0.0035    0.0951         0    1.0000    0.0006    0.0046         0
    1.0000    1.0000    1.0000         0         0         0         0
         0    0.0058    0.0006         0         0    0.3128    0.0573
    0.0058         0    0.0046         0    0.3128         0    0.2710
    0.0006    0.0046         0         0    0.0573    0.2710         0
         0         0         0         0    1.0000    1.0000    1.0000

  Column 8

         0
         0
         0
         0
    1.0000
    1.0000
```

```
    1.0000
         0

Here are its subcomponent matrices:

K1 =

         0    0.1313    0.0035    1.0000
    0.1313         0    0.0951    1.0000
    0.0035    0.0951         0    1.0000
    1.0000    1.0000    1.0000         0


CR =

         0    0.0058    0.0006         0
    0.0058         0    0.0046         0
    0.0006    0.0046         0         0
         0         0         0         0


K2 =

         0    0.3128    0.0573    1.0000
    0.3128         0    0.2710    1.0000
    0.0573    0.2710         0    1.0000
    1.0000    1.0000    1.0000         0

Here are some useful matrices: (inverse of K1 and K2, etc.)

K1inv =

   1.0e+03 *

   -0.7488   -0.1288    0.8776    0.0149
   -0.1288   -0.0274    0.1562    0.0031
    0.8776    0.1562   -1.0338   -0.0169
    0.0149    0.0031   -0.0169   -0.0003


K2inv =

   -8.7525    0.2510    8.5015    0.4339
    0.2510   -1.8520    1.6010    0.4876
    8.5015    1.6010  -10.1025    0.0785
    0.4339    0.4876    0.0785   -0.1570


A =

   -0.1768   -0.2611   -1.0757         0
   -0.0573   -0.0207   -0.2093         0
    0.2341    0.2817    1.2850         0
    0.0067    0.0075    0.0236         0


B =

    0.0069   -0.0112   -0.0045         0
   -0.0096    0.0088   -0.0084         0
    0.0027    0.0023    0.0129         0
    0.0029    0.0029    0.0025         0
```

```
Have a look at the condition numbers of the relevant matrices:
we should see that the conditions of K1 and K2 are less than C.

condC =

   3.5775e+03


condK1 =

   3.2668e+03


condK2 =

   35.3597


cond_IminusAB =

    1.0181


norm_AB =

    0.0179


cond_minusBA =

    1.0181


norm_BA =

    0.0179

So we invert I-AB and I-BA rather than C:

M1inv =

    0.9983   -0.0029   -0.0110         0
   -0.0008    1.0000   -0.0023         0
    0.0024    0.0029    1.0132         0
    0.0000    0.0000    0.0002    1.0000


M2inv =

    0.9983   -0.0029   -0.0110         0
   -0.0008    1.0000   -0.0023         0
    0.0024    0.0029    1.0132         0
   -0.0001   -0.0001   -0.0004    1.0000


Cinv =

   1.0e+03 *

  Columns 1 through 7
```

```
   -0.7568   -0.1302    0.8870    0.0150    0.0077    0.0013   -0.0090
   -0.1302   -0.0277    0.1579    0.0031    0.0013    0.0003   -0.0016
    0.8870    0.1579   -1.0449   -0.0171   -0.0090   -0.0016    0.0107
    0.0150    0.0031   -0.0171   -0.0003   -0.0001   -0.0000    0.0002
    0.0077    0.0013   -0.0090   -0.0001   -0.0088    0.0002    0.0086
    0.0013    0.0003   -0.0016   -0.0000    0.0002   -0.0019    0.0016
   -0.0090   -0.0016    0.0107    0.0002    0.0086    0.0016   -0.0102
    0.0003    0.0001   -0.0003   -0.0000    0.0004    0.0005    0.0001

  Column 8

    0.0003
    0.0001
   -0.0003
   -0.0000
    0.0004
    0.0005
    0.0001
   -0.0002

Define the RHS (made up variograms and cross-variograms at x0):

RHS =

    0.0362    0.0025
    0.0330    0.0023
    0.0175    0.0016
    1.0000         0
    0.0025    0.1747
    0.0023    0.1674
    0.0016    0.1245
         0    1.0000


      Now we compare results: first using the small systems:

cokriging_weights =

   -1.1323   -0.0228
    0.2238   -0.0039
    1.9085    0.0267
    0.0002    0.0001
    0.0114    0.0055
    0.0020    0.4208
   -0.0133    0.5737
    0.0001    0.0102

And then using the large Myers System:

cokriging_weights =

   -1.1323   -0.0228
    0.2238   -0.0039
    1.9085    0.0267
    0.0002    0.0001
    0.0114    0.0055
    0.0020    0.4208
   -0.0133    0.5737
    0.0001    0.0102

Identical!
```

```
      By the way, why not have a look at the kriging weights:

kriging_weights =

   -1.1205         0
    0.2259         0
    1.8946         0
   -0.0001         0
         0    0.0052
         0    0.4208
         0    0.5740
         0    0.0102


      And here is the linear approximation to the cokriging weights:

cheap_correction =

   -1.1205   -0.0225
    0.2259   -0.0039
    1.8946    0.0264
   -0.0001    0.0001
    0.0113    0.0052
    0.0019    0.4208
   -0.0132    0.5740
    0.0001    0.0102
```

Case Study Support

.1 Variogram Models

On the next few pages are Tables of the models used in the Nitrate study for the kriging and cokriging of nitrate concentrations. The models were (mostly) the result of using the automated fitting routines described in this dissertation, although some adjustments were made following visual inspections of all models.

The models all satisfied the Cauchy-Schwarz condition (pair-wise) over all lags for which they were used. These cross-variogram models are actually the models of the sum and difference variables, the latter subtracted from the former (which explains the negative sills). Thus we must multiply by .25 to get a cross-variogram model, as Myers showed in [66]:

$$\gamma_{ij} = .25(\gamma_{ij}^{+} - \gamma_{ij}^{-}).$$

As noted in the text, the corhograms and cross-variance maps are (at this stage) mostly useful as a diagnostic tool. They are included for comparison purposes: the corhograms for comparison especially with the corhogram models shown in Figure (3.6), and the cross-variance maps for comparison against the cross-validation results shown in Tables (6.2), (6.2), and (6.2).

variables	model type	sill	major range	minor range
nitrates	nugget	0.42074		
	spherical	0.505778	17.4928	17.4928
	spherical	0.170848	0.856532	0.856532
calcium	nugget	0.669432		
	exponential	0.567845	197.551	197.551
	gaussian	0.878349	88.4874	88.4874
nitrates/ca	nugget	0.889346		
	exponential	0.198518	0.667431	0.667431
	gaussian	0.943697	22.5276	22.5276
	spherical	0.993264	5.13112	5.13112
	linear	-0.300805	30.5527	30.5527
	spherical	-5.16660E-02	10.53693	10.53693
	spherical	-0.255302	2.82123	2.82123
chloride	nugget	0.226366		
	gaussian	0.687097	22.7946	22.7946
	spherical	0.135825	5.32890	5.3289
	gaussian	6.27085E-02	4.45168	4.45168
nitrates/ch	nugget	0.894672		
	gaussian	1.77185	18.3524	18.3524
	spherical	0.385983	4.32405	4.32405
	gaussian	3.35447E-04	3.55479	3.55479
	linear	-0.208631	30.5527	30.5527
	spherical	-0.275018	19.5898	19.5898
	spherical	-0.215388	3.28831	3.28831
magnesium	nugget	0.341738		
	linear	0.216525	30.5527	30.5527
	spherical	0.564624	6.98589	6.98589
nitrates/mg	nugget	1.1038		
	spherical	1.27748	20.3593	20.3593
	spherical	0.773462	5.57295	5.57295
	spherical	-0.217785	9.46021	9.46021
	spherical	-0.287147	3.14974	3.14974
	gaussian	-6.78255E-02	2.60880	2.6088

TABLE 1. Variograms and cross-variograms for 1977 cokrigings, I. Sills for cross-variograms should be multiplied by .25 (these are actually sum-difference variograms of the variables, so the cross-variogram is given by γ_{ij}=.25(sum-diff)).

variables	model type	sill	major range	minor range
pca reg.	nugget	0.302563		
	exponential	1.35605	101.0394	101.0394
	gaussian	0.459727	119.697	119.697
nitrates/pca	nugget	0.543123		
	gaussian	1.47213	22.6026	22.6026
	gaussian	0.203045	4.62081	4.62081
	spherical	0.416445	5.10391	5.10391
	gaussian	0.279664	0.750125	0.750125
	linear	-0.261009	30.5527	30.5527
	spherical	-0.290332	18.6677	18.6677
	spherical	-0.215663	3.07892	3.07892
	gaussian	-5.33472E-02	2.46490	2.4649
lin. reg.	nugget	0.494107		
	exponential	0.287306	26.1072	26.1072
	spherical	0.211852	7.67736	7.67736
	gaussian	5.87922E-02	6.24912	6.24912
nitrates/lr	nugget	1.00481		
	gaussian	1.19216	16.2890	16.289
	gaussian	0.407702	4.06527	4.06527
	spherical	0.492391	0.880435	0.880435
	spherical	0.168212	4.51484	4.51484
	exponential	-6.65601E-02	31.9969	31.9969
	gaussian	-0.136549	9.10415	9.10415
	gaussian	-0.131662	1.84874	1.84874
	spherical	-4.05557E-02	2.09480	2.0948

TABLE 2. Variograms and cross-variograms for 1977 cokrigings, II. Sills for cross-variograms should be multiplied by .25.

variables	model type	sill	major range	minor range
nitrates	nugget	0.289764		
	linear	0.526127	40.4944	40.4944
	gaussian	1.02792E-01	11.5645	11.5645
	spherical	0.162134	2.58734	2.58734
	gaussian	8.96023E-02	2.43956	2.43956
calcium	nugget	0.194048		
	gaussian	1.60796	56.5172	56.5172
	spherical	0.132535	2.83100	2.8310
	gaussian	1.03235E-01	9.13840	9.1384
nitrates/ca	nugget	0.445243		
	gaussian	3.34830	51.2163	51.2163
	spherical	0.563586	2.42568	2.42568
	spherical	0.537779	10.54830	10.5483
	gaussian	2.28766E-02	1.83202	1.83202
	exponential	3.26686E-04	3.14476	3.14476
	linear	-0.388786	40.4944	40.4944
	gaussian	-0.399827	76.5469	76.5469
	spherical	-0.116354	2.99049	2.99049
	gaussian	-0.143787	2.88600	2.8860
chloride	nugget	0.131832		
	linear	0.752781	40.4944	40.4944
	gaussian	1.42476	70.1714	70.1714
	gaussian	8.88868E-02	8.69715	8.69715
nitrates/ch	nugget	0.501492		
	exponential	0.850073	22.2423	22.2423
	gaussian	5.94618	97.5666	97.5666
	spherical	2.08493	119.788	119.788
	linear	-0.460608	40.4944	40.4944
	gaussian	-0.158956	44.7506	44.7506
	spherical	-3.12102E-02	63.8406	63.8406
	spherical	-2.08348E-02	3.96938	3.96938
	gaussian	-0.158869	3.48219	3.48219

TABLE 3. Variograms and cross-variograms for 1985 cokrigings, I. Sills for cross-variograms should be multiplied by .25.

variables	model type	sill	major range	minor range
magnesium	nugget	0.248134		
	gaussian	0.991270	45.7000	45.700
	spherical	0.320873	37.6213	37.6213
	gaussian	8.27806E-02	8.50180	8.5018
	spherical	3.51657E-02	2.99078	2.99078
	exponential	7.55144E-04	8.30604	8.30604
nitrates/mg	nugget	0.483458		
	linear	0.966404	40.4944	40.4944
	spherical	1.69624	37.2791	37.2791
	spherical	0.319350	2.64113	2.64113
	gaussian	0.136415	2.73304	2.73304
	gaussian	0.685259	44.8484	44.8484
	linear	-0.317701	40.4944	40.4944
	gaussian	-0.282748	70.9316	70.9316
	spherical	-6.19093E-02	2.96409	2.96409
	gaussian	-0.131639	2.90679	2.90679

TABLE 4. Variograms and cross-variograms for 1985 cokrigings, II. Sills for cross-variograms should be multiplied by .25.

variables	model type	sill	major range	minor range
pca reg.	nugget	0.107054		
	gaussian	1.94307	61.5769	61.5769
	spherical	0.236699	60.6609	60.6609
	gaussian	9.20100E-02	9.32408	9.32408
	spherical	5.79837E-02	2.14696	2.14696
	spherical	1.36684E-02	0.718086	0.718086
nitrates/pca	nugget	-0.239728		
	linear	2.40809	40.4944	40.4944
	gaussian	2.13982	63.7761	63.7761
	exponential	0.322951	0.763143	0.763143
	exponential	0.164408	8.21262	8.21262
	gaussian	0.308639	1.87242	1.87242
	exponential	-0.854502	106.308	106.308
	gaussian	-0.254184	141.005	141.005
	spherical	-0.123527	47.9941	47.9941
lin. reg.	nugget	0.134227		
	gaussian	0.812417	52.6957	52.6957
	spherical	0.414495	42.2701	42.2701
	exponential	2.97048E-02	7.18447E-02	7.18447E-02
	gaussian	0.150793	9.72960	9.7296
	spherical	8.24180E-02	2.75982	2.75982
nitrates/lr	nugget	0.54333		
	linear	2.81588	40.4944	40.4944
	spherical	0.419211	10.8113	10.8113
	gaussian	0.339471	26.8142	26.8142
	spherical	0.342556	2.52518	2.52518
	gaussian	1.21394E-02	9.54611	9.54611
	linear	-0.205934	40.4944	40.4944
	gaussian	-0.229260	64.9016	64.9016
	spherical	-0.112005	2.89269	2.89269
	gaussian	-0.124326	2.85581	2.85581
	exponential	-4.17343E-04	9.32696	9.32696

TABLE 5. Variograms and cross-variograms for 1985 cokrigings, III. Sills for cross-variograms should be multiplied by .25.

variables	model type	sill	major range	minor range
nitrate	nugget	0.293012		
	gaussian	0.366413	33.3804	33.3804
	spherical	0.273588	31.1430	31.143
	gaussian	0.337290	3.86012	3.86012
calcium	nugget	0.183254		
	gaussian	1.39759	42.1884	42.1884
	spherical	0.277054	57.8068	57.8068
	gaussian	1.28824E-02	1.89608	1.89608
nitrate/ca	nugget	0.236663		
	gaussian	3.81078	38.9175	38.9175
	spherical	0.727605	3.83310	3.8331
	linear	-0.597378	37.4790	37.479
	spherical	-0.257844	11.3180	11.318
	spherical	-0.179605	4.85909	4.85909
chloride	nugget	0.129331		
	gaussian	6.00816	96.3323	96.3323
nitrate/ch	nugget	-0.183394		
	gaussian	5.84938	53.9010	53.901
	spherical	0.748162	3.93145	3.93145
	gaussian	0.337721	0.466806	0.466806
	exponential	5.73005E-05	0.358862	0.358862
	linear	-0.799765	37.4790	37.479
	spherical	-0.214501	11.7570	11.757
	spherical	-1.01492E-01	4.80253	4.80253
magnesium	nugget	7.55406E-03		
	gaussian	1.60567	40.5097	40.5097
	spherical	0.152768	4.27598	4.27598
	exponential	1.95908E-02	0.359850	0.35985
	gaussian	5.88805E-02	0.471245	0.471245

TABLE 6. Variograms and cross-variograms for 1988 cokrigings, I. Sills for cross-variograms should be multiplied by .25.

variables	model type	sill	major range	minor range
nitrate/mg	nugget	0.230483		
	linear	0.575178	37.4790	37.479
	spherical	0.185621	4.73154	4.73154
	exponential	2.19043E-02	10.9273	10.9273
	exponential	1.61267E-02	9.66567	9.66567
	gaussian	6.23166E-02	9.00632	9.00632
pca reg.	nugget	8.87519E-02		
	linear	0.268435	60.4790	60.479
	gaussian	2.99265	74.1560	74.156
	spherical	0.173899	55.8853	55.8853
nitrate/pca	nugget	8.20860E-02		
	gaussian	5.09646	52.3110	52.311
	spherical	0.670534	4.04593	4.04593
	linear	-1.34358	60.4790	60.479
	spherical	-0.325774	13.9880	13.988
	spherical	-0.119512	4.64139	4.64139
lin reg.	nugget	0.112182		
	gaussian	1.24605	35.4509	35.4509
	spherical	4.11003E-02	31.9355	31.9355
	gaussian	6.14326E-02	4.09701	4.09701
	spherical	0.136326	4.38693	4.38693
	exponential	3.14396E-02	0.282199	0.282199
nitrate/lr	nugget	-0.222216		
	gaussian	3.63794	34.6973	34.6973
	spherical	0.936691	4.34813	4.34813
	gaussian	0.428659	0.476063	0.476063
	gaussian	0.111428	4.01913	4.01913
	spherical	0.109957	0.613066	0.613066
	linear	-0.130961	37.4790	37.479
	spherical	-0.140859	14.1453	14.1453
	exponential	-0.169567	6.06829	6.06829
	exponential	-6.38334E-05	9.01966	9.01966

TABLE 7. Variograms and cross-variograms for 1988 cokrigings, II. Sills for cross-variograms should be multiplied by .25.

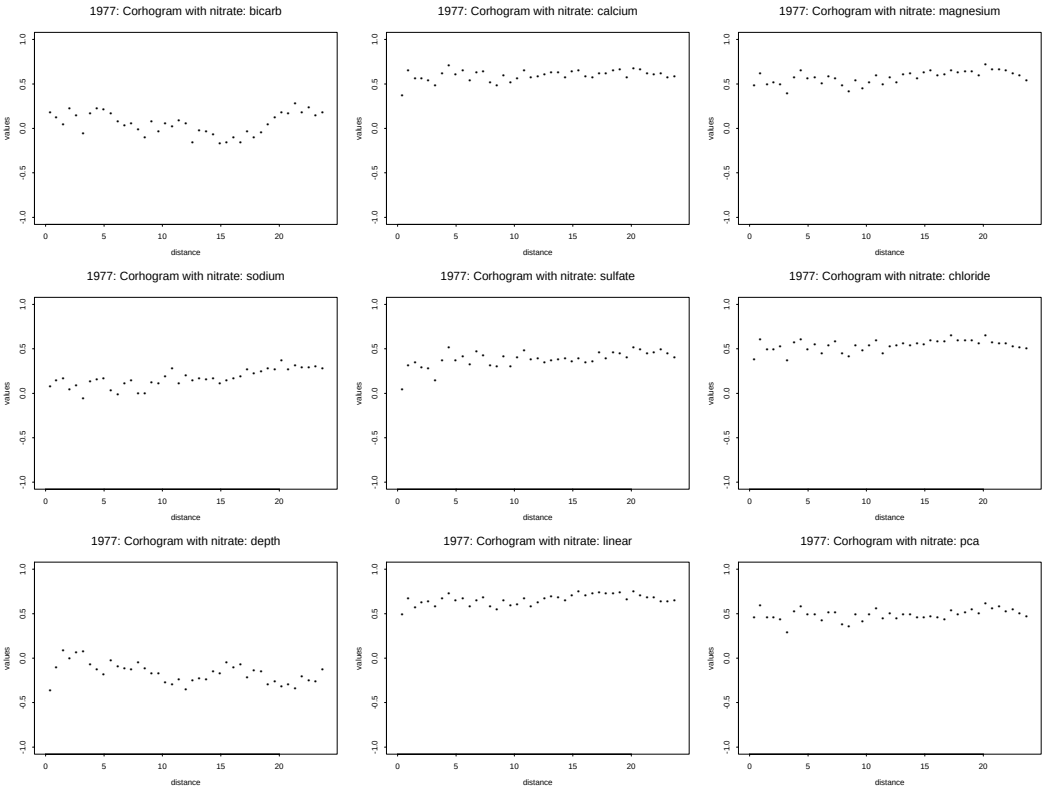


FIGURE 1. All Corhograms for the 1977 data set

.2 Corhograms

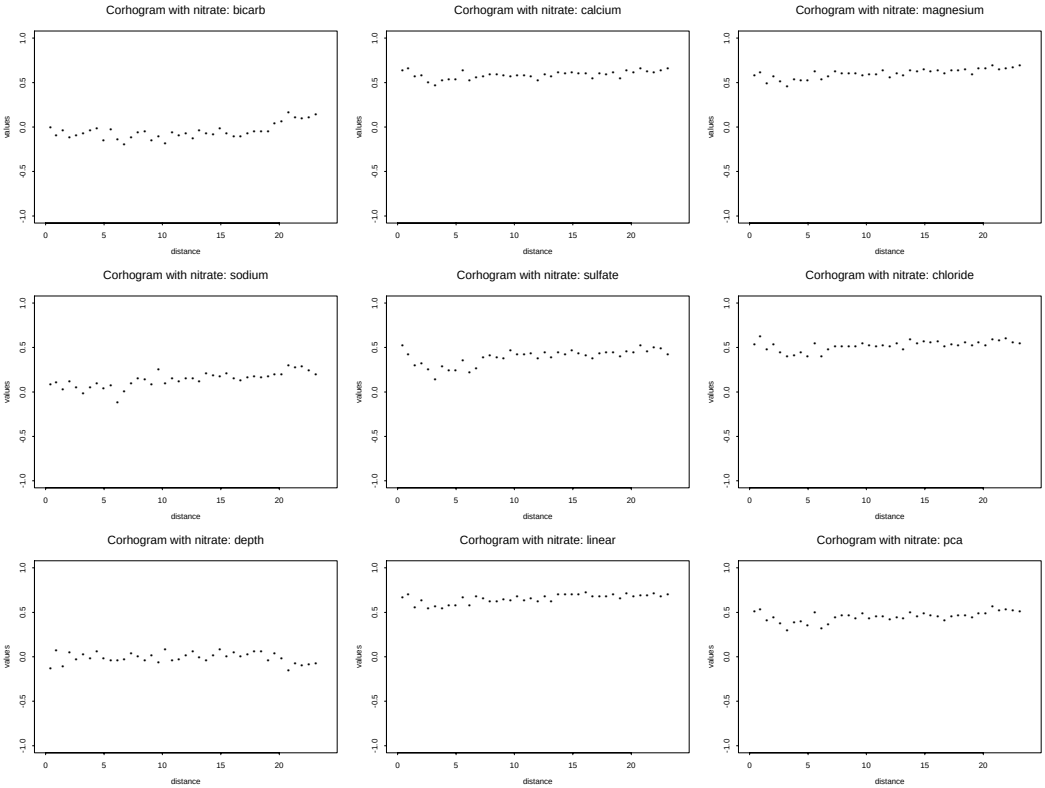


FIGURE 2. All Corhograms for the 1985 data set

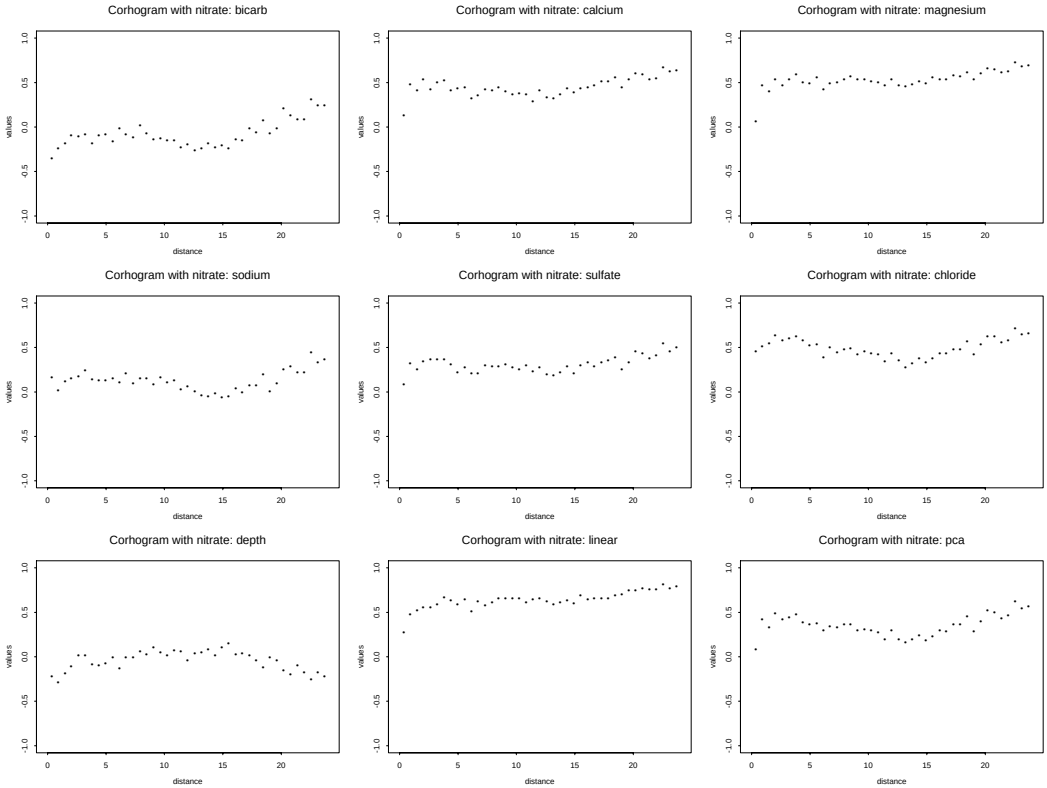


FIGURE 3. All Corhograms for the 1988 data set

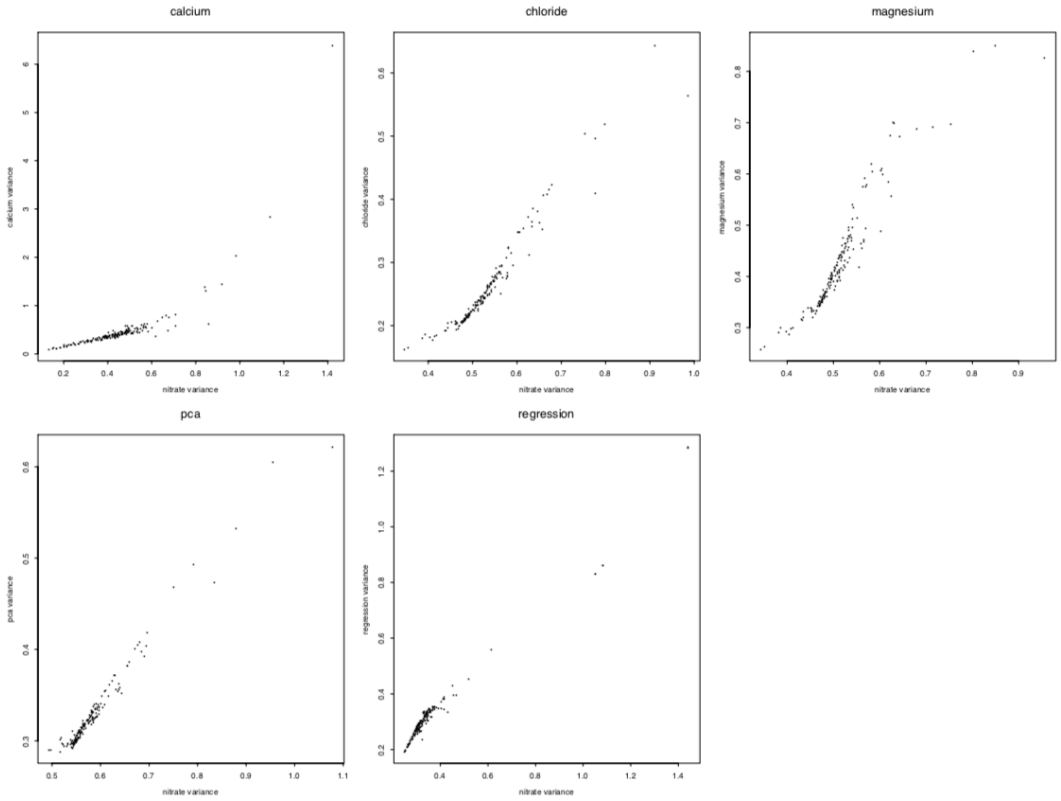


FIGURE 4. Cross-Variance Map, Circa 1977

.3 Variance Maps

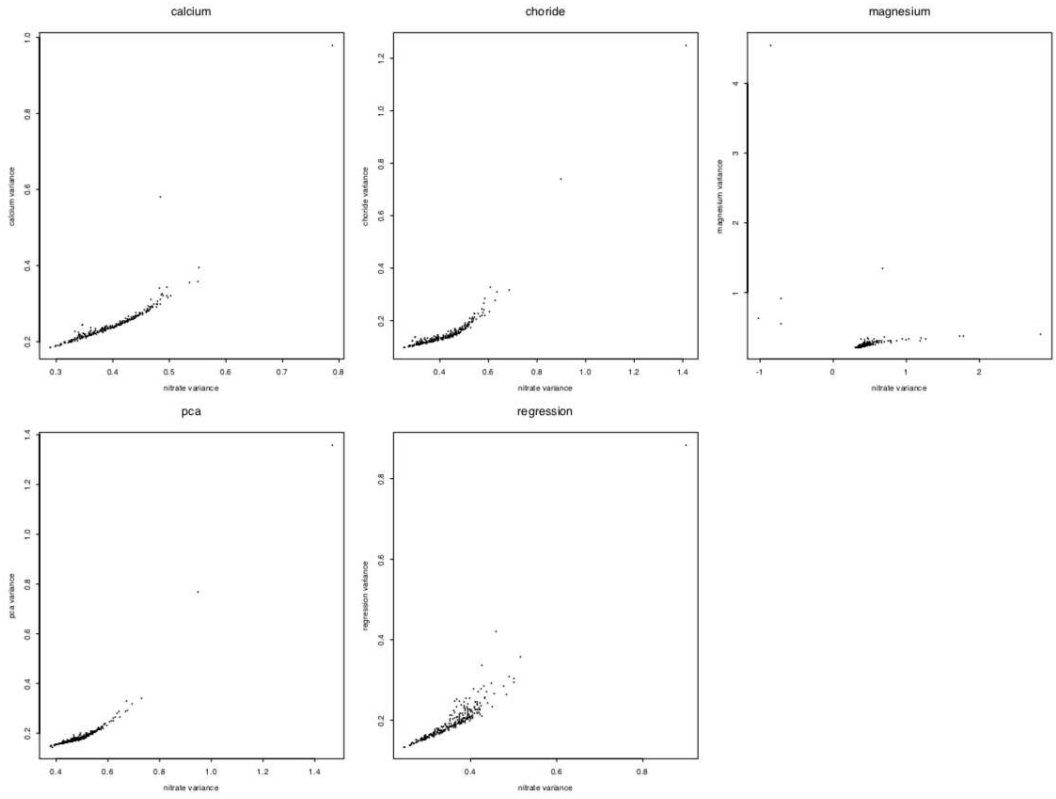


FIGURE 5. Cross-Variance Map, Circa 1985

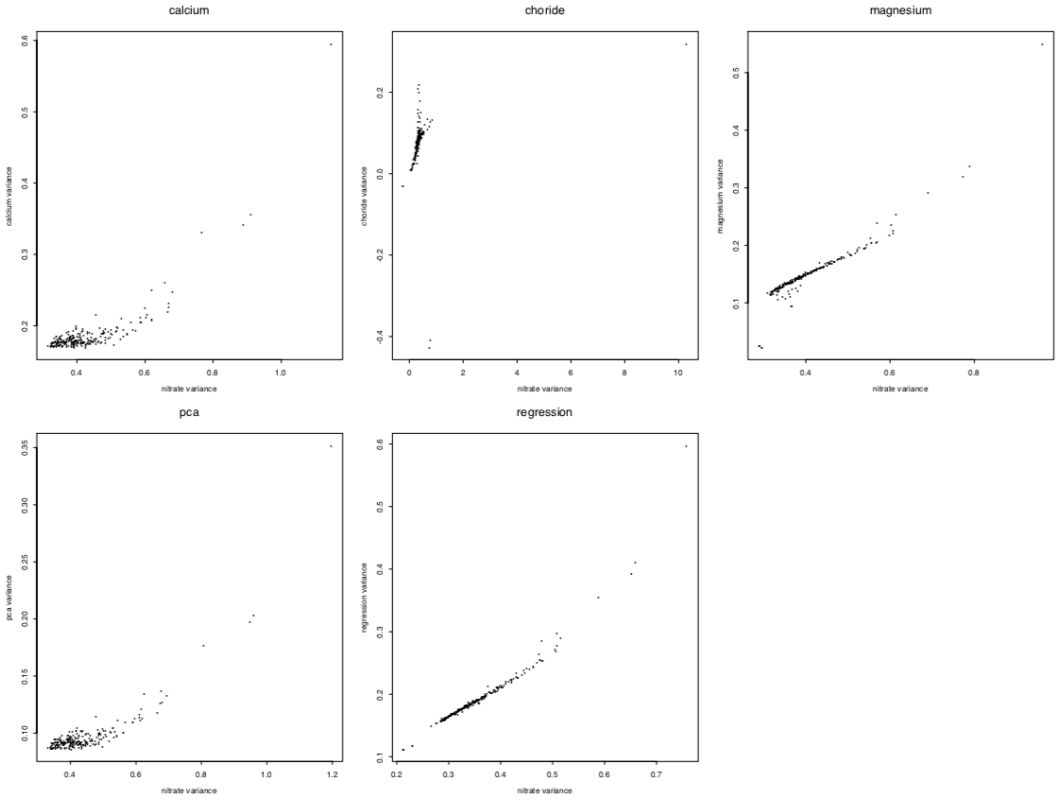


FIGURE 6. Cross-Variance Map, Circa 1988

References

[1] A. Agresti. *Categorical Data Analysis*. Applied Probability and Statistics. John Wiley and Sons, 1990.

[2] S. Ahmed. *Éstimation des transmissivités des aquifères par méthodes géostatistiques multivariables et résolution indirecte du problème inverse*. PhD thesis, l'École Nationale Supérieure des Mines de Paris, October 1987.

[3] A. Albert. *Regression and the Moore-Penrose Inverse*, volume 94 of *Mathematics in Science and Engineering*. Academic Press, New York, 1972.

[4] H. C. Andrews and C. L. Patterson. Outer product expansions and their uses in digital image processing. *American Mathematical Monthly*, 82(1):1–13, January 1975.

[5] M. Armstong and G. Matheron. Disjunctive kriging revisited: Part I. *Mathematical Geology*, 18(8):711–728, 1986.

[6] F. Avila and D. E. Myers. Correspondence analysis applied to environmental data sets: a study of Chatauqua lake sediments. *Chemometrics and Intelligent Laboratory Systems*, 11:229–249, 1991.

[7] C. G. Barancourt. *Étude de l'intermittence et de la variabilité des champs de précipitation par une approache stochastique*. PhD thesis, Université Joseph Fourier - Grenoble I, 1990.

[8] R. J. Barnes. The variogram sill and the sample variance. *Mathematical Geology*, 23(4):673–678, July 1991.

[9] R. Bellman. *Introduction to Matrix Analysis*. McGraw-Hill, 1960.

[10] G. Bourgault and D. Marcotte. Multivariable variogram and its application to the linear model of coregionalization. *Mathematical Geology*, 23(7):899–928, October 1991.

[11] J. R. Carr and D. E. Myers. Application of the theory of regionalized variables to the spatial analysis of LANDSAT data. In *Proceedings of the Ninth Pecora Symposium on Remote Sensing, 2-4 October*. IEEE Computer Society Press, 1984.

[12] J. R. Carr and D. E. Myers. Efficiency of different equation solvers in cokriging. *Computers and Geosciences*, 16(5):705–716, 1990.

[13] J. R. Carr, D. E. Myers, and C. E. Glass. Cokriging - a computer program. *Computers and Geosciences*, 11(2):111–127, 1985.

[14] C. K. Chu and J. S. Marron. Comparison of kernel regression estimators. Technical report, University of North Carolina, 1988.

[15] D. R. Cox and H. D. Miller. *The Theory of Stochastic Processes*. Methuen, London, 1965.

[16] N. Cressie. Spatial prediction and ordinary kriging. *Mathematical Geology*, 20(4):405–421, May 1988.

[17] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252, 1990.

[18] N. A. C. Cressie. Fitting variogram models by weighted least squares. *Mathematical Geology*, 17(5):563–586, July 1985.

[19] N. A. C. Cressie and D. M. Hawkins. Robust estimation of the variogram: I. *Mathematical Geology*, 12:115–125, 1980.

[20] M. W. Davis and C. Grivet. Kriging in a global neighbourhood. *Mathematical Geology*, 16(3):249–266, 1984.

[21] P. Delfiner. Linear estimation of non-stationary spatial phenomena. In M. Guarascio, M. David, and C. Huijbregts, editors, *Advanced Geostatistics in the Mining Industry*, pages 49–68, Dordrecht, Holland, October 1975. NATO Advanced Study Institute, D. Reidel.

[22] F. Dewilde and E. F. Deprettere. Singular value decomposition: an introduction. In E. F. Deprettere, editor, *SVD and Signal Processing : Algorithms, Applications, and Architecture*, Amsterdam, 1988. Institute of Electrical and Electronics Engineers. Region 8, Elsevier Science Pub. Co.

[23] O. Dubrule. Cross validation of kriging in a unique neighborhood. *Mathematical Geology*, 15(6):687–699, 1983.

[24] E. J. Englund and A. R. Sparks. Geo-EAS (Geostatistical Environmental Assessment Software). (Developed by the Environmental Monitoring Systems Laboratory - Las Vegas for DOS; approved by the Environmental Protection Agency (EPA/600/4-88/033); extended to UNIX by the Geostatistics Group, University of Arizona, 1993, with the addition of several programs.), 1988.

[25] W. Feller. *An Introduction to Probability Theory and Its Applications: Volume II*. John Wiley and Sons, Inc., 1966.

[26] B. Flury. *Common Principal Components and Related Multivariate Models*. Wiley, New York, 1988.

[27] G. E. Forsythe and M. A. Malcolm. *Computer Methods for Mathematical Computations*. Prentice-Hall, 1977.

[28] A. Galli, F. Gerdil-Neuillet, and C. Dadou. Factorial kriging analysis: a substitute to spectral analysis of magnetic data. In G. Verly, M. David, A. G. Journel, and A. Marechal, editors, *Geostatistics for Natural Resources Characterization: Part I*, pages 543–558, Dordrecht, Holland, 1984. NATO Advanced Study Institute on Geostatistics for Natural Resources Characterization, D. Riedel Publishing Company.

[29] A. Galli, E. Murillo, and J. Thomann. Dual kriging - its properties and its uses in direct contouring. In G. Verly, M. David, A. G. Journel, and A. Marechal, editors, *Geostatistics for Natural Resources Characterization: Part II*, pages 621–634, Dordrecht, Holland, 1984. NATO Advanced Sudy Institute on Geostatistics for Natural Resources Characterization, D. Riedel Publishing Company.

[30] P. Geladi, H. Isaksson, L. Lindqvist, S. Wold, and K. Esbensen. Principal component analysis and multivariate images. *Chemometrics and Intelligent Laboratory Systems*, 5:209–220, 1989.

[31] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *J. SIAM Numer. Math.*, 2(2):205–223, 1965.

[32] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solution. *Numer. Math.*, 14:403–420, 1970.

[33] P. Goovaerts. *Multivariate Geostatistical Tools for Studying Scale-Dependent Correlation Structures and Describing Space-Time Variations*. PhD thesis, Université Catholique de Louvain, 1992.

[34] P. Goovaerts. On a controversial method for modeling a coregionalization. *Mathematical Geology*, 26(2):197–204, February 1994.

[35] P. Goovaerts. Study of spatial relationships between two sets of variables using multivariate geostatistics. *Geoderma*, 62:93–107, 1994.

[36] J. M. Hamlett, R. Horton, and N. A. C. Cressie. Resistant and exploratory techniques for use in semivariogram analysis. *Soil Sci. Soc. Am. J.*, 50:868–875, 1986.

[37] D. R. Helsel. Less than obvious: statistical treatment of data below the detection limit. *Environ. Sci. Technol.*, 24(12):1767–1774, 1990.

[38] J. D. Helterbrand and N. Cressie. Universal cokriging under intrinsic coregionalization. *Mathematical Geology*, 26(2):197–204, February 1994.

[39] T. Hern and C. A. Long. Viewing some concepts and applications in linear algebra. *MAA Notes Series*, 19:173–190, May 1991.

[40] K. Hoechsmann. Singular values and the spectral theorem. *The American Mathematical Monthly*, 97(5):413–414, May 1990.

[41] C. Huijbregts and G. Matheron. Universal kriging (an optimal method for estimating and contouring in trend surface analysis. In *Decision-Making in the Mineral Industry*, pages 159–169, Montreal, 1970. Canadian Institute of Mining and Metallurgy, Canadian Institute of Mining and Metallurgy.

[42] A. G. Journel. Kriging in terms of projections. *Mathematical Geology*, 9(6):563–586, 1977.

[43] A. G. Journel. Nonparametric estimation of spatial distributions. *Mathematical Geology*, 15(3):445–468, 1983.

[44] A. G. Journel. Answers to Margaret Armstrong and Robert F. Shurtz's comments on 'The deterministic side of geostatistics'. *Mathematical Geology*, 17(8):869, November 1985.

[45] A. G. Journel. The deterministic side of geostatistics. *Mathematical Geology*, 17(1):1–15, January 1985.

[46] A. G. Journel. Geostatistics: Models and tools for the earth sciences. *Mathematical Geology*, 18(1):119–140, January 1986.

[47] A. G. Journel. New distance measures: the route toward truly non-gaussian geostatistics. *Mathematical Geology*, 20(4):459–475, May 1988.

[48] A. G. Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978.

[49] V. E. Kane, C. L. Begovich, T. R. Butz, and D. E. Myers. Interpretation of regional geochemistry using optimal interpolation parameters. *Computers and Geosciences*, 9(2):117–136, 1982.

[50] J. L. Kelley. *General Topology*. D. Van Nostrand Company, Princeton, New Jersey, 1955.

[51] L. D. Lathauwer. Personal communication. E.E. Department, Kath. Universiteit Leuven, Leuven, Belgium. January 25, 1994.

[52] A. E. Long, D. Gellenbeck, and J. G. Brown. Geostatistical Analysis of Nitrates in the Western Part of the Salt River Valley Area, Maricopa County, Arizona. Technical report, United States Geological Survey, 1994. In review.

[53] C. A. Long. Visualization of matrix singular value decomposition. *Mathematics Magazine*, 56(3):161–167, May 1983.

[54] D. Lovelock and H. Rund. *Tensors, Differential Forms, and Variational Principals*. Dover, 1989.

[55] D. Marcotte. Cokriging with Matlab. *Computers and Geosciences*, 17(9):1265–1280, 1991.

[56] D. M. Mark and M. Church. On the misuse of regression equations in earth sciences. *Mathematical Geology*, 9(1):63–75, 1977.

[57] G. Matheron. *Les Variables Régionalisées et Leur Estimation*. Masson et Cie, Paris, 1965.

[58] G. Matheron. *The Theory of Regionalized Variables and its Applications*. Centre de Geostatistique, Fontainebleau, France, 1971.

[59] G. Matheron. Les concepts de base et l'évolution de la géostatistique miniere. In M. Guarascio, M. David, and C. Huijbregts, editors, *Advanced Geostatistics in the Mining Industry*, pages 3–10, Dordrecht, Holland, October 1975. NATO Advanced Study Institute, D. Reidel.

[60] G. Matheron. A simple substitute for conditional expectation: the disjunctive kriging. In M. Guarascio, M. David, and C. Huijbregts, editors, *Advanced Geostatistics in the Mining Industry*, pages 221–236, Dordrecht, Holland, October 1975. NATO Advanced Study Institute, D. Reidel.

[61] G. Matheron. *Estimer et Choisir: Fasc. 7*. Centre de Geostatistique, Fontainebleau, France, 1978.

[62] G. Matheron. Recherche de simplification dans un probleme de cokrigeage. *Publication N-628, Centre des Géostatistique, Ecole des Mines*, 1979.

[63] G. Matheron. *Estimating and Choosing*. Springer-Verlag, Berlin, 1989. Translated by A. M. Hasofer.

[64] D. W. McCarn and J. R. Carr. Influence of numerical precision and equation solution algorithm on computation of kriging weights. *Computers and Geosciences*, 18(9):1127–1167, 1992.

[65] K. Messer. A comparison of a spline estimate to its equivalent kernel estimate. *The Annals of Statistics*, 19(2):817–829, 1991.

[66] D. E. Myers. Matrix formulation of co-kriging. *Mathematical Geology*, 14(3):249–257, 1982.

[67] D. E. Myers. Interpolation with positive definite functions. *Sciences de la Terre*, 28:251–265, 1988.

[68] D. E. Myers. To be or not to be...stationary: That is the question. *Mathematical Geology*, 21(3):347–362, April 1989.

[69] D. E. Myers. On variogram estimation. In *The Frontiers of Statistical Scientific Theory and Industrial Applications*, pages 261–281. American Sciences Press, Inc., 1991.

[70] D. E. Myers. Kriging, cokriging, radial basis functions and the role of positive definiteness. *Computer Math. Applic.*, 24(12):139–148, 1992.

[71] R. J. O'Dowd. Conditioning of coefficient matrices of ordinary kriging. *Mathematical Geology*, 23(5):721–740, July 1991.

[72] G. Pan, D. Gaard, K. Moss, and T. Heiner. A comparison between cokriging and ordinary kriging: case study with a polymetalic deposit. *Mathematical Geology*, 25(3):377–398, April 1993.

[73] G. M. Philip and D. F. Watson. Matheronian geostatistics: Quo vadis. *Mathematical Geology*, 18(1):93–117, January 1986.

[74] D. Posa. Conditioning of the stationary kriging matrices for some well-known covariance models. *Mathematical Geology*, 21(7):755–766, 1989.

[75] D. Posa. Limiting stochastic operations for stationary spatial processes. *Mathematical Geology*, 23(5):695–702, April 1991.

[76] R. W. Preisendorfer and C. D. Mobley. *Principal Component Analysis in Meteorology and Oceonography*, volume 17 of *Developments in Atmospheric Science*. Elsevier Science Publishers, Amsterdam, 1988.

[77] C. E. Puente and R. L. Bras. Disjunctive kriging, universal kriging, or no kriging: small sample results with simulated fields. *Mathematical Geology*, 18(3):287–305, 1986.

[78] J. J. Royer and P. C. Vieira. Dual formalism of kriging. In G. Verly, M. David, A. G. Journel, and A. Marechal, editors, *Geostatistics for Natural Resources Characterization: Part II*, pages 691–702, Dordrecht, Holland, 1984. NATO Advanced Sudy Institute on Geostatistics for Natural Resources Characterization, D. Riedel Publishing Company.

[79] L. Sandjivy. The factorial kriging analysis of regionalized data: Its application to geochemical prospecting. In G. Verly, M. David, A. G. Journel, and A. Marechal, editors, *Geostatistics for Natural Resources Characterization: Part I*, pages 559–571, Dordrecht, Holland, 1984. NATO Advanced Study Institute on Geostatistics for Natural Resources Characterization, D. Riedel Publishing Company.

[80] R. F. Sanford, C. T. Pierson, and R. A. Crovelli. An objective replacement method for censored geochemical data. *Mathematical Geology*, 25(1):59–80, January 1993.

[81] R. F. Shurtz. A critique of A. Journel's 'The deterministic side of geostatistics'. *Mathematical Geology*, 17(8):861–868, November 1985.

[82] B. W. Silverman. Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*, 12(3):898–916, 1984.

[83] A. R. Solow. Mapping by simple indicator kriging. *Mathematical Geology*, 18(3):335–352, 1986.

[84] I. Stakgold. *Green's Functions and Boundary Value Problems*. Wiley, 1979.

[85] G. W. Stewart. On the early history of the singular value decomposition. *Siam Review*, 35(4):551–566, December 1993.

[86] J. Stewart. Positive definite functions and generalizations, an historical survey. *Rocky Mountain Journal of Mathematics*, 6(3):409–434, 1976.

[87] G. Strang. *Linear Algebra and Its Application*. Academic Press, New York, second edition, 1980.

[88] M. R. Stytz and R. W. Parrott. Using kriging for 3D medical imaging. *Computerized Medical Imaging and Graphics*, 17(6):421–442, November-December 1993.

[89] F. van der Meer. Quantification of grain shape and texture using mathematical morphology and geostatistical techniques. In *Proceedings of Prague Conference*, 1993.

[90] H. Wackernagel. The inference of the linear model of the coregionalization in the case of a geochemical data set. *Sciences de la Terre*, 24:81–93, 1985.

[91] H. Wackernagel. Geostatistical techniques for interpreting multivariate spatial information. In C. F. Chung et al., editor, *Quantitative analysis of mineral and Energy Resources*, pages 393–409, Dordrecht, 1988. Reidel.

[92] H. Wackernagel. Description of a computer program for analyzing multivariate spatially distributed data. *Computers and Geosciences*, 15(4):593–598, 1989.

[93] H. Wackernagel. Cokriging versus kriging in regionalized multivariate data analysis. *Geoderma*, 62:83–92, 1994.

[94] H. Wackernagel, P. Petitgas, and Y. Touffait. Overview of methods for coregionalization analysis. In M. Armstrong, editor, *Geostatistics, Vol. 1*, pages 409–420. Kluwer Academic Publishers, 1989.

[95] A. W. Warrick, R. Zhang, M. K. El-Haris, and D. E. Myers. Direct comparison between kriging and other interpolators. In *Validation of Flow and Transport Models in the Unsaturated Zone*, pages 505–510, Ruidoso, New Mexico, 1988. May 23-26, 1988.

[96] G. S. Watson. Smoothing and interpolation by kriging and with splines. *Mathematical Geology*, 16(6):601–615, 1984.

[97] G. P. Williams. Improper use of regression equations in earth sciences. *Geology*, 11:195–197, April 1983.

[98] T. Xie. *Positive Definite Matrix-Valued Functions and Variogram Modelling.* PhD thesis, University of Arizona, 1994.

[99] S. J. Yakowitz and F. Szidarovszky. A comparison of kriging with nonparametric regression methods. *Journal of Multivariate Analysis*, 16:21–53, 1985.

[100] E. A. Yfantis, M. Au, and F. S. Makri. Image compression and kriging. In R. Dimitrakopoulos, editor, *Geostatistics for the Next Century*, pages 156–161. Kluwer Academic Publishers, 1994.