Optimal Correction of Indefinite Correlation Matrices

Abhay Kumar and Clayton V. Deutsch

Correlation matrices must be positive semi definite, but sometimes unavailability and/or erroneous data produces indefinite correlation matrices. Some methods are presented to correct an indefinite correlation matrix to the nearest positive definite one so that geostatistical calculations can be performed. Uncertainty in the correlation coefficients can be handled by associating weights to them. These weights determine the relative change in the individual elements of the initial correlation matrix.

Introduction

Correlation matrices are built to describe the dependency between different data sets. In reservoir estimation the primary well data, which is expensive to obtain by means of drilling is predicted using the easy and cheap obtaining secondary seismic data. Generally linear model of coregionalization (LMC) is used to estimate the primary variable by means of secondary variable (s). Even selecting the positive semi-definite variogrma models for primary and secondary variables doesn't guarantee that the Correlation matrix of whole system is positive semi-definite (for details see [5]). Generally seismic data are available in abundance but the primary well data are not. A correlation matrix formed with some missing primary data can result in an indefinite matrix. Even error in a single element of 100 x 100 correlation matrix can give an indefinite matrix.

In order to overcome this problem one can find the error full elements of the correlation matrix and change only these elements to fix the problem. But this is not a good idea because the resulting matrix can be far from the original matrix in terms of matrix norm. To get the optimal positive definite correlation matrix all non-diagonal elements of matrix must be allowed to change. At the same time the relative change in the individual elements of the matrix must be controlled since we may be confident about some elements, that they are correct one and should be allowed to change minimum. This objective can be achieved by introducing an appropriate weighting matrix to reflect the uncertainty in each element of the original correlation matrix. In this paper two main algorithms which are adapted from Higham [1] and Pietersz & Gorenen [1],are presented herein with some modifications necessary to handle problem of indefinite correlation matrices in geostatistics. The latter algorithm is able to handle the general weights applied on the individual elements of the matrix. The objective is to minimize the "**weighted** *Frobenius norm"* i.e.

$$\| X_W - R \|_W = \sqrt{\sum\sum w_{ij} * (x_{ij} - r_{ij})^2} \tag{1}$$

The unweighted Frobenius norm is the special case of (1) when weight matrix $w_{ij} = 1$ i.e. all elements have same weight. The problem of finding optimum nearest positive definite matrix can be stated as follows:

$Find \quad X \in \mathbb{R}^{n \times n}$

to

$$Minimize: \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} * (x_{ij} - r_{ij})^2 \tag{2}$$

$Subjected\ to: X \in P$

$where, R \in C\ and$

$P = \left\{ A : A = A^T \in \mathbb{R}^{n \times n}, A > 0,\ a_{ij} = \begin{cases} 1, & if\ i = j \\ \leq 1, & if\ i \neq j \end{cases} \right\}$ is a set of symmetric positive definite

matrices with unit diagonal elements and

$$C = \left\{ \; A : A = A^T \in \mathbb{R}^{n\times n},\; a_{ij} = \begin{cases} 1, & if\ i = j \\ \leq 1, & if\ i \neq j \end{cases} \right\}$$

The objective function is just double of square of weighted Frobenious norm (1), thus minimizing (1) is same as minimizing (2).

Literature Review:

There are various papers on finding the solution of (2). Some of them can handle general weights and some cannot. The projection method has been presented in Higham, 2002. The fact that both sets C and P are closed convex sets, gives the idea that there is a unique optimal solution for this problem [1]. The nearest optimum matrix to R will be a positive semi definite matrix with number of eigen values equal to 0 is same as number of negative eigen values of R. The uniqueness of optimal solution has been explained in this paper. The main drawback of algorithm explained in Higham's paper is that it is unable to handle general weights. The second method based on Majorization approach has been developed by Pietersz and Groenen [2]. This algorithm has ability to handle general weights. Numerical tests on various indefinite correlation matrices with equal weight matrix conforms that both algorithms give the same optimal solution.

Methodology:

For weighted matrix norm

Using the fact that if X is a symmetric positive definite matrix, there will be one and only one matrix $Y \in \mathbb{R}^{n\times n}$ such that $X = Y\,Y^T$. Representing each row of Y as a column vector $y_i, i = 1, 2....n$:

$$Y = \begin{pmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nn} \end{pmatrix}, \quad \mathbf{y_i} = [y_{i1}, ...y_{in}]^T \qquad \text{Therefore, } Y = [\mathbf{y_1^T}, \mathbf{y_2^T}, ..., \mathbf{y_n^T}]^T \qquad (3)$$

Then each element of matrix X can be expressed as:

$$x_{ij} = <\mathbf{y_i}, \mathbf{y_j}> = \mathbf{y_i^T} * \mathbf{y_j}, \qquad i, j = 1, 2, ...n$$

Then problem (2) can be expressed as:

$$Find \quad X \in \mathbb{R}^{n\times n}$$

$$to\ minimize: \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} * (r_{ij} - <\mathbf{y_i}, \mathbf{y_j}>)^2 \qquad (4)$$

$$Subjected\ to: \| \mathbf{y_i} \|_F = 1, \qquad i = 1, 2, ...n$$

$$X = [\mathbf{y_1^T}, \mathbf{y_2^T}, ..., \mathbf{y_n^T}]^T * [\mathbf{y_1^T}, \mathbf{y_2^T}, ..., \mathbf{y_n^T}]$$

This is a type of constrained optimization problem. The method of **Lagrange Multipliers** will generate the n^2 simultaneous equations in degree 3, which is very difficult to solve. This method has been explained by Zhang & Wu (2003) and Wu (2003) and theoretically guarantees the global convergance, but associated algorithm does not guarantee the same and is not good in practice.

The majorization approach has been used by Pietersz & Groenen (2004) to solve (4). Majorization is an iterative algorithm and when applied guarantees the convergence. The approach is simple: suppose we want to minimize a *complex* function $f(\mathbf{x})$. The first step is to find *a simple function* $g(\mathbf{x})$ which is equal to f at $\mathbf{x} = \mathbf{x}^{(k)}$ (k^{th} iterative stage) and greater than or equal to elsewhere i.e. $g(\mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)})$ and $f(\mathbf{x}) \leq g(\mathbf{x})$. Then find the minimizer $\mathbf{x}^*$ of $g(\mathbf{x})$ and iterate again using $\mathbf{x}^{(k+1)} = \mathbf{x}^*$ by establishing new $g(\mathbf{x})$. This algorithm has a decent property because after each iteration the function value is

guaranteed to be smaller. The globally convergence of this algorithm has been explained by Pietersz & Groenen (2004).

Objective function can be written as:

$$f(X)=\sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}*(r_{ij}-<\mathbf{y_i},\mathbf{y_j}>)^2=\sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}*(r_{ij}^2+(\mathbf{y}_i^T\mathbf{y_j})^2-2r_{ij}\mathbf{y}_i^T\mathbf{y_j}) \quad (5)$$

Now, $f(X)$ can be expressed as a function of $\mathbf{y}=\mathbf{y_i}$ only and treating all other $\mathbf{y_k}$'s as constant. Denote this by $f(X_{\mathbf{y_i}|\mathbf{y_k}})=f(\mathbf{y})$.

$$f(\mathbf{y})=\sum_{j=i+1}^{n} w_{ij}(r_{ij}+(\mathbf{y^T y_j})^2-2r_{ij}\mathbf{y^T y_j})+\text{Constant}$$

$$f(\mathbf{y})=\sum_{j=i+1}^{n} w_{ij}(r_{ij}+(\mathbf{y^T y_j})^2-2r_{ij}\mathbf{y^T y_j})+\text{Constant}$$

$$=\left(\frac{1}{2}\right)\mathbf{y^T}\left\{\sum_{j=i+1}^{n} w_{ij}\,\mathbf{y_j^T y_j}\right\}\mathbf{y}\;-\;\mathbf{y^T}\left\{\sum_{j=i+1}^{n} w_{ij}\,r_{ij}\mathbf{y_j}\right\}+\text{Constant} \quad (6)$$

$$\text{where,}\quad \mathbf{y^T y}=1$$

$f(\mathbf{y})$ is a quadratic function of $\mathbf{y}$. This way the original problem (4) can be subdivided into several constrained quadratic optimization problems in different $\mathbf{y}$'s. Before going into the details of subdividing (4) let us look at the method of optimization of a quadratic function by majorization.

3.1.1 Optimization of quadratic objective function subjected to unit norm constraint by majorization:

In this section an iterative algorithm will be developed to find the optimal of quadratic objective function. To generalize the method, standard notations (independent of other sections) have been used. This algorithm can be used on any type of quadratic objective function with unit norm constraint and it guaranties the convergence. The problem formulation is as follows:

$$\text{Find:}\quad \mathbf{x}\in\mathbb{R}^n$$

$$\text{to minimize:}\quad f(\mathbf{x})=\frac{1}{2}\mathbf{x^T}Q\mathbf{x}-\mathbf{x^T}b \quad (7)$$

$$\text{subjected to:}\quad h(\mathbf{x})=1-\mathbf{x^T x}=0$$

The first idea to solve this problem will be by approach of Lagrange multipliers. But the partial derivatives of Lagrange function will generate a system of simultaneous equation in $\mathbf{x}$ of degree 3. To avoid the complexity of solving the huge number of 3rd degree simultaneous equation we are using the majorization technique. Suppose we are at iteration-k, the current iteration poin is $\mathbf{x}^{(k)}$, we need to find such $g(\mathbf{x})$ so that $g(\mathbf{x}^{(k)})=f(\mathbf{x}^{(k)})\ \ and\ \ f(\mathbf{x})\le g(\mathbf{x})$, $for\ all\ \mathbf{x}\in\mathbb{R}^n$. If λ is the largest eigen value of matrix Q, then $Q-\lambda I_n$ will be negative semi definite, therefore for any vector $(\mathbf{x}-\mathbf{x}^{(k)})$:

$$(\mathbf{x}-\mathbf{x}^{(k)})^T(Q-\lambda I_n)\,(\mathbf{x}-\mathbf{x}^{(k)})\le 0$$

After some manipulations and using the fact that $\mathbf{x^T x}=1$ and $\mathbf{x}^{(k)^T}\mathbf{x}^{(k)}=1$ we get

$$\mathbf{x^T}Q\,\mathbf{x}\le 2\lambda-2\mathbf{x^T}(\lambda\mathbf{x}^{(k)}-Q\mathbf{x}^{(k)})-\mathbf{x}^{(k)^T}Q\mathbf{x}^{(k)}$$

Therefore objective function of (7) can be written as:

$$f(\mathbf{x})\le 2\lambda-2\mathbf{x^T}(b+\lambda\mathbf{x}^{(k)}-Q\mathbf{x}^{(k)})-\mathbf{x}^{(k)^T}Q\mathbf{x}^{(k)}$$

Therefore, $g_k(\mathbf{x})=-2\mathbf{x^T}(b+\lambda\mathbf{x}^{(k)}-Q\mathbf{x}^{(k)})+\text{Constant}$ (8)

Thus we are always able to find a linear function in $\mathbf{x}$ at each iteration. Then we need to find the minimizer of following objective function:

Minimize: $g_k(\mathbf{x})$ (9)

Subjected to: $h(\mathbf{x}) = 1 - \mathbf{x}^{\mathbf{T}}\mathbf{x} = 0$

This is a straight forward constrained problem and the Lagrange function can be written as:

$l_k(\mathbf{x}, d) = g_k(\mathbf{x}) + d(1 - \mathbf{x}^{\mathbf{T}}\mathbf{x}),$ d is the Lagrange multiplier.

Partial derivatives of Lagrange function with respect to $\mathbf{x}$ and d will give the solution:

$$\mathbf{x}^{(k+1)} = \frac{\mathbf{z}^{(k)}}{\| \mathbf{z}^{(k)} \|}, \qquad where \quad \mathbf{z}^{(k)} = b + \lambda \mathbf{x}^{(k)} - Q\mathbf{x}^{(k)}$$

The algorithm is summarized below:

Min_Quad($Q, b, \mathbf{x}^{(0)}$ ***)***

- *Set* $k = 0$*, and* λ *=max of eigen value of* Q
- *Do until* $\frac{(f_{(k-1)} - f_k)}{f_k} < \varepsilon_1$ *and / or* $\| \nabla f \| < \varepsilon_2$
  - *Set* $\mathbf{z}^{(k)} = b + \lambda \mathbf{x}^{(k)} - Q\mathbf{x}^{(k)}$
  - *If (* $\mathbf{z}^{(k)} \neq \mathbf{0}$ *) then*
    - *Set* $\mathbf{x}^{(k+1)} = \frac{\mathbf{z}^{(k)}}{\| \mathbf{z}^{(k)} \|}$
    - *else*
      - *set* $\mathbf{x}^{(k+1)}$ *= any vector of unit length*
  - *set* $k = k + 1$
- *end loop*
- *Return* $\mathbf{x} = \mathbf{x}^{(k-1)}$

end Mim_Quad

Algorithm for minimum weighted matrix norm

Now let us back to the original matrix problem (4). The ***Min_Quad*** algorithm can be used to solve this problem, but first we need to find a good starting matrix Y .For this purpose we can use the eigedecomposition of our correlation matrix $R = Q\Lambda Q^T$, where Q is orthonormal and Λ is a diagonal matrix with eigen values of R . Then the starting matrix can be set as

$$Y^{(0)} = \frac{P}{\| P \|}, \qquad P = \left\{ Q\Lambda^{1/2} \right\} \tag{10}$$

This method is known as "principle component analysis (PCA)" [].

The algorithm to find the nearest positive definite matrix to R is as follows:

Algorithm- 1

Find_Nearest (R,W)

- *Find* $Y^{(0)}$ *using (10), set k=0*
- *Do until* $\frac{\{f(X^{(k-1)}) - f(X^{(k)})\}}{f(X^{(k)})} < \varepsilon_1$*, determine* $f(X)$ *using (5)*
  - *for* $i = 1, 2 \ldots\ldots n$

- $set\, \mathbf{x}^{(0)} = i^{th}\ row\, of\ matrix\, Y^{(k)}, Q = \left\{ \sum_{j=i+1}^{n} w_{ij}\, \mathbf{y_j^T y_j} \right\},$

  $b = \left\{ \sum_{j=i+1}^{n} w_{ij}\, r_{ij} \mathbf{y_j} \right\}$

  - $\mathbf{y} =$ ***Min_Quad(*** $Q, b, \mathbf{x}^{(0)}$ ***)*** *, set* $i^{th}\ row\, of\ matrix\, Y^{(k)} = \mathbf{y}$
- *end for loop*
- *Set* $X^{(k)} = Y^{(k)}\, Y^{(k)^T}$
- *Set k=k+1*

- *end loop*
- *Return* $X^{(k-1)}$

end Find_Nearest

The final matrix X will have number of eigen values equal to 0 is same as number of negative eigen values of R. To make X positive definite all the zero eigen values can be made positive by a small number. This can be achieved by doing the eigenvalue decomposition of X and making all the zero eigen values in the diagonal matrix equal to a small positive number but taking the same orthogonal eigen vectors.

Algorithm in case of no weights

The previous algorithm gives the optimum positive definite matrix whether weight matrix is used or not. Algorithm-1 can be used with a weight matrix having all elements equal to 1 in order to handle non-weighted correlation matrix optimization. But there is another algorithm presented in Higham[] which works much faster than previous one, but this algorithm cannot handle the general weight matrix. In case of large correlation matrices (mostly dimension more than 100 x 100) and unweighted matrix problems, Higham's algorithm can be used to save the running time. Both algorithms give the same result in case of unweighted matrix problem. The idea behind the latter algorithm is finding the successive projections of matrix in set S (set of correlation matrix) onto the set PS (set of positive semi definite matrix) and again from set PS onto set S until the stopping criteria has met. This way the successive projections come closer and closer after each iteration and at the end the resultant matrix lies on the intersection of sets PS and S.
Algorithm-2

Unweighted_Find_Nearest (R)

- *Set* $S_0=0$ *,* $Y_0 = R$.
- *for k=1,2……..until stopping criteria = true*
  1. $A_k = Y_{k-1} - S_{k-1}$
  2. *Do the eigendecomposition of* $A_k = V*D*V^{-1}$*: D is diag(λ), where λ's are eigenvalue of* R_k.
     - *Find D1 by making all negative* $d_{ii} = 0$*, i.e. replace all the negative eigenvalues with 0.*
     - $X_k = V*D_1*V^{-1}$
  3. $S_k = X_k - A_k$
  4. $Y_k = X_k$
  5. *Set* $y_{ii} = 1$
- *end for loop*

end Unweighted_Find_Nearest

The procedure mentioned line-2 inside the for loop is the projection from correlation matrix set S onto the positive semi definite set P, and line-5 finds the opposite projection that is projection from set PS onto set S. The resultant matrix will be positive semi definite (if the correlation matrix R is indefinite), it can be

made positive definite as mentioned in previous section. The uniqueness of the solution and optimality theorems related with algorithm-2 is explained in [1].

Numerical experiments:
The following example is designed to show the different ways of optimizing an indefinite correlation matrix and how much the solution differs in different approaches. Let's look at a simple correlation matrix of size 3 x 3:

$$R = \begin{pmatrix} 1 & 0.9 & 0.85 \\ 0.9 & 1 & 0.2 \\ 0.85 & 0.2 & 1 \end{pmatrix}$$ A simple observation shows that there is something wrong in the correlation between (2,3) because it is low even when the correlation between (1,2) and (1,3) is high. It means that when (1,2) and (1,3) are highly correlated then so (2,3). The eigen values of matrix R are -0.1442, 0.8003, and 2.3418. One negative eigen value is making this matrix indefinite. Now, suppose we want to change the error full element R(2,3) only by increasing this value and not effecting the other ones. Changing R(2,3) to 0.68 (call this matrix as R_new) will hev eigen values 0.0036, 0.5371, and 2.4594. New matrix is positive definite and the *Frobenius norm* of the difference $\| R\_new - R \|$ = 0.48.

The optimal correlation matrix for equal weights to all elements, based on the algorithm-2 is:

$$X = \begin{pmatrix} 1 & 0.8122 & 0.7697 \\ 0.8122 & 1 & 0.2530 \\ 0.7697 & 0.2530 & 1 \end{pmatrix}$$ the eigen values of X are 0.0001, 0.7475, and 2.2524. And

$\| X - R \| = 0.1842$ The previous norm $\| R\_new - R \|$ is about 160% higher than this optimal one.

Now let us consider the case when we want to restrict the relative change in the individual elements of correlation matrix R. Suppose, based on some uncertainty calculations we end up with a weight matrix W on the individual elements of R.

$$W = \begin{pmatrix} 1 & 0.95 & 0.80 \\ 0.95 & 1 & 0.10 \\ 0.80 & 0.10 & 1 \end{pmatrix}$$ the dialgonal elements of W will always be 1 because the diagonal elements of a correlation matrix are always 1. Also the weight matrix will be symmetric.

The optimal nearest positive definite matrix is (based on algorithm-1):

$$X_W = \begin{pmatrix} 1 & 0.8617 & 0.8106 \\ 0.8617 & 1 & 0.4014 \\ 0.8106 & 0.4014 & 1 \end{pmatrix}, \quad \| X_W - R \|_W = 0.1157$$

As we see the relative change in the individual matrix element agrees with the weight matrix W.

Numerical test has been performed on algorithm-1 and algorithm-2. Correlation matrix R was created randomly by making unit diagonal elements and generating uniformly distributed random numbers between 0 and 1 for upper half of non-diagonal elements of R. Then, taking lower half same as of the upper half of R in order to make it symmetric correlation matrix. The probability of getting an indefinite correlation matrix by this random method is nearly 1 when the R is large enough (more than 6 x 6). Numerical experiments on many randomly generated correlation matrices in the case of unweighted matrix norm give same result in both algorithms. The algorithms were implemented in "MatLAb-7.5". Comparison in terms of time between algorithm-1 and algorithm-2 is shown in the figure-1. The code was run on Intel Dual 2GHz, 3GB memory computer. As we see algorithm-2 is very fast, it takes about only 4 seconds for matrix of size 150 x 150. Whereas algorithm-2 is very slow as compared to previous one. In

the plot run time was determined by taking the average of 10 run times for same matrix dimension. A cubic function has been fitted for algorithm-1 based on run up to matrix dimension of 50 x 50:

$$t = (6.92\ x\ 10^{-5})\,n^3 + (0.0099)\,n^2 - (0.1495)n + 0.8428 \qquad \text{seconds},$$

where n is the dimension of correlation matrix R

Based on this the run time for algorithm-2 having 150 x 150 matrix size will be about 435 seconds. The cubic approximation seems valid because eigenvalue decomposition of a square matrix of size n has mostly order $\Theta(n^3)$.

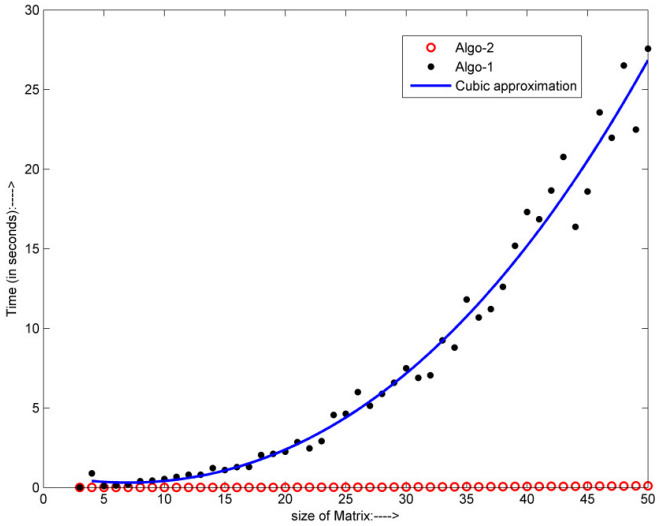

Figure 1: Comparison between run times of algorithms

To see the effect of weights on the final matrix we used the same matrix R used in the example above. Figure-2 represents the value of changed matrix elements when weight corresponding to R(2,3) is increased, but other weights are equal to 1.

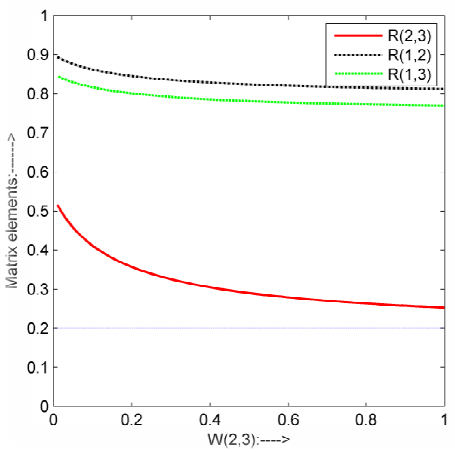

Figure 2: Effect of weight on matrix R

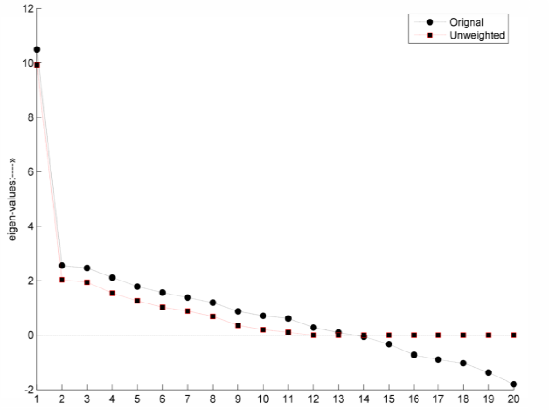

Figure 3: Eigen values of indefinite matrix and optimal one.

Sorted eigen values of a randomly generated indefinite correlation matrix are plotted in figure-3 along with the unweighted optimal positive definite matrix. As we see all the negative eigen values of original matrix shifts near to zero and the remaining positive eigenvalues decreased.

Comparison with a method presented in CCG report [see 4]: Comparison was done on a randomly generated correlation matrix of size 20 x 20. The method presented in [4] generates a positive definite

correlation matrix but the difference of norm between fixed and original one was 13.2, whereas algorithm-2 gives an optimal matrix with difference in norm equal to 3.3.

Concluding remarks

This work presents the methodology to fix the indefinite correlation matrices. Algorithm-1 and 2 guarantee to find the nearest positive definite matrix. Currently we are investing on (1.) How to select an appropriate weight matrix based on correlation coefficient uncertainty in order to incorporate the desired relative change in the elements of the correlation matrix, and (2.) The effect and possible solutions of ill-conditioned optimal matrix.

Program Documentation

Two separate subroutines were written to implement algorithm-1 and algorithm-2. These subroutines were written in FORTRAN and as well as in MATLAB. The details of FROTRAN subroutines are as follows:

Fix indefinite correlation matrix by means of weights, FixCorrMat_weight This program finds the nearest positive definite matrix based on Algorithm-1. The structure of this subroutine is as follows:

FixCorrMat_weight(R,W,n,C)
INPUT ARGUMENTS:

- **R**: a square matrix (must be 2 dimensional array of size n x n)
- **W**: weight matrix, same as size of R, Each element of W, W(i,j) corresponds to the confidence in the correlation coefficent R(i,j). If W(i,j)=1, the fixed matrix C will have C(i,j)=R(i,j). Thus all the diagonal elements of W must be 1. and W(i,j)<=1,
- **n**: Size of the matrix (a square matrix of size n by n).

OUTPUT ARGUMENT

- **C**: The fixed correlation matrix returned by subroutine.

Fix indefinite correlation matrix without weights, FixCorrMat
This program find the nearest positive definite matrix based on Algorithm-2 as presented in section-3.2. The structure of this subroutine is as follows:
FixCorrMat(R,n,C)
INPUT ARGUMENTS:

- **R**: a square matrix (must be 2 dimensional array of size n x n)
- **n**: Size of the matrix (a square matrix of size n by n).

OUTPUT ARGUMENT

- **C**: The fixed correlation matrix returned by subroutine.

References:

[1] **Higham, NJ** (2002). "Computing the Nearest Correlation Matrix – A Problem from Finance", *IMA Journal of Numerical Analysis*, 22, pp. 329-343.

[2] **Pietersz, R and PJF Groenen** (2004) "Rank Reduction of Correlation Matrices by Majorization", *Econometric Institute Report EI 2004-11*, Erasmus Univ. , Rotterdam.

[3] **Qi, H and Sun,D** (2005) "A Quadratically Convergent Newton Method for Computing the Nearest Correlation Matrix", *SIAM Journal on Matrix and Application s*, 22, pp. 360-385.

[4] **Manchuk J.G., Deutsch, C.V.** (2008) "Robust Solution of System of Equations in Geostatistics", CCG *Annual report-2008*, 402.
[5] **Goovaerts, Pierre** (1997) "Geostatistics for Natural Resources Evaluation", *Applied Geostatistics Series, pp. 104-124.*