Lab 7: Binomial geostatistical models

Peter Diggle & Emanuele Giorgi

Lancaster Medical School, Lancaster University, Lancaster, UK

Model-based geostatistics: geospatial statistical methods for public health applications, 5-9 October 2015

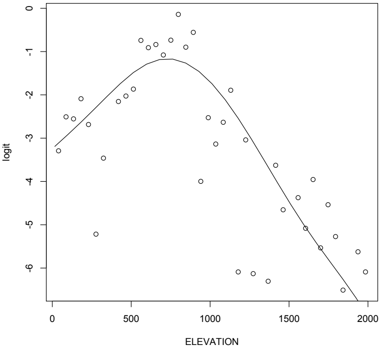
Overview

- Exploratory analysis of prevalence data.
- Geostatistical linear models for logit-transformed prevalence data: parameter estimation.
- Binomial geostatistical models: likelihood-based and Bayesian parameter estimation.

R packages: `PrevMap`, `geoR`, `sm`.

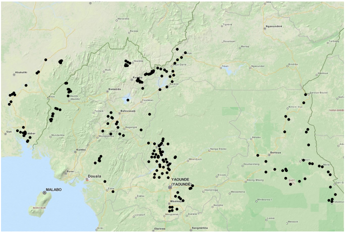
Exploring associations

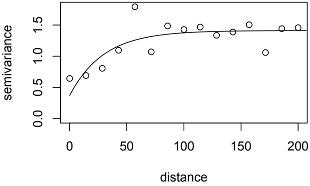
```
> library(PrevMap)
> library(raster)
> data(loaloa)
> elev <- raster("CMR_NGA_alt.tif")
> coords <- SpatialPoints(
+           loaloa[,c("LONGITUDE", "LATITUDE")],
+           CRS("+init=epsg:4236"))
> loaloa$ELEVATION <- extract(elev,coords)
Warning message:
In .local(x, y, ...) : Transforming
SpatialPoints to the CRS of the Raster
>
> library(sm)
> sm.fit <- sm.binomial(x=loaloa$ELEVATION,
+             y=loaloa$NO_INF,
+             N=loaloa$NO_EXAM,h=300)
>
> logit.data <- log((sm.fit$data$y+0.5)/
+                 (sm.fit$data$N-sm.fit$data$y+0.5))
> plot(sm.fit$data$x,logit.data,xlab="ELEVATION", ylab="logit")
> points(sm.fit$eval.points,sm.fit$linear.predictor,type="l")
```



Exploring the presence of spatial correlation

```
> par(mfrow=c(2,1))
> coords.osm <- spTransform(coords,osm())
> library(OpenStreetMap)
> ul <- c(6.904614+1,11.821289-4)
> lr <- c(6.904614-4,11.821289+3.5)
>
> Map <- openmap(ul,lr,type="mapquest")
> plot(Map)
> points(coords.osm,pch=20,cex=0.5)
>
> loaloa$logit <- log((loaloa$NO_INF+0.5)/
+                 (loaloa$NO_EXAM-loaloa$NO_INF+0.5))
>
> lm.fit <- lm(logit ~ ELEVATION+I(ELEVATION^2),
+           data=loaloa)
> coords.utm <- spTransform(coords,
+             CRS("+init=epsg:32632"))
> loaloa$utm_x <- coordinates(coords.utm)[,1]/1000
> loaloa$utm_y <- coordinates(coords.utm)[,2]/1000
> vari <- variog(coords=loaloa[,c("utm_x","utm_y")],
+                data=residuals(lm.fit),
+                uvec=seq(0,200,length=15))
variog: computing omnidirectional variogram
> plot(vari)
> vari.fit <- variofit(vari)
Warning message:
In variofit(vari) :
  initial values not provided - running the default search
> vari.fit
  tausq sigmasq     phi
 0.3711  1.0404 29.8610
> lines(vari.fit)
```





Fitting of geostatistical linear models (1)

```
> library(PrevMap)
> lm.geo.fit <- linear.model.MLE(logit ~ ELEVATION+I(ELEVATION^2),
+               coords=~utm_x+utm_y,
+               data=loaloa,kappa=0.5,
+               start.cov.pars=c(vari.fit$cov.pars[2],
+                                vari.fit$nugget/vari.fit$cov.pars[1]),
+               method="nlminb",messages=FALSE)
>
> summary(lm.fit)

Call:
lm(formula = logit ~ ELEVATION + I(ELEVATION^2), data = loaloa)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1208 -0.5781  0.2495  0.8981  2.3274

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -3.689e+00  1.988e-01  -18.55   <2e-16 ***
ELEVATION       5.962e-03  5.389e-04   11.06   <2e-16 ***
I(ELEVATION^2) -4.065e-06  3.106e-07  -13.09   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.163 on 194 degrees of freedom
Multiple R-squared:  0.4838, Adjusted R-squared:  0.4785
F-statistic:  90.9 on 2 and 194 DF,  p-value: < 2.2e-16
```

Fitting of geostatistical linear models (2)

```
> summary(lm.geo.fit)
Geostatistical linear Gaussian model
Call:
linear.model.MLE(formula = logit ~ ELEVATION + I(ELEVATION^2),
    coords = ~utm_x + utm_y, data = loaloa, kappa = 0.5, start.cov.pars = c(vari.fit$cov.pars[2],
        vari.fit$nugget/vari.fit$cov.pars[1]), method = "nlminb",
    messages = FALSE)

                  Estimate      StdErr z.value  p.value
(Intercept)    -2.3169e+00  7.4326e-01 -3.1173 0.001825 **
ELEVATION       1.3379e-03  1.3971e-03  0.9576 0.338242
I(ELEVATION^2) -1.7268e-06  6.0480e-07 -2.8552 0.004301 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Log-likelihood: -75.99718

Covariance parameters Matern function (kappa=0.5)
             Estimate StdErr
log(sigma^2)  0.62859 0.5235
log(phi)      4.92675 0.6564
log(tau^2)   -0.90729 1.0529

Legend:
sigma^2 = variance of the Gaussian process
phi = scale of the spatial correlation
tau^2 = variance of the nugget effect
```

Exercise

Fitting of geostatistical linear models

Read the data-frame ``LiberiaRemoData.csv''.

1. Explore the relationship between elevation and river-blindness prevalence using the `sm.binomial` function.
2. Fit a linear for the empirical logit of prevalence including elevation as explanatory variables. Fit and plot the empirical variogram based on the residuals of the linear model. Use `vario.fit` to obtain initial values for the covariance parameters.
3. Fit a geostatistical linear model using the `linear.model.MLE` function.

Binomial geostatistical models: the MCML algorithm (1)

```
> par0 <- coef(lm.geo.fit)[-6]
> c.mcmc <- control.mcmc.MCML(n.sim=10000,burnin=2000,thin=8,
+           h=1.65/(nrow(loaloa)^(1/6)))
> bin.geo.fit <- binomial.logistic.MCML(
+                NO_INF~ELEVATION+I(ELEVATION^2),
+                units.m=~NO_EXAM,
+                coords=~utm_x+utm_y,
+                data=loaloa,
+                fixed.rel.nugget=0,
+                control.mcmc=c.mcmc,
+                par0=par0,kappa=0.5,
+                start.cov.pars=par0["phi"],
+                messages=FALSE, plot.correlogram=FALSE)
> bin.geo.fit$log.lik
[1] 8.337682
> par0 <- coef(bin.geo.fit)
> bin.geo.fit <- binomial.logistic.MCML(
+                NO_INF~ELEVATION+I(ELEVATION^2),
+                units.m=~NO_EXAM,
+                coords=~utm_x+utm_y,
+                data=loaloa,
+                fixed.rel.nugget=0,
+                control.mcmc=c.mcmc,
+                par0=par0,kappa=0.5,
+                start.cov.pars=60,
+                messages=FALSE,plot.correlogram=FALSE)
Fixed relative variance of the nugget effect: 0
```

Binomial geostatistical models: the MCML algorithm (2)

```
> summary(bin.geo.fit)
Binomial geostatistical model
Call:
binomial.logistic.MCML(formula = NO_INF ~ ELEVATION + I(ELEVATION^2),
    units.m = ~NO_EXAM, coords = ~utm_x + utm_y, data = loaloa,
    par0 = par0, control.mcmc = c.mcmc, kappa = 0.5, fixed.rel.nugget = 0,
    start.cov.pars = 60, messages = FALSE, plot.correlogram = FALSE)

                  Estimate      StdErr z.value   p.value
(Intercept)    -3.3288e+00  6.0189e-01 -5.5307  3.19e-08 ***
ELEVATION       4.7999e-03  1.3736e-03  3.4944 0.0004752 ***
I(ELEVATION^2) -3.4961e-06  6.6119e-07 -5.2876  1.24e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Objective function: 1.986309

Covariance parameters Matern function
(fixed relative variance tau^2/sigma^2= 0)
             Estimate StdErr
log(sigma^2)   0.2997 0.2722
log(phi)       3.9307 0.3152

Legend:
sigma^2 = variance of the Gaussian process
phi = scale of the spatial correlation
```

Binomial geostatistical models: Bayesian estimation (1)

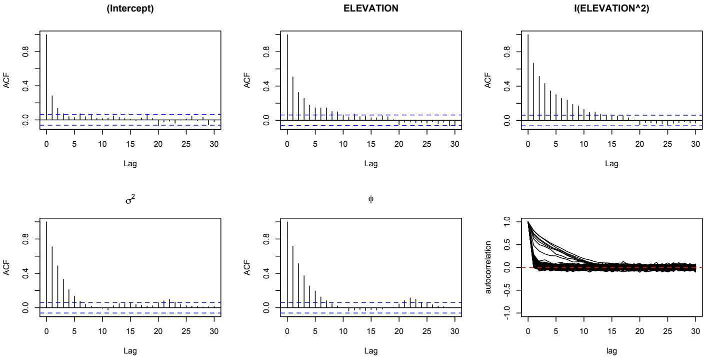
```
> c.prior <- control.prior(beta.mean=rep(0,3),
+                          beta.covar=diag(1000,3),
+                          log.normal.sigma2=c(0.2997, 0.2722),
+                          log.normal.phi=c(3.9307, 0.3152))
> c.mcmc.Bayes <- control.mcmc.Bayes(
+                 n.sim=5000,burnin=1000,thin=1,
+                 h.theta1=0.05,h.theta2=0.05,
+                 L.S.lim=c(5,10),epsilon.S.lim=c(0.05,0.12),
+                 start.S=predict(lm.fit),start.beta=rep(0,3),
+                 start.sigma2=exp(0.2997),
+                 start.phi=exp(3.9307))
> bin.geo.Bayes <-  binomial.logistic.Bayes(
+                NO_INF~ELEVATION+I(ELEVATION^2),
+                units.m=~NO_EXAM,
+                coords=~utm_x+utm_y,
+                data=loaloa,
+                fixed.rel.nugget=0,
+                control.mcmc=c.mcmc.Bayes,
+                control.prior=c.prior,kappa=0.5,
+                start.cov.pars=60)
```

Binomial geostatistical models: Bayesian estimation (2)

```
> summary(bin.geo.Bayes)
Bayesian binomial geostatistical logistic model
Call:
binomial.logistic.Bayes(formula = NO_INF ~ ELEVATION + I(ELEVATION^2),
    units.m = ~NO_EXAM, coords = ~utm_x + utm_y, data = loaloa,
    control.prior = c.prior, control.mcmc = c.mcmc.Bayes, kappa = 0.5)

                         Mean        Median          Mode        StdErr      HPD 0.025
(Intercept)    -3.227648e+00 -3.251519e+00 -3.259715e+00 5.170325e-01 -4.155766e+00
ELEVATION       4.701623e-03  4.735153e-03  4.751075e-03 1.256302e-03  1.900925e-03
I(ELEVATION^2) -3.571101e-06 -3.609107e-06 -3.672173e-06 6.916466e-07 -4.872764e-06
                    HPD 0.975
(Intercept)    -2.121649e+00
ELEVATION       6.815630e-03
I(ELEVATION^2) -2.154335e-06

Covariance parameters Matern function (kappa=0.5)

             Mean    Median      Mode     StdErr  HPD 0.025 HPD 0.975
sigma^2  1.370408  1.353765  1.393572  0.2328517  0.9627482  1.855822
phi     47.385426 45.761661 43.424978 10.1137336 31.5288220 69.958095

Legend:
sigma^2 = variance of the Gaussian process
phi = scale of the spatial correlation
```

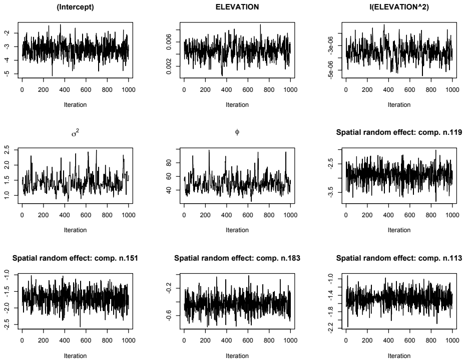
Autocorrelation plots

```
> par(mfrow=c(2,3))
> autocor.plot(bin.geo.Bayes,"beta",component.beta=1)
> autocor.plot(bin.geo.Bayes,"beta",component.beta=2)
> autocor.plot(bin.geo.Bayes,"beta",component.beta=3)
> autocor.plot(bin.geo.Bayes,"sigma2")
> autocor.plot(bin.geo.Bayes,"phi")
> autocor.plot(bin.geo.Bayes,"S",component.S="all")
```



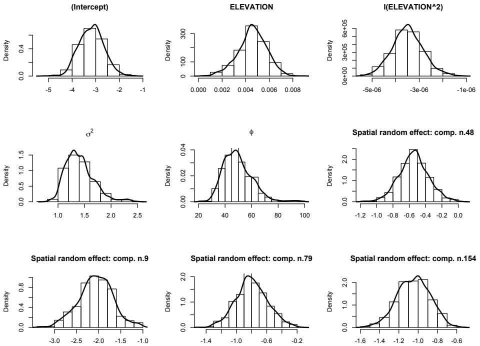
Trace-plots

```
par(mfrow=c(3,3))
trace.plot(bin.geo.Bayes,"beta",component.beta=1)
trace.plot(bin.geo.Bayes,"beta",component.beta=2)
trace.plot(bin.geo.Bayes,"beta",component.beta=3)
trace.plot(bin.geo.Bayes,"sigma2")
trace.plot(bin.geo.Bayes,"phi")
trace.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
trace.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
trace.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
trace.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
```



Posterior density plots

```
> par(mfrow=c(3,3))
> dens.plot(bin.geo.Bayes,"beta",component.beta=1)
> dens.plot(bin.geo.Bayes,"beta",component.beta=2)
> dens.plot(bin.geo.Bayes,"beta",component.beta=3)
> dens.plot(bin.geo.Bayes,"sigma2")
> dens.plot(bin.geo.Bayes,"phi")
> dens.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
> dens.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
> dens.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
> dens.plot(bin.geo.Bayes,"S",component.S=sample(1:nrow(loaloa),1))
```



Exercise

Fitting of binomial geostatistical models

Read the data-frame ``LiberiaRemoData.csv''.

1. Fit a geostatistical binomial model using the MCML algorithm.
2. Fit a geostatistical binomial model using Bayesian methods. Use diagnostics plots to check the mixing and convergence of the MCMC algorithm.