A Note on Programs Performing Kriging with Nonnegative Weights

Ute Christina Herzfeld

This note deals with the problem of solving kriging systems with nonnegative weights. Mathematically the question is considered as a case of quadratic programming. Hints concerning computational approaches and software are given.

KEY WORDS: Kriging under constraints, quadratic programming, software.

As is well known, kriging (point kriging) gives an optimal linear estimator of a phenomenon described by a regionalized variable in the sense of Matheron (cf. Journel and Huijbregts, 1978, p. 568ff): The data are considered as realization of a random process (under a stationary or intrinsic hypothesis). The value $y_0$ at a point $x_0$ is estimated by

$$y_0^* = \sum_{i=1}^{n} \lambda_i y_i, \quad \lambda_i \in \mathbb{R}$$

where $y_i$ are measured values at points $x_i$ in a suitable neighborhood of $x_0$ ($i = 1, \ldots, n$). Weights $\lambda_i$ are determined such that the estimation variance

$$\text{var}(y_0^* - y_0)$$

is minimal and the estimation is unbiased.

Note that, in the estimation, negative coefficients may occur. In some applications, however, the geologic setting of the problem requires an estimator involving only nonnegative weights.

Given nonnegative data $y_i$, nonnegativity of weights $\lambda_i$ is a sufficient, but not a necessary, condition to gain nonnegative estimates [examples are found easily, in opposition to Limić and Mikelić (1984, p. 425)].
For instance, in resource exploration, geologic layers should always have nonnegative thicknesses and hanging and foot walls must not cross each other.

If nonnegative weights are required, kriging has to be performed subject to the additional condition

$$\lambda_i \geq 0, \quad i = 1, \ldots, n$$

(3)

This amounts to a problem that may be considered in the wider context of convex programming.

The function \( f \) in point kriging is denoted,

$$f(\lambda_1, \ldots, \lambda_n) = \sum_{i,j=1}^{n} \lambda_i \lambda_j C(x_i - x_j) - 2 \sum_{i=1}^{n} \lambda_i C(x_i - x_0) + C(0)$$

(4)

where \( C \) is the covariance function.

Function \( f \) has to be minimized with respect to the unbiasedness condition

$$\sum_{i=1}^{n} \lambda_i = 0$$

(5)

and nonnegativity conditions (3). (The use of a positive definite covariance model \( C \) or variogram \( \gamma \) and of pairwise different data points ensures that the matrix constructed by \( [C(x_i - x_j)]_{i,j=1}^{n} \) is strictly positive definite (Journel and Huijbregts, 1978, p. 307-308), and, consequently, \( f \) is in fact a convex function.) Herzfeld (1988) provides the theoretical background.

In this form, software is available that does the major part of the optimization in nonnegative kriging; these procedures involve either modified simplex-type methods derived from the Kuhn–Tucker optimality conditions to solve the above problem of convex programming Eqs. (3)–(5) (e.g., Limić and Mikelić 1984) or projection methods on active sets of constraints (Goldfarb and Idnani, 1983). The ‘active constraints’ constitute the subset of constraints \( \{ \lambda_i \geq 0, \ i = 1, \ldots, n \} \) characterized by the following property: If the active constraints are satisfied as exact data (i.e., \( \lambda_i = 0 \) for \( i \) in the active set), all other constraints are satisfied as well. As soon as the set of active constraints is known, the solution of kriging under constraints is equivalent to (usual) kriging relying on the original (exact) data plus the active constraints taken as exact data.

To avoid the computational effort of quadratic optimization, Barnes and Johnson (1984) use the latter property of the active set concept and develop a fast algorithm that first solves the ordinary point kriging system and then stepwise eliminates components with negative weights. Though it serves the most likely practical cases, however, it might yield a solution that is not consistent with the theoretical optimal one (Barnes and Johnson, 1984, p. 240).

Szidyrovsky et al. (1987) base their algorithm on the property of the active constraints mentioned above. They generate the power set (set of all subsets)