

New updated BlockSIS program with MDE

Yupeng Li and Clayton V. Deutsch

Sequential indicator simulation (SIS) remains a reasonable approach for stochastic simulation of categorical variable such as facies or geological units. There are a number of techniques implementing SIS in practice varying in how they integrate the secondary data. The original BlockSIS program collected nine of them. In this new updated program, the multivariate probability distribution estimation (MDE) is added. The algorithm and program parameters are presented out in this paper.

Introduction

Indicator techniques are commonly used for categorical variables such as facies and rock types before other continuous properties. Different indicator variograms are used for each rock type. Secondary data are easily coded as soft indicator data for integration with a form of cokriging or non-stationary kriging. Indicator kriging (IK) and sequential indicator simulation (SIS) are widely used. IK estimates the probability of each given the available data. SIS samples the conditional probabilities in a Monte Carlo simulation fashion. IK and SIS are robust and provide a straightforward way to transfer uncertainty in categories through to the geostatistical models. The required variogram models are easy to infer from hard data. A number of SIS variations have evolved over years. Most of the variations related to the use of soft secondary data arising from geological interpretation or geophysical measurements.

In this paper, a new conditioning probability distribution procedure is introduced based on transition probability matrices (TPM). The multivariate probability distribution estimation (MDE) process is inferred from the TPM of the conditioning data configuration. The related theories are illustrated earlier in this report. This paper focuses on the program implementation.

The *BlockSIS* code published previously in CCG is open and permits easy integration of new algorithm developments. The program has been modified to calculate estimation results as well as simulated realizations. The new MDE method has also been integrated in the program.

Transition Probability Matrix Construction

Spatial relationships between the conditioning data may be characterized by transition probability matrices at different lag distances. Assuming the geological discrete data such as lithological type is a Markov chain along the vertical profile, which means the lithological type at location X_t will only depend upon the one at the preceding location X_{t-1} . The possible values of X_t will form a countable set $S(k_i)$ ($k_i = 1, \dots, K, i \leq m$) called the state space of the chain, where k_i denote the mutually exclusive, exhaustively defined states of a stochastic process. If a sequence of states has the Markov property, then every future state is conditionally independent of every prior state. The changes of states are called transitions. The conditioning probability of a state $P(X_{n+1} = k_{n+1} | X_n = k_n)$ is called the transition probability. The transition probability of a finite state space of $S(k_i)$ ($k_i = 1, \dots, K, i \leq m$) going from state k_i to state k_j in n steps will form a $m * m$ matrix $T_{i,j}^{(n)}(x, h = n)$ with the i^{th} row and j^{th} column element of $T_{i,j}^{h=n} = P(X_n = k_j | X_0 = k_i)$, called a transition probability matrix (TPM).

Assuming the lithological types at location X_t in the vertical profile of a well will only depend upon the lithological type at the preceding location X_{t-1} , it is recognized as a Markov chain process. The TPM at different observe interval lags are built from observations spaced equally along a vertical profile. Suppose the vertical profile is H which divided into n equal segments using an equal segment. The state space in this Markov chain is the facies category set k_i ($k_i = 1, \dots, K, i \leq m$). Then, in each segment will define a state of a Markov chain and the transition probability of the whole profile will form a TPM.

For example, the total observed number of state K_i followed the state K_j giving the observation interval $h=n$ is denoted as n_{ij} , while n_i is the counted total number of k_i . When the interval is 1, the transition probability from state K_i to state K_j will be denoted as:

$$t_{i,j}^{h=1} = n_{i,j} / n_i$$

The transition probability from k_1 to $k_1, k_2, k_3, \dots, k_m$ is given by $t_{1,j}^{h=1}$ ($j=1,2,\dots,m$) in the first row and so on for the second line and others, then the matrix will be:

$$T^{h=1} = \begin{pmatrix} t_{1,1}^{h=1} & t_{1,2}^{h=1} & \dots & t_{1,m}^{h=1} \\ t_{2,1}^{h=1} & \dots & \dots & t_{2,m}^{h=1} \\ \dots & \dots & \dots & \dots \\ t_{m,1}^{h=1} & \dots & \dots & t_{m,m}^{h=1} \end{pmatrix}$$

Generally, from the account approach, the elements of the TPM ($\forall h$) will be:

$$t_{i,j}(x, h) = \left(\frac{n_{i,j}}{n_i} \right) = \left(\frac{n_{i,j} / \sum_{i,j} n_{i,j}}{n_i / \sum_{i,j} n_{i,j}} \right) = \frac{p_{i,j}(x, h)}{p_i(x, h)}$$

Where: $t_{i,j}(x, h)$ is the transition probability of state K_i to state K_j

$n_{i,j}$ is the number of state K_i followed by K_j after h steps;

n_i is the row sum of the $n_{i,j}$, $n_i = \sum_j n_{i,j}$;

$\sum_{i,j} n_{i,j}$ is the whole sum of the tally matrix entries;

$p_{i,j}(x, h)$ is the joint probability of two state K_i and K_j ;

$p_i(x, h)$ is the univariate marginal probability of state K_i

The TPM has the following properties: (1) Its elements are non-negative, $0 \leq t_{i,j} \leq 1$; and (2) The elements of each row sum up to one, $\sum_{j=1}^m t_{i,j}(x, h) = 1$; Generally,

$$t_{i,j}(x, h) = p\{k_j \text{ exit at } (x+h) | k_i \text{ exit at } x\} = \frac{p\{k_j \text{ exit at } (x+h) \text{ AND } k_i \text{ exit at } x\}}{p(k_i \text{ exit at } x)}$$

When the process is stationary or homogenous, the transition probability is independent of position x , the transition probability $t_{i,j}(h)$ and the bivariate joint probability $p(\mathbf{h}; k_i, k_j)$ will depend only on the intervals vectors. From the transition probability $t_{i,j}(x, h)$, the bivariate joint probability $p(\mathbf{h}; k_i, k_j)$ ($\forall \mathbf{h}; k_i, k_j; i, j = 1, \dots, K$) can be calculated as:

$$p(h, k_i, k_j) = p\{k_i \text{ exit at } (x+h) \text{ AND } k_j \text{ exit at } x\} = t_{i,j}(h) * p_i(h)$$

That's to say, there also exists a bivariate joint probability matrix for a particular lag h which is fully defined by its relatively transition probabilities matrix and the univariate marginal probability.

In the MDE approach, the spatial relationship between the data and unsampled location are characterized by the transition probability. It is related to the variogram model in traditional geostatistics. A GSLIB style program *calctp.for* will perform TPM construction. Figure 1 is the parameter file for *calctp.for*. In

line 1, it is the totally category number in the vertical profile. The **line 2** is the category type in the data file. **Line 3** is the input data file name. **Line 4** is the column number for well ID, depth and category data. In **line 5**, the first data is for the minimum count interval along the depth: *xlag*, the second number is the maximum observer lag *nlag*. These two parameters are very important for building the reasonable transition probability matrices. When the maximum observation lag is large enough, the bivariate joint probability will be equally to their independent case. $P_{i,j}(x, h) = p_i * p_j$. **Line 6,7,8 and 9** are all the output files name including the bivariate joint probability output file for MDE, bivariate joint plotting output file, transition probability plot file and the debug file during the calculation. The plot files are all in variogram model format and can be plot using gslib file *vargplt*. Figure 2 is one example of the transition probability matrices plotting.

Vertical to Horizontal TPM transform

Usually, in the vertical profile of the well data or outcrop, the data are sampled with a high density which can build the transition probability matrices easily. While for estimation and simulation, the grid needed are always located in a 3D space. In sedimentology, it is pointed out that in a simple sedimentary cycle, the sedimentation in a vertical sequence of facies will mirrors the original lateral distribution of sedimentation. "Sedimentary environments that started out side-by-side will end up overlapping one another over time due to transgressions and regressions". This rule is called Walther's law and named after a German geologist. Walther's law provided a clue to related lateral facies changes to vertical changes (vertical succession). In the program parameter file will ask three parameters named as *astio1*, *astio2*, *astio3* in the MDE approach. The first two is for horizontal direction, along the main and minor difference direction. The last one is the vertical ratio. The details of this point are illustrated in paper 103 of this report.

MDE based on bivariate joint probability

After the bivariate joint probabilities are inferred, the main concern is the construction of a multivariate joint distribution given these sets of bivariate joint probabilities. In general, assume in the domain there are n locations ($u_1 \dots u_n$), each of the n location has K categories. That is to say, the n random variables will form a multivariate distribution function(pdf) $P_{MV}(u_1, \dots, u_n)$, which is defined as below and represents the probability of a specific configuration of categories $k_i (k_i = 1, \dots, K, i \leq n)$ existing at locations $u_1 \dots u_n$.

$$P_{MV}(u_1, u_2, \dots, u_n) = \text{prob}(u_1 \in k_1, u_2 \in k_2, \dots, u_n \in k_n); k_i = 1, 2, \dots, K$$

In this distribution, there are totally K^n possible values. Each of these values occurs with a given frequency which is identified with an index that is:

$$I_{mv} = 1 + \sum_{n=1}^N (u_n - 1) * K^{n-1}$$

Where: $I_{mv} = 1, 2, \dots, K^n$; u_n is the code of the n^{th} location in the data configuration that identifies its categories; K is the total number of the categories. Given a known multivariate probability distribution $P_{MV}(u_1, \dots, u_n)$, the univariate marginal probability $p(u_j, k')$ for any category k' at any location $u_j (j = 1, \dots, n)$ calculated from multivariate probability will be:

$$p(u_j, k') = \sum_{k_1=1}^K \sum_{k_2=1}^K \dots \sum_{k_j=k'} \dots \sum_{k_n=1}^K P_{MV}(u_1, \dots, u_n)$$

Where: $k_i, k' = 1, 2, \dots, K; i, j = 1, \dots, n$. Because the univariate proportion will be locked when bivariate joint probability are satisfied. After the bivariate probability obtained from the vertical profile, there are already embedded another univariate marginal probability $p^*(u_j, k')$. If these two univariate marginal probability $p(u_j, k')$ and $p^*(u_j, k')$ are characterize the same multivariate probability, they should be

consistent to each other. That is to say, the univariate marginal of categories $p_{aim}(\mathbf{h}; k')$ calculated from the multivariate probability should be consistent with those calculated from the bivariate joint distribution $p_{cal}(\mathbf{h}; k')$.

Scaling the bivariate joint probability with the univariate marginal probabilities is needed here. An iterative modification process is adopted. The process of bivariate joint probability modification with univariate marginal probability is:

- Step 1: calculate the univariate marginal probability $p_{cal}(\mathbf{h}; k')$ from the bivariate joint probability

$$p_{cal}(\mathbf{h}; k') = \sum_{k''=1}^K p_{aim}^*(\mathbf{h}; k', k'');$$

- Step 2: calculate the modification factors F_{univ} for all the bivariate joint probability;

$$F_{univ} = \frac{p_{aim}(\mathbf{h}; k')}{p_{cal}(\mathbf{h}; k')}$$

- Step 3: modify the bivariate joint probability to get a new bivariate joint probability;
- Step 4: check whether it is consistent or not, if not, use this new bivariate joint probability as input of step1 and continue another modification; if yes, stop; the final bivariate joint probability denoted as: $p_{aim}(\mathbf{h}; k', k'')$;

For the specification location, the univariate proportion maybe different with that in bivariate joint probability. It will be an inconstant problem. This scaling process also provides a chance to integrate secondary information into the algorithm. After the correction, the new bivariate marginal carry all the two-point spatial information.

Given a known multivariate probability distribution $p_{MV}(u_1, \dots, u_n)$, the bivariate joint probability calculated from this distribution will be:

$$p_{cal}(u_{j_1}, u_{j_2}, k', k'') = \sum_{k_0=1}^K \sum_{k_1=1}^K \dots \sum_{k_{j_1}=k'} \dots \sum_{k_{j_2}=k''} \dots \sum_{k_n=1}^K p_{MV}(u_0, u_1, \dots, u_n) = p_{cal}(\mathbf{h}; k', k'')$$

Where: $k_i, k', k'' = 1, 2, \dots, K$; $j_1, j_2 = 0, 1, \dots, n$; $j_1 \neq j_2$; $i = 0, 1, \dots, n$

According the conditioning data configuration, the bivariate joint probability transformed from the vertical profile and modified by the univariate marginal probability is donated as $p_{aim}(\mathbf{h}; k', k'')$; those two bivariate joint probability distributions ($p_{cal}(\mathbf{h}; k', k'')$ and $p_{aim}(\mathbf{h}; k', k'')$) should be consistent to each other. This will impose $(n * (n - 1) * K^2)$ constraints on the multivariate probability distribution.

So now, the MDE iteration scheme based on these bivariate joint probability constraints will be:

- Step 1: calculate the initial multivariate probability distribution. It assumes that the variables at each location are independent. The multivariate probability distribution will be:

$$p_{MV}(\mathbf{u}_1 = k_1, \dots, \mathbf{u}_n = k_n) = \prod_{j=1}^n p_{k_j} \quad (k_1, \dots, k_n = 1, \dots, K)$$

- Step 2: calculate the bivariate joint probability from the initial multivariate probability, denoted as: $p_{cal}(\mathbf{h}; k', k'')$;

- Step 3: based on the conditioning data configuration, aiming bivariate joint probability transformed from the vertical profile is denoted as $p_{aim}(\mathbf{h}; k', k'')$, then the modification factor F_{biv} for the initial multivariate probability will be:

$$F_{biv} = \frac{p_{aim}(\mathbf{h}; k', k'')}{p_{cal}(\mathbf{h}; k', k'')}$$

- Step 4: In a randomly modification path, modify the initial multivariate probability distribution by all the bivariate joint probability between any two locations. After standardization a new multivariate probability $p_{MV}^*(\mathbf{u}_0 = k_0, \mathbf{u}_1 = k_1, \dots, \mathbf{u}_n = k_n)$ will obtained.
- Step 5: Check whether all the bivariate joint probabilities are reproduced or not. If not, the distribution of $p_{MV}^*(\mathbf{u}_0 = k_0, \mathbf{u}_1 = k_1, \dots, \mathbf{u}_n = k_n)$ will act as the initial multivariate probability distribution and go to step 2 and do the next modification. If yes, the iteration will stop.

After the multivariate probability distribution is obtained; the conditioning probability for the unsampled location can be calculated according the Bayes' law:

$$p(u_0 | u_1, u_2, \dots, u_n) = \frac{P_{MV}(u_0, u_1, \dots, u_n)}{P_{MV}(u_1, \dots, u_n)} = \frac{P_{MV}(u_0, u_j = k_j)}{\sum_{k_i, k_j=1}^K P_{MV}(u_0 = k_i, u_j = k_j)}$$

where: $u_0, u_1, \dots, u_n = 1, \dots, K; i, j = 1, \dots, n$

The denominator will be sum of several multivariate probabilities that specified by the conditioning data category values; the nominator is the specific category's multivariate probability at the unsampled location.

While in iteration, the calculated bivariate joint probabilities are not exactly the same as the aiming bivariate joint probability although they are very close to each other. So, in the program, after certain times iteration, it will calculate the maximum conditioning probability difference $\max(P_{diff}(k), k = 1, \dots, K)$ from the iterated multivariate probability distribution and decide stop iteration or not. If the maximum conditioning probability difference is less than 0.005 in three continuous iterations, the program will stop iteration. The automatic stop scheme also saves CPU time greatly.

MDE estimation and simulation algorithm

For estimation and simulation process using MDE, it is the same as the sequential simulation in traditional geostatistics, the estimation is based on hard data and the simulation will be a sequential style as shown in figure 3. The main steps are:

- Reading the data and bivariate joint probability matrix;
- Pick up a path (orderly path for estimation and randomly path for simulation) visiting all locations;
- Search the conditioning data (and the previous simulated nodes if for simulation);
- Calculated the distance and transform to vertical direction using the anisotropy ratio and setting up the bivariate joint probability matrix;
- Correct the bivariate joint probability matrix by the univariate probability at this unsampled location;
- Iterate from the initial multivariate probability distribution to satisfy the constraints of the bivariate joint probability to get the multivariate probability distribution;
- Calculate the conditioning probability based on the conditioning data category and the multivariate probability distribution using Bayesian law;
- If estimation, output the conditioning probability, if simulation, using Monte Carlo draw a category value from the conditioning probability distribution;
- Go to the next node until all the nodes are estimated (simulated);

Program

The new updated program follows standard GSLIB styles. Some small modifications are done on the original codes making it possible to do estimation and simulation using this single program. Three source files are required: *BlockSIS.for*, *BlockSISsubs.for* and *mdesub.for*. The parameter file for the new *BlockSIS* is shown in Figure 4.

The kriging option is specified on **line 1**. Totally, there are 10 options. The *isimtype* 9 is *MDE* approach. In the original paper, suggestions on choosing a right option from 9 kinds of simulation algorithms are very helpful. Now, the MDE approach is ready for compare the results with other algorithm. From **line 2** to **line 8** is the same as the original program. If *isimtype* is 9, the MDE approach will ask for a file providing the bivariate joint probability calculated from the vertical profile of well data. The **line 9** is asking for that file's name. **Line 10** is the totally lag number *nlag* used in the transition probability construction. It is the same number as named "maximum observer lag" *nlag* in program *calctp*. **Line 11** are three vertical-horizontal transformation anisotropy ratio for x,y and z direction separately. In **line 12**, first number is the maximum iteration time during MDE process, the second number is the option for using auto-stop or not. If it is 1, during the iteration, the program will check after certain times iteration to see whether the conditioning probability is stable or not. If the maximum change for totally K categories during three times is less than 0.005, the program will stop iteration. **Line 21** is the number of realization, 0 means the program will do estimation, if not 0, the number will be the simulation realization number. **Line 26** is the maximum original data for each kriging, **line 27** is the maximum previous nodes used in each kriging. While in MDE approach so far, because the CPU time of iteration, the program will make sure the sum of those two numbers are less than 7, the maximum number will be 11. **Line 28** is specifies whether the data are assigned to the grid node location or not. In program, it is enforced that no data are assigned to grid node location in kriging estimation and in MDE estimation and simulation. The left lines are the same parameters as original *BlockSIS* program.

Estimation and simulation example

A small 3-category example is used for illustration. The 3 categories cases are representative test. In most cases, the categories in reservoir can separated into 3 categories according their qualities: non-net, media and net sand. The data set location map is shown in upper left of Figure 5. The variogram modes for those three categories are also shown in upper right of Figure 5. The transition probability matrix for those three categories comes from a true well data. The totally count number is 40, and then the transition probability matrices is shown in bottom of Figure 5. The research grid is 10 by 10 grid nodes (100 total).

As already stated out before, the anisotropy ratio will affect the MDE results greatly. In figure 6, different estimation results using different anisotropy ratio are presented. When a small ratio used, less information are provided for horizontal estimation from their vertical transition probability. When the ratio is small enough, the bivariate joint probability of any two categories in the horizontally will be $p(\mathbf{h}; k', k'') = p(\mathbf{h}; k') * p(\mathbf{h}; k'')$.

The simulation results using MDE approach and the traditional sisim results are shown in figure 7. The light clean option is used.

Algorithm comparison

In Figure 6 and Figure 7, the MDE results and the traditional IK and SIS results are put together, which formed a visual comparison between different algorithm's results. A well established comparison approach is those different algorithms' cross validation results comparison. In cross validation, it is to compare model results to true values which are held back from the analysis. True data are left out one at a time and re-estimated from the surrounding data. For category variable, one kind of quantity criteria is presented on

Figure 8. The true probabilities for categorical variables at one location are always 0 or 1; the facies is present or not. Estimated probabilities are between 0 and 1. Ideally, when the estimated probability of a facies is high, then the location will likely be that facies; when the estimated probability of a facies is lower, then the location should not be that facies. . The b statistic measures the difference between these probabilities. The b value should be as close to 1 as possible.

From the small test cast, the calculation results are shown in chart 1. From the comparison, the MDE technique has a better estimation for those three variables.

Chart 1 the comparison of IK and MDE estimation results

category	1		2		3	
technique	IK	MDE	IK	MDE	IK	MDE
exist	0.6488	0.8351	0.5629	0.8806	0.5961	0.9532
non-exist	0.1756	0.0825	0.2236	0.0597	0.2071	0.0733
b	0.4732	0.7526	0.3393	0.8209	0.389	0.8799

In another words, the MDE technique requires more CPU time because of the huge multivariate probability distribution dimension. Consider a data configuration with 10 locations and in each location there are 3 categories, the total number of multivariate probability distribution will be 59,047. Bivariate joint probability in this case will be 810; one iteration need to update 59,047 values with these 810 constraints. A realization with 100 grids with 3 categories in estimation needs 156s to finish (Inter Xero CPU 3.19 Hz). While a realization with 2,500 grids; the estimation time will be 11 minutes. In the future, improvement the CPU time to a practical amount of time will be one of the efforts.

Conclusion and future work

This new multivariate probability distribution estimation technique is based on transition probability matrix, which provides the spatial relationships of the bivariate joint probability in the conditioning data. It is non-kriging. There are no order relationship problems in this approach. It opens a new research area for categorical data simulation. As stated in previous session of this paper, integrate the secondary information and improve the CPU calculation time will be the future research.

Acknowledgements

The authors would like to thank Steven Lyster and Chad Neufeld for their help in the codes debugging. And the author also should acknowledge the support of the sponsor companies of the Centre for Computational Geostatistics.

Reference

Deutsch, C.V., *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*, Ph.D. Thesis, Stanford University, 1992.

Deutsch, C. V., and Journel, A.G., 1998, *GSLIB: Geostatistical Software Library*, Oxford University Press, New York.

Deutsch, C. V., A sequential indicator simulation program for categorical variables with point and block data: *BlockSIS, Computers & Geosciences*; 32 2006(10), Pages 1669-1681.

<http://www.wvup.edu/ecrisp/g103lecstratpinciples.html>

parameter file for calctp

Line START OF PARAMETERS:

1	3	--category type number in the vertical profile
2	1 2 3	--category type in the vertical profile
3	hekwell-short.dat	-- input markov chain data file
4	1 4 5	-- well ID,depth, faces codes column
5	0.1 50	--transition count interval, and totally number
6	hekwell_mde.out	-output bivariate for mde in file 1
7	hekwell_biv.out	-output bivariate as the variogram plot format
8	hekwell_tp.out	--output TP as the variogram plot format
9	calctp.dbg	-debug file including the tally matrix, and thickness of each facies etc.

Figure 1 parameter file for program *calctp*

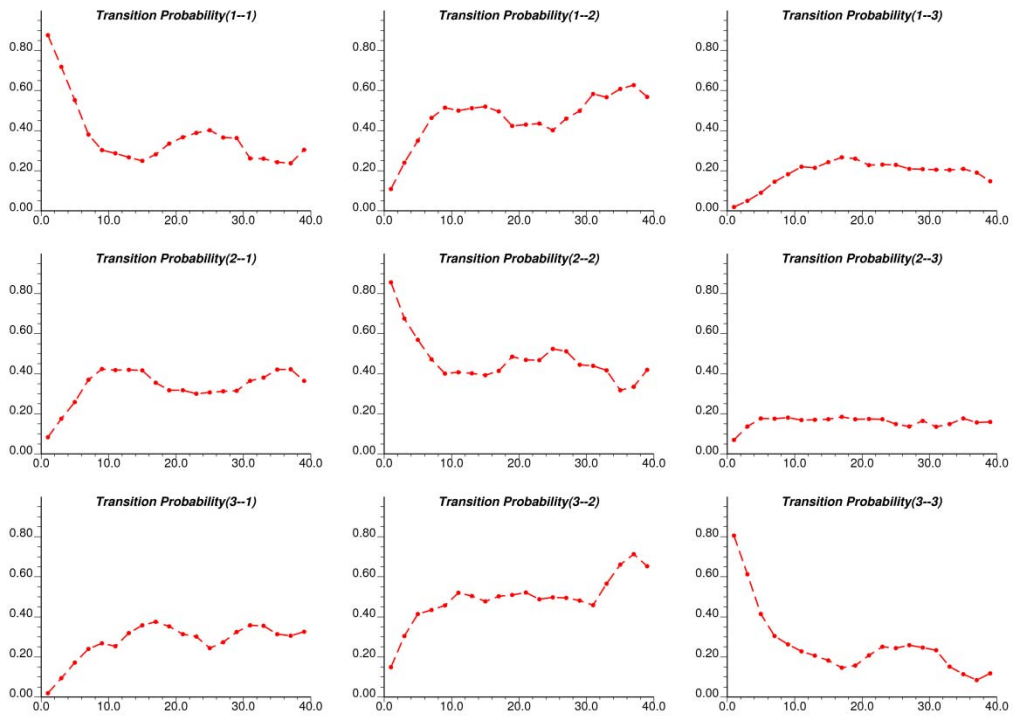


Figure 2 transition probability matrices build from the vertical profile of a well data

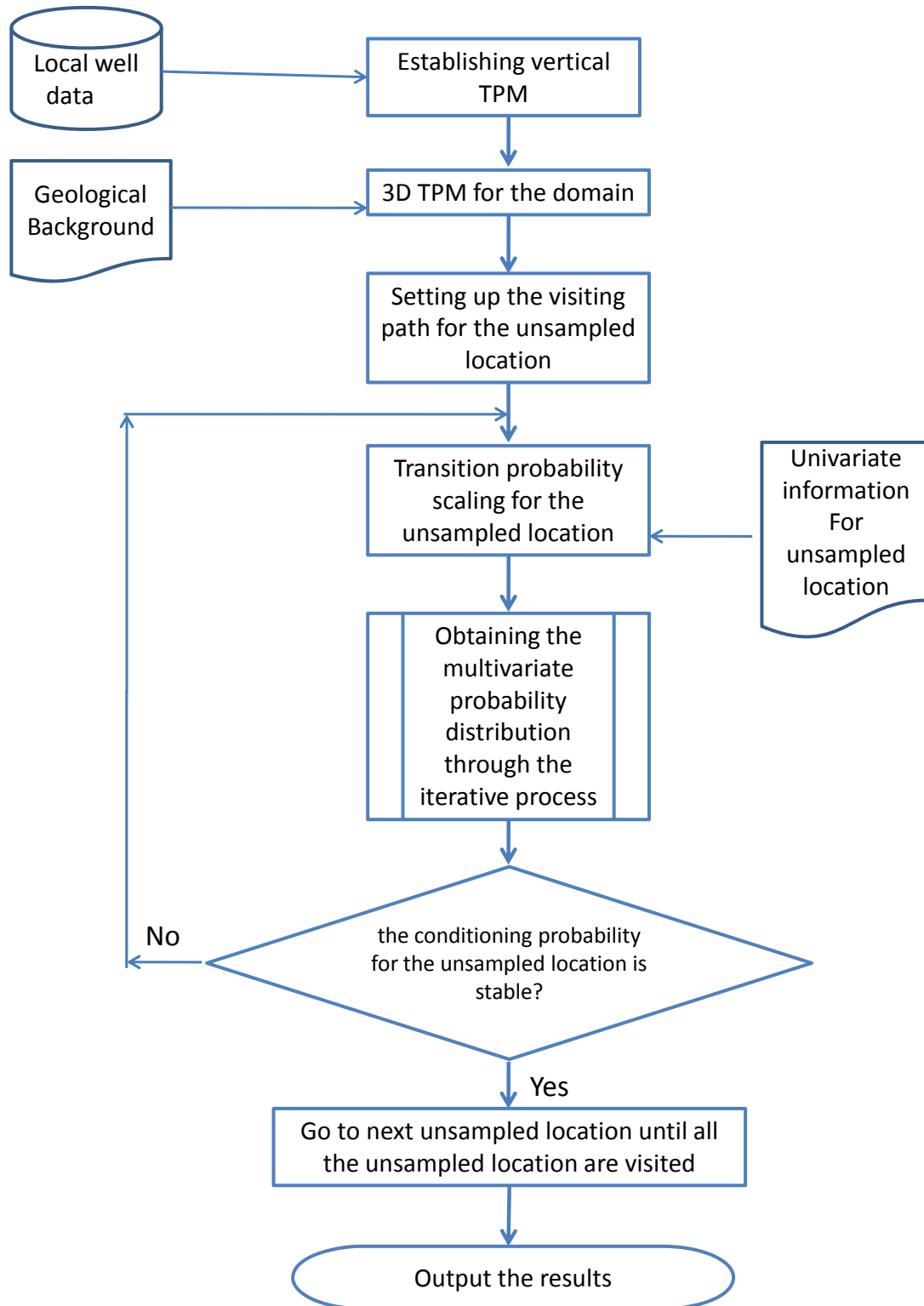


Figure 3 the workflow of the MDE based on TPM in the new BlockSIS program

```

Line  START OF PARAMETERS:
1      9      -0=SK,1=OK,2=L1,3=L2,4=CC,5=BU,6=PR,7=BK,8=BC,9=MDE
2      1      -Clean: 0=none, 1=light, 2=heavy, 3=super
3      3      -number of categories
4      1 2 3  -categories
5      0.36 0.42 0.22 -global proportions
6      0.5 0.50 0.50 -correlation coefficients for soft data
7      test.dat -file with local data
8      1 2 0 3 -columns for X,Y,Z, and category
9      hekwell_mde.out -if mde option used input the bivariate marginal
10     40      -the total lag in vertical TP calculation,
11     10.0 10.0 1.0 -the anisotropy transform ratio for three direction
12     100 0 -maximum iteration time(mde),1:autostop,0:non-autostop
13     000d.out -file with gridded prior mean values
14     1 2 3 -columns for each category
15     2 - 2-D areal map (2) or 3-D cube (3)
16     keyout.dat -file with keyout array
17     1 - column for keyout indicator
18     1 -debugging level: 0,1,2,3,4
19     BlockSIS.dbg -file for debugging output
20     sis_mde.out -file for esitimaiton/simulation output
21     0 -number of realizations(0:estimation;>1:simulation number)
22     10 0.5 1.0 -nx, xmn, xsiz
23     10 0.5 1.0 -ny, ymn, ysiz
24     1 0.5 1 -nz, zmn, zsiz
25     69069 -random number seed
26     20 -maximum original data for each kriging
27     20 -maximum previous nodes for each kriging
28     0 -assign data to nodes? (0=no,1=yes)
29     0 -maximum per octant (0=not used)
30     115.0 115.0 10.0 -maximum search radii
31     30. 0. 0. -angles for search ellipsoid
32     10 10 1 -size of covariance lookup table
33     1 0.2 -cat1 nst, nugget effect
34     1 0.8 20.0 0.0 0.0 -it, cc, angl, ang2, ang3
35     20.0 10.0 1.0 -a_hmax, a_hmin, a_vert
36     1 0.2 -cat2 nst, nugget effect
37     1 0.8 30.0 0.0 0.0 -it, cc, angl, ang2, ang3
38     16.0 11.0 1.0 -a_hmax, a_hmin, a_vert
39     1 0.15 -cat3 nst, nugget effect
40     1 0.85 135. 0.0 0.0 -it, cc, angl, ang2, ang3
41     20.0 14.0 1.0 -a_hmax, a_hmin, a_vert

```

Figure 4 the parameter file for new BlockSIS program

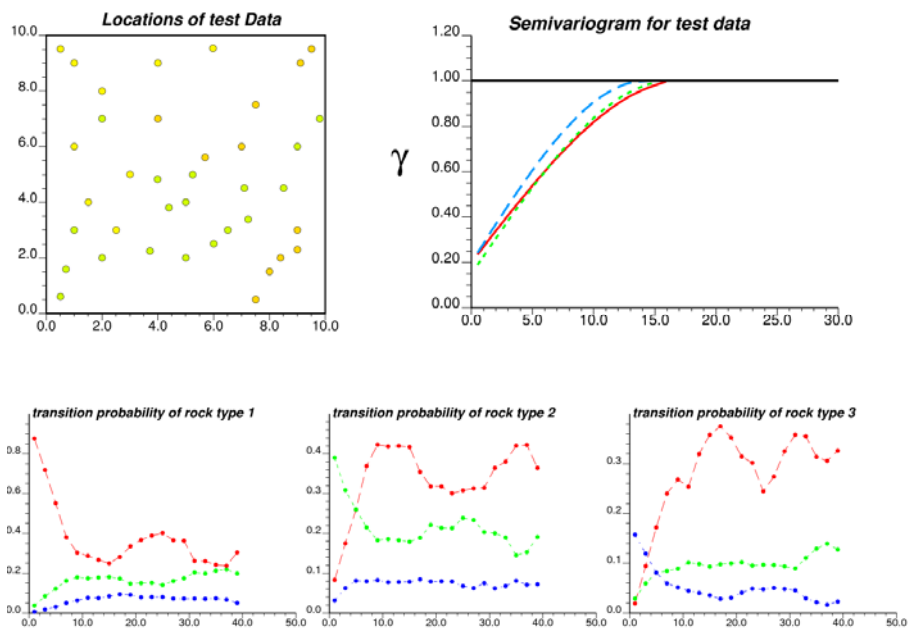


Figure 5 the test data location map, variogram models and transition probability matrices for the test case

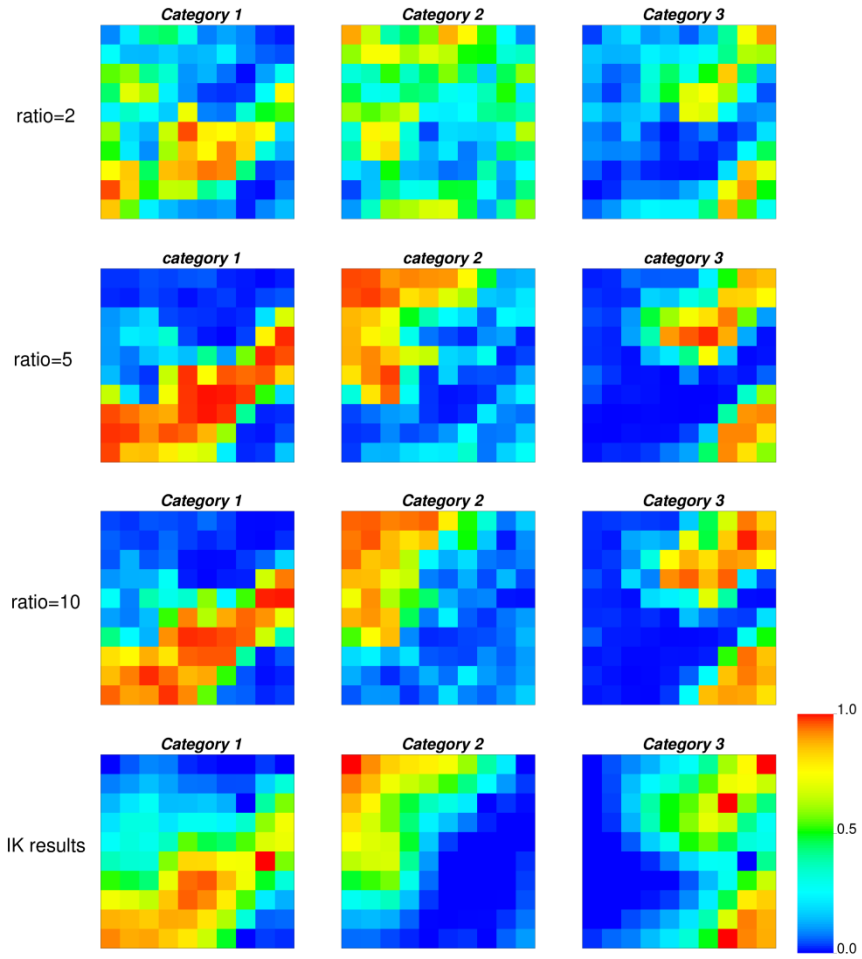


Figure 6 estimation results using different approach

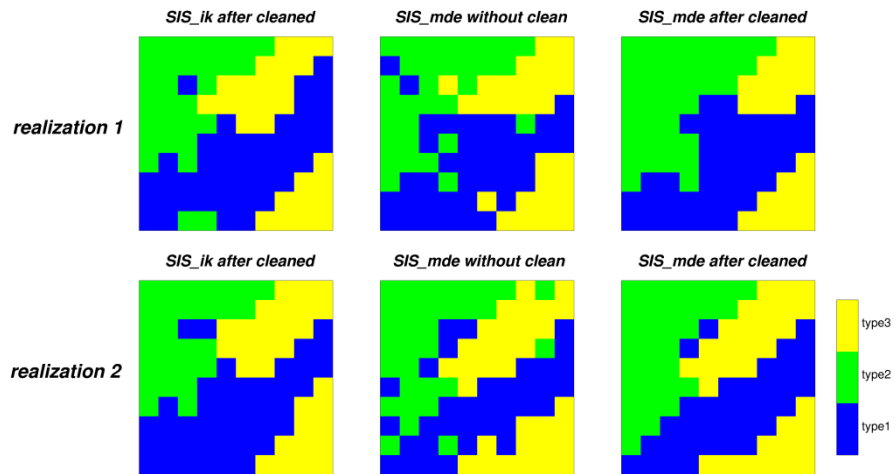


Figure 7 simulation results using different approach

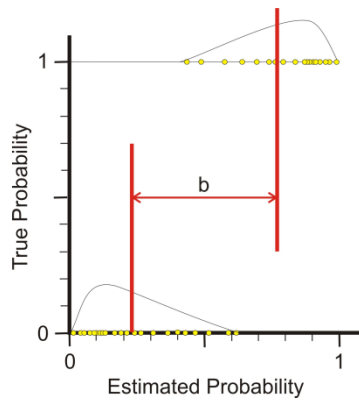


Figure 8 Illustration of cross validation results for categorical variables (Clayton 2008)