A Julia package for optimal spatial stratification using a grid of predictions with error. User's Manual – Version 1.01

Jaap de Gruijter[a,*], Brendan Malone[b]

[a] *postal code NL-6707CG-7*
[b] *Faculty of Agriculture and Environment, Biomedical Building C81, The University of Sydney, NSW 2006, Australia*

The methodology of optimal spatial stratification by package *ospats* is documented in de Gruijter et al. (2015). Users are encouraged to read this article before using the software.

It is assumed here that the user has installed Julia, freely available from https://julialang.org/.

Package *ospats* can be downloaded from https://github.com/jjdegruijter/ospats. The package contains 3 Julia script files:
1. 'main': the script for reading the grid data, setting process parameters and calling the function *ospatsmr*() or function *allocatemr*().
2. 'ospatsmr': containing only the function *ospatsmr*().
3. 'allocatemr': containing only the function *allocatemr*().

> *Explanation*: The functions *ospatsmr*() and *allocatemr*() both lead to a stratification by the Ospats method. The difference is that *opspatsmr*() applies the method to the entire grid, while *allocatemr*() applies the method to a systematic sample from the grid, followed by allocation of the remaining grid points to the strata as constructed from the sample. This function should be used if coarse-gridding is necessary, i.e. if the grid has too many points to enable processing of the pair-wise distance matrix. The requested function is called by setting the parameter *in* (from

*Corresponding author
Email addresses: jjdegruijter468@gmail.com (Jaap de Gruijter), brendan.malone@sydney.edu.au (Brendan Malone)

sampling INterval) in the 'main' script. If $in = 1$, then *ospatsmr*() is called. If $in \geq 2$, then *allocatemr*() is called. For instance, if $in = 3$, then every third grid point gets included in the sample.

1. Initialisation

Before running the script 'main' the following data must be filled in.

Fill in $filename$: name of the file with the grid data.

> *Explanation*: A text file with 5 comma-separated columns without headers, and N (the number of grid points) rows. Default order of the columns is: x-coordinate, y-coordinate, grid point identification, prediction, prediction error variance. In case of a different order, adapt the column numbers in the section 'Reading the data file'. If the data file is incomplete, i.e. not all columns have the same length, then Julia issues a LoadError.

Fill in the following process parameters:
(Note: If an invalid number has been filled in for a parameter (out of range, or not an integer when an integer is expected), then 'main' issues an error message and stops Julia.)

$R2$: squared multiple correlation coefficient, to be obtained from the regression model used for prediction (see de Gruijter et al. (2015) for explanation).

range: the parameter of the exponential auto-covariance of the prediction errors, to be obtained from the regression model used for prediction (see de Gruijter et al. (2015) for explanation).

H: the number of strata, usually in the range of 5 - 15.

> *Explanation*: Generally, the higher H, the higher the precision of the estimate for a given sample size, or the lower the sample size needed to achieve a given precision. However, note that setting H too large may result in strata with sample sizes that are smaller than required, e.g. 3 or 4. If that happens H should be lowered.

maxcycle: the maximum number of iteration cycles allowed for iterative reallocation.

Explanation: This is intended as a safe-guard against unforeseen endless looping. In our experiments the number of iteration cycles needed to fully complete the re-allocation process has never exceeded 100. The setting $maxcycle = 0$ forces the system to skip the iterative re-allocation, and to proceed with the final calculations on the best initial stratification.

runmax: the number of runs of the re-allocation process with different random initial stratifications.

Explanation: Iterative re-allocation of the grid points does not warrant the best possible stratification, as the process may get trapped in a local minimum. To reduce the risk of ending with a sub-optimal solution, *ospats* enables the use of multiple random initial solutions. The best of the resulting stratifications is then selected, i.e. the one with the lowest value of O. Our provisional advice is to do 3 runs.

in: the sampling interval used for systematic sampling from the list of grid points.

Explanation: $in = 1$ then no coarse-gridding is done, and function *ospatsmr*() will be called, which optimises a stratification for the entire grid. If $in > 1$ then function *allocatemr*() will be called, which optimises a stratification after coarse-gridding. The size of the sample is determined by *in*. For instance, if $in = 10$ then every 10th point is included in the sample, starting with a randomly chosen first point. In principle, the sample size should be taken as large as computer capacity allows for calculating the $N \times N$ matrix of generalised distances. Without recourse to supercomputing, that will be in the order of 10^5.

Without coarse-gridding, the computation time depends heavily on the mesh width (δ) of the grid. Disregarding possible boundary effect, N is proportional to δ^2, so the number of pairwise distances that has to be dealt with iteratively is proportional to δ^4. In case of large grids, coarse-gridding is therefore a logical step to keep memory usage and computation time within practical limits.

In case of coarse-gridding by function *allocatemr*() the process consists of two phases. In the first phase iterative re-allocation

is applied to a sample of N points, resulting in an optimised stratification. This stratification is then the basis for allocation of the remaining N_a points in the second phase. The method of allocation is discussed in the next section.

Clearly, the reduction of memory usage and computation time by coarse-gridding will generally go at the cost of an increase of the optimisation criterion $\bar{O}$. The larger the sampling interval, the smaller the sample for the first phase, and the larger the deviation from the true optimum solution may be. It is not possible to predict beforehand how far from optimal the final stratification will be for a chosen sampling interval. However, a check is possible afterwards by comparing the value $\bar{O}$ before and after the allocation.

$n_{\max}$: the sample size for which the Relative Standard Error (RSE) is calculated. The RSE is defined as the predicted standard error of the estimate, expressed as percentage of the mean of the predictions on the grid.

$RSE_{\max}$: limit of the RSE for which the sample size is calculated that is minimally needed for not exceeding it. This sample size is used to calculate the Neyman allocation of sample sizes to the strata, which are then used to draw a stratified random sample.

seed: seed for the pseudo-random number generator, used to randomise sequences of grid points and to draw a stratified random sample.

2. Starting the optimisation process

The optimization process is started by running 'main', calling either of the functions:
1) *ospatsmr*() for *maxrun* runs on the entire grid;
2) *allocatemr*() for *maxrun* runs on a sample from the grid, followed by allocation of the remaining grid points.
(Note: if Julia issues an error message saying that *ospatsmr*() or *allocatemr*() is unknown, then run these functions first before running 'main'.)

See the Appendix for a brief summary of the algorithms for the functions.

```
File name :  Nowley.txt
Grid size : 4382
Seed for random number generator : 1234
Number of strata :  5
Sampling interval :  2
R2 :  0.36
Range :  582
Maximum number of iteration cycles :  100
Number of runs :  3
Sample size for which the RSE is calculated :  50
RSE for which the sample size is calculated :  1.5
Calling function allocatemr
-------------------- START FUNCTION ALLOCATEMR ----------------------
sample size = 2191
     Start run nr  1
Total number of transfers = 4091
Number of iteration cycles = 19
     Start run nr  2
Total number of transfers = 3815
Number of iteration cycles = 22
     Start run nr  3
Total number of transfers = 3715
Number of iteration cycles = 16
Obar values from the runs: [2.14522,2.14522,2.14402]
FINAL RESULTS:
Lowest Obar (O/n) from the sample = 2.144024369800265
Obar (O/N) from the entire grid : 2.1516566918103823
Size of the sample-strata: [131 707 390 352 611]
Size of grid-strata : [258 1425 784 721 1194]
Contributions to O from grid-strata (cbObj) : [1766.91 2227.98 1695.28 1655.46
2082.92]
Mean of z-predictions : 15.663553213693845
Predicted sample size needed to attain RSEmax : 84
Neyman sample allocation : [4.0 31.0 13.0 12.0 24.0]
Predicted RSE for nmax : 1.9426639879314327
---- END OF FUNCTION ALLOCATEMR ----
```

Figure 1: Process parameters and results for stratification with grid data Nowley.txt.

3. Output

The resulting stratification is written in file "Stratification", as a column of stratum numbers for the N grid points. The present version of *ospats*

```
1       1       3855    1517655.54010775 -3635976.61267572
2       1       4211    1517955.54010775 -3636276.61267572
3       1       4207    1517655.54010775 -3636276.61267572
4       1       3859    1517955.54010775 -3635976.61267572
5       2       2348    1511655.54010775 -3634926.61267572

80      5       1943    1514880.54010775 -3634551.61267572
81      5       2600    1516230.54010775 -3635076.61267572
82      5       3169    1513830.54010775 -3635526.61267572
83      5       1952    1515555.54010775 -3634551.61267572
84      5       2478    1514280.54010775 -3635001.61267572
```

Figure 2: First and last 5 sample points of the sample generated by *ospats* with grid data Nowley.txt and process parameters as in Fig. 1

does not provide a map of the stratification. The stratified random sample is written in file "Sample" with five columns: sample number, stratum number, grid point number, x-coordinate and y-coordinate.

Along with the settings of the process parameters, the following final results are displayed on screen from where they can be copied, edited and printed as desired:
1) Lowest value of $\bar{O}$ from the sample used for coarse-gridding;
2) Value of $\bar{O}$ from the entire grid;
3) Sizes of the sample strata;
4) Sizes of the grid strata;
5) Contributions to O from the grid strata (*cbObj*);
6) Mean of the z-predictions ($\bar{\bar{z}}$);
7) Sample size needed to attain $RSE_{\max}$;
8) Neyman allocation of sample sizes to the grid strata;
9) Predicted RSE for $n_{\max}$

See Figure 1 for an example generated by *allocatemr*().

4. Replication test

To check if the package as installed works correctly, run *ospats* on the test data Nowley.txt at https://github.com/jjdegruijter/ospats, with process parameters as given in Figure 1. The resulting sample should be as specified in Figure 2

References

de Gruijter, J.J., Minasny, B., McBratney, A.B., 2015. Optimizing stratification and allocation for design-based estimation of spatial means using predictions with error. Journal of Survey Statistics and Methodology 3, 19–42.

Appendix

Function *ospatsmr*() follows six steps:
1) Calculate the matrix of squared generalised distances between the points of the grid;
2) Generate a random initial stratification;
3) Calculate the contributions from the strata to the objective function O;
4) Optimise the stratification by iterative re-allocation;
Do steps 2, 3 and 4 $runmax$ times, each time recording the final stratification and its associated value of O.

5) For the final stratification with lowest O: calculate the total sample size and Neyman allocation of sample sizes to the strata, given $RSE_{\max}$, and RSE given $n_{\max}$;
6) Draw a stratified random sample from the grid, using the final stratification from step 4, and the total sample size and sample sizes per stratum from step 5. In each stratum, sampling is done with equal probability, without replacement.

Function *allocatemr*() follows six steps:
1) Draw a systematic sample ($Sample$) from the grid, by taking every in-th grid point after a random start point in the range [1, in];
2) Calculate the matrix of squared generalised distances between the points in $Sample$;
3) Apply the Ospats method of stratification to $Sample$, as steps 2 - 4 given above for *ospatsmr*();
Do steps 1, 2 and 3 $runmax$ times, each time recording the final stratification and its associated value of O.

4) Using the final stratification with lowest O: allocate the grid points not in $Sample$ to the $Sample$ strata;

5) Calculate the total sample size and Neyman allocation of sample sizes to the strata, given $RSE_{\max}$, and RSE given $n_{\max}$;
6) Draw a stratified random sample from the grid, using the final stratification from step 3, and the total sample size and sample sizes per stratum from step 4. In each stratum, sampling is done with equal probability, without replacement.