UBC Spatial Stats Course IV

Jürgen Pilz

Institut für Statistik
Universität Klagenfurt
Universitätsstr. 65-67, 9020 Klagenfurt, Austria
juergen.pilz@uni-klu.ac.at

November 16, 2010 / UBC Vancouver

Likelihood approach: Non-Gaussian Fields

A practical tool for handling small to moderate deviations from Gaussianity is the use of the **Box-Cox transformation**, which is characterized by a parameter λ. The original data z_i at given locations x_i are transformed to new data $y_i; i = 1, ..., n$ given by

$$y_i = g_\lambda(z_i) = \begin{cases} \frac{z_i^\lambda - 1}{\lambda} & : \quad \lambda \neq 0 \\ \log(z_i) & : \quad \lambda = 0 \end{cases}$$

and the transformed data vector $Y = (g_\lambda(z_1), ..., g_\lambda(z_n))^T$ is assumed to follow a Gaussian distribution with

$$E(Y) = F\beta, Cov(Y) = K_\theta = \sigma^2(H_\psi + \tau_0^2 I_n).$$

Here $\tau_0^2 = \tau^2/\sigma^2$ denotes the noise-to-signal ratio, sometimes called the relative nugget effect.

The log-likelihood function is then given by

$$L(Z;\beta,\theta,\lambda) = -\frac{n}{2}log(2\pi) - \frac{1}{2}log\left(det(K_\theta)\right)$$

$$-\frac{1}{2}\left(g_\lambda(Z) - F\beta\right)^T K_\theta^{-1}\left(g_\lambda(Z) - F\beta\right) + (\lambda - 1)\sum_{i=1}^{n} log(z_i),$$

where the last term comes from the Jacobian of the transformation:

$$J_\lambda(z) = \prod_{i=1}^{n}\left|\frac{dy_i}{dz_i}\right| = \left(\prod_{i=1}^{n} z_i\right)^{\lambda-1}$$

The maximization of the log-likelihood function proceeds by looking at the so-called **profile-(log) likelihoods**; e.g. the profile likelihood of λ means L considered as a function of λ, maximized with respect to the remaining parameters β and θ.

Analysis of Swiss rainfall data

The SIC rainfall data come from $n = 467$ locations where rainfall was measured (in units of *mm*). The data can be found in the R-library **geoR**: **data(SIC)**, for detailed information use **help(SIC)**.

Assumptions: Matérn covariance function + constant mean β.

Note: The constant mean assumption we make for the sake of simplicity. A better trend model would be the inclusion of "altitude", you can try this yourself.

Clearly, for large n (not the case here), the profile likelihoods will be hard to compute. An effective approximation strategy is to maximize the likelihood under the *false* assumption of independent observations z_i and ignoring the nugget effect, which leads to

$$\tilde{L}(z, \beta, \sigma^2, \lambda) = (\lambda - 1) \sum_{i=1}^{n} log(z_i) - \frac{n}{2} log(2\pi)$$
$$-\frac{n}{2} log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (g_\lambda(z_i) - \beta)^2$$

instead of L. This results in an estimate $\hat{\lambda} = 0.537$ (check it!). The correct estimates (based on the profile likelihood, see the attached R-code) are

- for $\kappa = 0.5 : \hat{\lambda} = 0.514$ ($logL = -2464.25$)
- for $\kappa = 1.0 : \hat{\lambda} = 0.508$ ($logL = -2462.41$)
- for $\kappa = 2.0 : \hat{\lambda} = 0.508$ ($logL = -2464.16$).

There is little change in response to changes in κ, nevertheless, $\kappa = 1$ is slightly favoured. So, the choice of λ is clear: we choose $\lambda = 0.5$, which implies a square-root transformation of the data.

Now we can go on with the profile likelihood, again using the function **likfit** in geoR. We get the following estimates:

κ	$\hat{\beta}$	$\hat{\sigma}^2$	$\hat{\phi}$	$\hat{\tau}^2$	*log – lik.*
0.5	18.36	118.82	87.97	2.48	−2464.3
1.0	20.13	105.06	35.79	6.92	−2462.4
2.0	21.36	88.58	17.73	8.72	−2464.2

We note the following effects:

- $\hat{\tau}^2$ increases with κ
- $\hat{\phi}$ decreases considerably with increasing κ. This points to the fact that interpretation of $\phi =$ *range* cannot be made independently of $\kappa =$ *smoothness*.

Most importantly, the likelihood fitted values result in a variogram model which is usually in good agreement with the conventional variogram fit at short distances, but differs at medium distances (see the Matérn variogram below).

As a rule, the likelihood fitted variogram discounts the empirical semivariogram values at medium distances.

For producing maps we may use **Monte Carlo** approximations using the geoR- built in functions **likfit**, **krige.control** and **output.control**.

First we create a *likfit* object with our fixed values $\kappa = 1$ and $\lambda = 0.5$, and (rough) initial values for sill, range and nugget:

mlsic=*likfit*(geodata=sic.all, ini.cov.pars=c(100,50), nugget=5,
kappa=1,lambda=0.5, cov.model="mat")

which is then passed on to *krige.control*:

kctrl = *krige.control*(obj.model=mlsic)

The MC-option and the number of simulations, 1000, say, are passed on via octrl = *output.control*(n.pred=1000, simul=T, thres=250)

This option will allow us to produce a map of probabilities for the rainfall exceeding a given threshold of 250 mm, say. After creating a prediction grid with regular spacing of 7.5 km (and using the border lines which are included in data(SIC)),

pgrid= *pred_grid*(sic.borders, by= 7.5)

we pass on all specifications to the conventional krige function:

krige.conv(sic.all, loc=pgrid, borders=sic.borders,
krige=kctrl, out=octrl)

With these specifications, we are quite flexible in evaluating even non-linear functionals of our predictions, just by use of appropriate MC evaluations. We give two examples of such evaluations:

Example 1:

We want to estimate the proportion of the total area (of Switzerland) for which the rainfall exceeded an amount of $150mm$. This can be easily done using the built-in function **apply** and then taking the mean:

```
area150= apply(pred$simul, 2, function(x)
                {sum(x>150)/length(x)})
```

```
mean(area150)
```

Looking at the histogram we can get the whole predictive distribution of $area150$:

```
hist(area150, freq=F)
```

From this distribution we can easily infer confidence intervals a.s.o. (see RScript and figures to follow).

Example 2:

We want to produce a map displaying the probabilities of exceeding a rainfall threshold value of 250mm, i.e. we are interested in getting $P(Z(x) > 250)$ all over Switzerland.
We have already set the corresponding options "**simul=T**" and "**thres=250**" in **output.control** and specified the number of simulations to be done (here: n.pred=1000).

The 1000 realizations of conventional krige predictions are contained in **pred$prob** as part of the output list of the object:

pred=*krige.conv*(sic.all, loc=pgrid, borders=sic.borders,
krige=kctrl, out=octrl)

The map can then be displayed in the usual way:

image(pred,val=1-pred$prob,col=gray(seq(1,0.1,l=25)),
x.leg=c(0,350),y.leg=c(-60,-30))

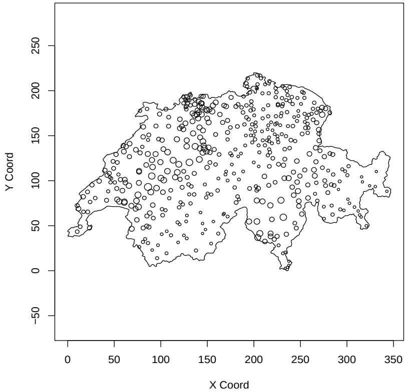


Figure: SIC97: Swiss rainfall data

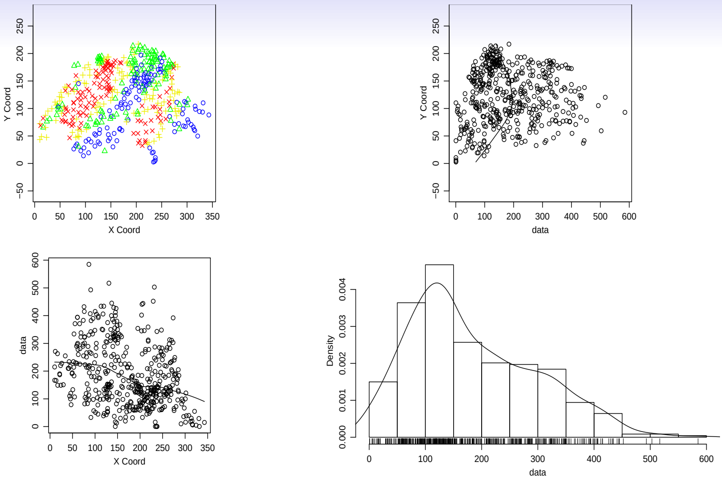


Figure: Exploratory analysis, constancy along the coordinates

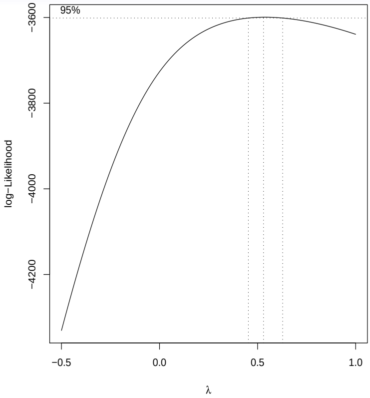


Figure: Profile likelihood of λ assuming independence

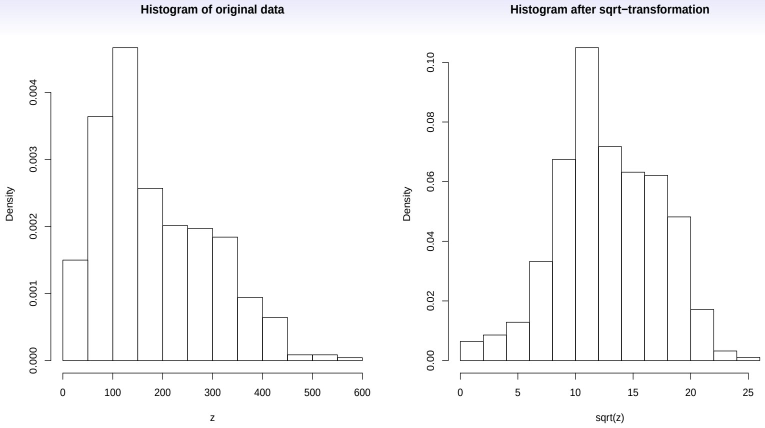


Figure: Histograms before and after transformation

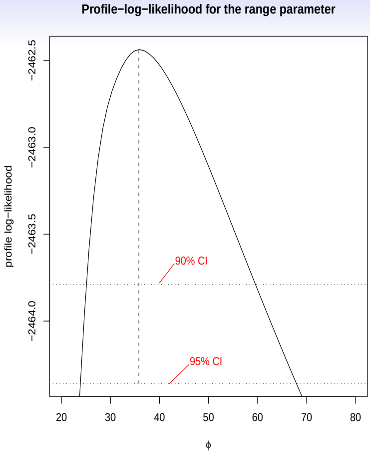


Figure: SIC97: Profile likelihood of range

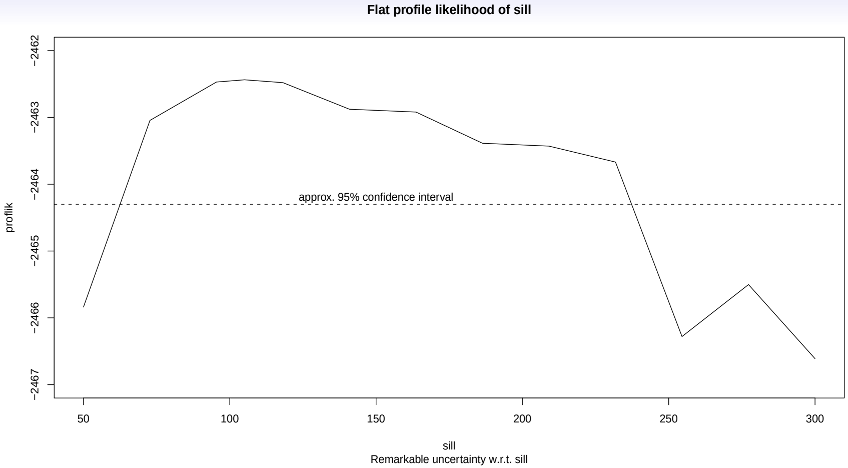


Figure: Profile likelihood of sill

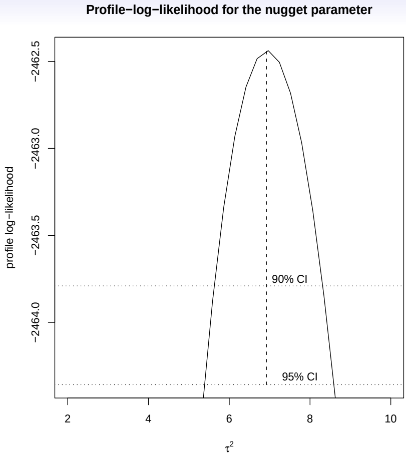


Figure: Profile likelihood of nugget

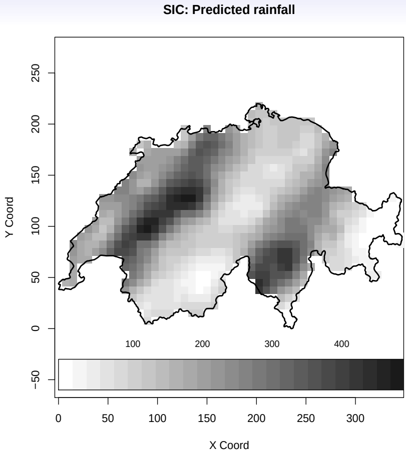


Figure: SIC97: Plug-in predictions

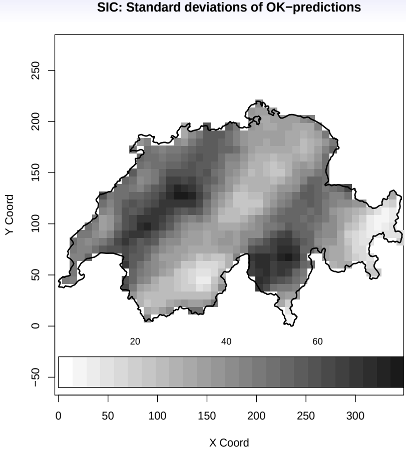


Figure: SIC97: Plug-in standard deviations

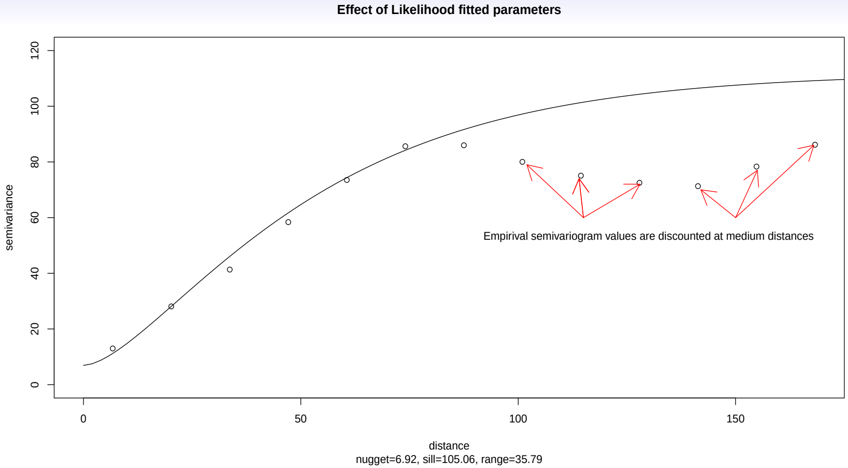


Figure: Matérn variogram with likelihood fitted parameters ($\kappa = 1.0$)

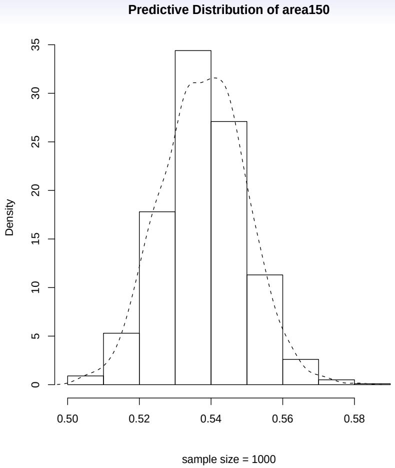
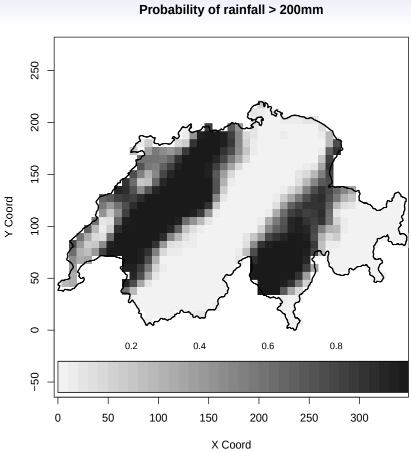


Figure: SIC97: Proportion of area >150



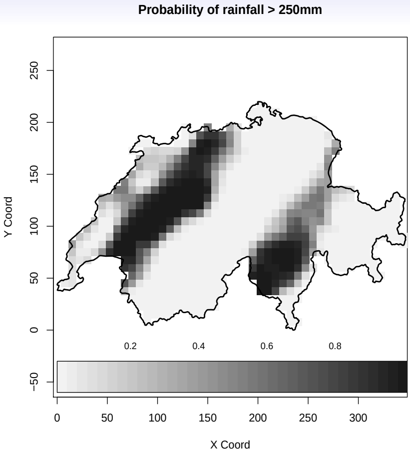


Figure: Exceedance probabilities