Spatial Statistics

Computer lab – Session 2

Madlene Nussbaum Andreas Papritz (ehem. ETHZ)

15 Oct 2024

Table of contents

1 Geostatistical analysis of `elevation` data (continued)

For the following exercises use the R code of the lecture slides on the introduction into spatial data and the analysis of the Wolfcamp aquifer dataset as template.

You will continue to analyse the `elevation` data from the exercises of *session 1*.

1.1 Maximum likelihood estimation of spatial linear model

So far, you explored the data, you fitted a trend model, you plotted a sample variogram and you fitted a spherical variogram function.

> Task 1
>
> Next estimate the parameters of trend model and variogram jointly by maximum likelihood (ML). Use the same trend model and the same variogram function as before.

Task 2

Compare the ML estimates of the regression coefficients and their standard errors with those obtained before by `lm()`. Compare the ML estimates of the variogram parameters with those obtained by fitting a variogram function to the sample variogram of the residuals.

Task 3

Refit the sample variogram from session 1 (last task). Then, overlay the two estimated variogram functions on the sample variogram of the residuals.

Task 4

Report the confidence intervals for the variogram parameters.

1.2 Residual diagnostics

Task 1

Assess the ML fit of the spatial model by a Tukey-Anscombe plot and a normal QQ-plot of the ML regression residuals $Y_i - \mathbf{d}(\mathbf{x}_i)^{\mathrm{T}\hat{}} = \hat{E}(\mathbf{x}_i) + \hat{Z}_i$.

Task 2

Create in addition a normal QQ-plot of the estimated independent errors $\hat{Z}_i$ and of the spatially correlated component $\hat{E}(\mathbf{x}_i)$ of the residuals. Comment on the plots.

Hint

Use `?residuals.georob`, `?rstandard.georob` and `?ranef.georob` to see how to extract the various types of residuals from the fitted `georob` object.

1.3 Comparing alternative models

Task 1

If you have not yet done so, fit now a spatial model that uses a quadratic function in the coordinates (second order polynomial) as trend model. Use again ML for estimating the regression coefficients and the variogram parameters.

Task 2

Fit in addition a model that uses a linear function of the coordinates (first-order polynomial) as trend model.

Task 3

Use a Wald test to compare the fit of the two models. Compare also the fitted variogram parameters of the two models.
Which model fits the `elevation` data better?

1.4 Stepwise model selection

Task 1

Then use stepwise backward and forward covariate selection by `step()` to simplify the model. Use the argument `fixed.add1.drop1=FALSE` when using `step()`.
This has the effect that the variogram parameters are re-fitted for each tested model. What model minimizes the AIC?

Task 2

Compare the three models by 10-fold cross-validation. Which model do you prefer based on bias (`me`) and root mean square error (`rmse`)?

1.5 Computing kriging predictions

Task 1

Using the model which minimized AIC in the previous problem, compute the kriging predictions of `height` for a rectangle with the lower left corner in (0.1, -0.1) and top right corner in (6.4, 6.3). Use a grid increment of 0.05 when you generate the nodes of the grid for which you compute the predictions.
Convert the dataframe with the prediction results to a `SpatRaster` using the functions `rast` of the package `terra`.

Task 2

Plot the predicted `height` and its prediction standard error along with the lower and upper limits of pointwise 95%-prediction intervals.

Task 3

For comparison, we plot also the quadratic trend surface along with the bubble plot of the regression residuals superimposed. Use the `points` second level plot function to add the bubbles.

2 Influence of variogram parameters on universal kriging predictions

Load the `geoR` package which provides the `wolfcamp` data. Create by the following code a `data.frame` that contains a grid of prediction points.

```
library(geoR)
d.wolfcamp.grid <- expand.grid(
  x=seq(from=-250, to=250, length=52),
  y=seq(from=-150, to=150, length=32)
)
```

Then use the following code to explore the effect of the variogram parameters on the UK predictions:

```
persp(
  krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
    krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=1000, range=100),
```

```
    nugget=3000)),
  theta=150, phi=30, d=5, scale=FALSE,
  expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure"
)
```

Task 1

Vary the nugget-to-sill ratio for a total sill (= nugget + sill) equal to 4000.

Task 2

Using a constant nugget-to-sill ratio of 1:3 and a range equal to 100, compare the predictions for a total sill of 400 and 4000.

Task 3

Vary the range for a nugget-to-sill ratio equal to 1:3.