Simple Kriging With Constant Attribute Mean

Download archive GeostatsToolbox.zip from the class Web page, unzip it, add it (with subfolders) to MATLAB’s path, and type `help coords2modelstruct` to verify its availability. Download the binary data file bayAreaData.mat from the class Web page, add it to MATLAB’s path, and import it into our Matlab workspace as `load bayAreaData.mat`.

Array `bayprecip` contains the time-average (from Nov 01 1981 to Jan 31 1982) of daily precipitation in the Bay Area recorded at 77 rain gauges. More precisely, `bayprecip` consists of 4 columns with: #1 rain gauge number, #2 and #3 rain gauge longitude/latitude coordinates (in degrees), and #4 precipitation (in mm/day).

Array `baycoast` contains 2 columns with longitude/latitude coordinates of points comprising a digitized coastline for the study region. The objective hereafter is to predict the unknown precipitation values at \( M = 300 \times 360 \) target locations coinciding with the nodes of a regular 2D grid of size \( (300 \times 360) \), whose specifications are stored in array `GridSpecs`. We will be using Simple Kriging (SK) with (an assumed) known constant mean \( m = \text{mean}(bayprecip(:,4)) \), and an assumed known semivariogram model specified as: `ModelSpecs = [0 0.1 0.10110 ; 29 0.2 0.3 0.13 0.3]`.

1. Display the sample precipitation data as: `scattermap(bayprecip,[2 3],20,4,0,[1 14],[0 1]); hold on; plot(baycoast(:,1),baycoast(:,2),'k-'); hold off`. Compute summary statistics of the sample data and display their histogram using function `univstats`.

2. Define the following distance class midpoints: `lagMid = (0:0.05:1.5)'`, and construct array `LagMidTol = [lagMid 0.05*ones(size(lagMid))]` with midpoints and tolerances for these classes. Compute the omni-directional sample precipitation semivariogram as: `S = scatter2struct(bayprecip,[2 3],[],LagMidTol,[4 4 1])`; and display it, along with the assumed model, as `plotsamplestruct(S,1,{[1 1]},'+'),1,[ ],[ ],ModelSpecs);`

3. Use function `krigep2p`, along with the source data array `bayprecip`, to compute the SK predictions and standard deviations at the target grid nodes as: `precipSK = krigep2p(1,GridSpecs,bayprecip,[2 3],4,ModelSpecs,[ ],0,m);` where the next to last argument 0 toggles the Simple Kriging option. **Note:** You can mask out the predicted values over the ocean as: `precipSK(baymask==0,:) = NaN`; Display the resulting SK predictions as: `rastermap(GridSpecs,precipSK,1,0,[1 14],[0 1])`, and the SK standard deviations as: `rastermap(GridSpecs,precipSK,2,0,[0 3],[0 1])`; comment on the spatial patterns of these 2 sets of values. **Interpret the SK predictions and standard deviations.** Compute summary statistics of the SK predictions using function `univstats` and compare them to those of the source rain gauge data. Last, compare the distribution of SK predictions to that of the source data using function `qqplot` and comment on the results.

4. Repeat Requisite #3, but now reducing the range of the exponential model in the 2-nd row of `ModelSpecs` to 0.3 instead of 1.3, and comment on the results.

5. Use function `postkrige` to compute the probability that the unknown precipitation value at any grid node be \( \leq 5 \) and \( \leq 8 \) as: `iDistr = 0` (Gaussian distribution), `iMode = 2` (probability computation), `precipSKprob = postkrige(precipSK,iDistr,iMode,[5 8]);` Display the exceedance probabilities \((1-\text{precipSKprob})\) using function `rastermap` and comment on the results. Last, use function `postkrige`, with `iMode = 4` and `simPars = [50 2345]`, to generate simulated precipitation values at the prediction locations from the SK-derived local conditional Gaussian distributions. Display a couple of realizations using function `rastermap` and comment on their spatial patterns.
Simple Kriging with Spatially Varying Attribute Mean

In what follows, we will consider a known “expected” precipitation surface stored in array bayprecipexpect, which can be displayed as: rastermap(GridSpecs,bayprecipexpect,1,0,[1 14],[0 1]). The values of this expected surface at the 77 grid nodes closest to the 77 rain gauge locations are stored in the 5-th column of array bayprecip.

1. Construct a new data array with residual source data as: bayprecipres = [bayprecip(:,1:3) bayprecip(:,4)-bayprecip(:,5)], and display them using: scattermap(bayprecipres,[2 3],20,4,0,[-6 8],[0 1]); Compute the mean & variance of these residual data, and compare them to those of the original rain gauge data. Last, compute and display the histogram of these residuals using function univstats and compare it to that of the original data.

2. Compute the sample semivariogram of the residual rain gauge precipitation data as: R = scatter2struct(bayprecipres,[2 3],[ ],LagMidTol,[4 4 1]); and display it as: plotsamplestruct(R,1,[[1 1]],[’+’],1); Compare the resulting semivariogram to that of the original data obtained in Requisite #2 of the previous page. Assume that the residual semivariogram model is given as: ModelSpecsRes = [0 0.1 0 1 1; 2 6.6 0 0.9 0.9 0], and overlay it on the sample residual semivariogram as: plotsamplestruct(R,1,[[1 1]],[’+’],1,[ ],[ ],ModelSpecsRes);

3. Now predict the unknown residuals at the nodes of the (300 × 360) grid using: mRes = 0; precipSKres = krigep2p(1,GridSpecs,bayprecipres,[2 3],4,ModelSpecsRes,[ ],0, mRes); where the last entry mRes is the known mean of the residuals from the expected surface. Display the resulting predictions and standard deviations using function rastermap, and compare them to those obtained in Requisite #3 of the previous page.

4. Add these residual surface to the expected surface to compute the final SK precipitation predictions as: precipSKexp = bayprecipexpect + precipSKres(:,1); Display the results using function rastermap and comment on their spatial patterns. What is the surface of standard deviations associated with these final SK precipitation predictions? Compare the map of final SK predictions to: (i) the one obtained using only the rain gauge data (in Requisite #3 of the previous page), and (ii) the map of expected precipitation stored in array bayprecipexpect.

5. The multiple steps in the Requisites above (for a given residual semivariogram model) can be condensed as: precipSKexp = krigep2p(1,GridSpecs,bayprecip,[2 3],[4 5], ModelSpecsRes,[ ],0,bayprecipexpect); where the residual semivariogram model specification ModelSpecsRes must be passed to function krigep2p.

6. Use function postkrige to repeat Requisite #5 of the previous page but now with the new SK-predictions and error variances stored in array precipSKexp above. Comment on the resulting exceedance probabilities and simulated realizations, comparing them with those you obtained in Requisite #5 of the previous page.