University of California, Los Angeles
Department of Statistics

Statistics C173/C273 **Instructor: Nicolas Christou**

Fitting a model variogram to the sample variogram

The parameters of the model variogram can be estimated through ordinary or weighted least squares (and the method of maximum likelihood which is not discussed here):

a. Method of ordinary least squares (ols):
The parameters of the model variogram are estimated by minimizing the following function:

$$\min \sum_h \left(2\hat{\gamma}(h) - 2\gamma(h)\right)^2$$

b . Method of weighted least squares:

1. The weights are the number of pairs for each lag distance. The parameters of the model variogram are estimated by minimizing the following function:

$$\min \sum_h N(h) \left(2\hat{\gamma}(h) - 2\gamma(h)\right)^2$$

2. Cressie's weighted least squares. Here the weights are given by $\frac{N(h)}{\gamma(h)^2}$. The parameters of the model variogram are estimated by minimizing the following function:

$$\min \sum_h N(h) \left(\frac{\hat{\gamma}(h)}{\gamma(h)} - 1\right)^2$$

The `geoR` library uses the `variofit` function for the estimation of the parameters of the model variogram as follows:

```
> b <- as.geodata(a)
> variogram1 <- variog(b)
> fit <- variofit(variogram1, cov.model="name of model variogram ",
         ini.cov.pars=c(partial sill, range),
         fix.nugget=TRUE, nugget=value or
         fix.nugget=FALSE, nugget=value (default is zero),
         max.dist=value (default is the maximum distance from
         the variogram calculation: variogram1$max.dist),
         wei="equal", "npairs", "cressie" (default is "npairs"))
```

For the initial parameters we can give a range of values for the partial sill and the range. The `variofit` function will select the values of the partial sill and the range that give the smallest value of the summation and use them as initial values for the minimization. For example:

```
> initial.values <- expand.grid(seq(10, 100, by=10), seq(200, 250, by=10))
```

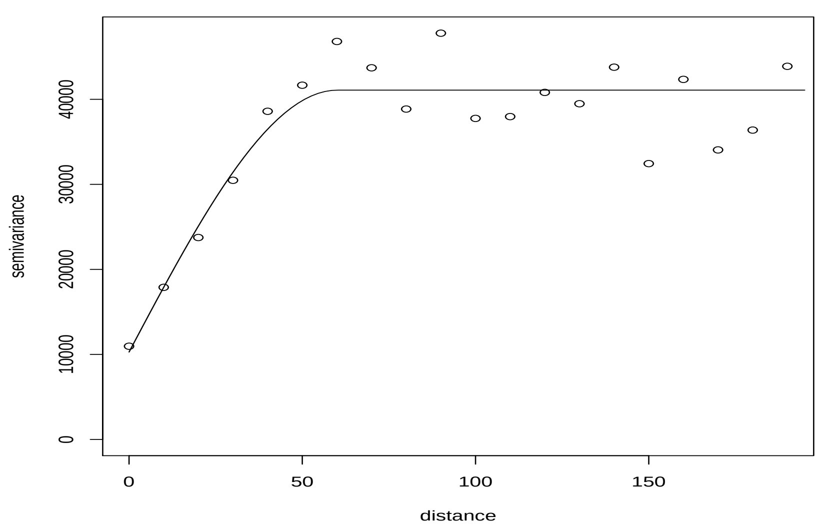
And then inside the varofit function we have: ini.cov.pars=initial.values. We can now plot the sample variogram and fit the model variogram to it:

```
> plot(variogram1)
> lines(fit)
```

Here is an example using the Wolfcamp aquifer data:

```
> site="http://www.stat.ucla.edu/~nchristo/statistics_c173_c273/
  wolfcamp.txt"
> a<-read.table(file=site, header=T)
> b <- as.geodata(a)
> variogram1 <- variog(b, trend="1st", uvec=seq(0, 190, by=10))
variog: computing omnidirectional variogram
> initial.values <- expand.grid(seq(25000, 35000, by=1000),
  seq(50, 80, by=10))
> fit <- variofit(variogram1, cov.model="sph", ini.cov.pars=initial.values,
fix.nugget=FALSE, nugget=10000, wei="cressie")
variofit: weights used: cressie
variofit: minimisation function used: optim
variofit: searching for best initial value ... selected values:
              sigmasq phi   tausq   kappa
initial.value "31000" "60"  "10000" "0.5"
status        "est"   "est" "est"   "fix"
loss value: 30.9766544355334
> plot(variogram1)
> lines(fit)
```

And here is the sample and model variograms:



Another example:

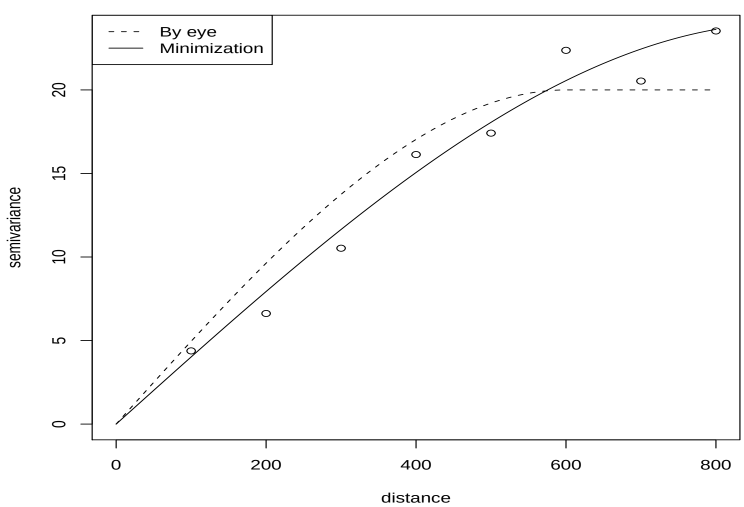
Fitting a model variogram to the sample variogram:

- By eye.

- Using weighted least squares.

Here are the R commands:

```
> library(geoR)
> a <- read.table("http://www.stat.ucla.edu/~nchristo/data_var.txt",
  header=TRUE)
> b <- as.geodata(a)
> var1 <- variog(b, max.dist=800, uvec=seq(0, 800, by=100))
> plot(var1)
> lines.variomodel(cov.model="sph", cov.pars=c(20,600), nug=0, max.dist=800, lty=2)
> fit <- variofit(var1, cov.model="sph", ini.cov.pars=c(20,600),
  fix.nugget=TRUE, nugget=0)
> lines(fit, lty=1)
> legend("topleft", legend=c("By eye", "Minimization"), lty=c(2,1))
> fit
variofit: model parameters estimated by WLS (weighted least squares):
covariance model is: spherical
fixed value for tausq =  0
parameter estimates:
 sigmasq      phi
 23.9955 892.1655
Practical Range with cor=0.05 for asymptotic range: 892.1655
variofit: minimised weighted sum of squares = 1473.104
```



More examples:
Below you can see an example with the three methods for fitting a model variogram (spherical) to the sample variogram:

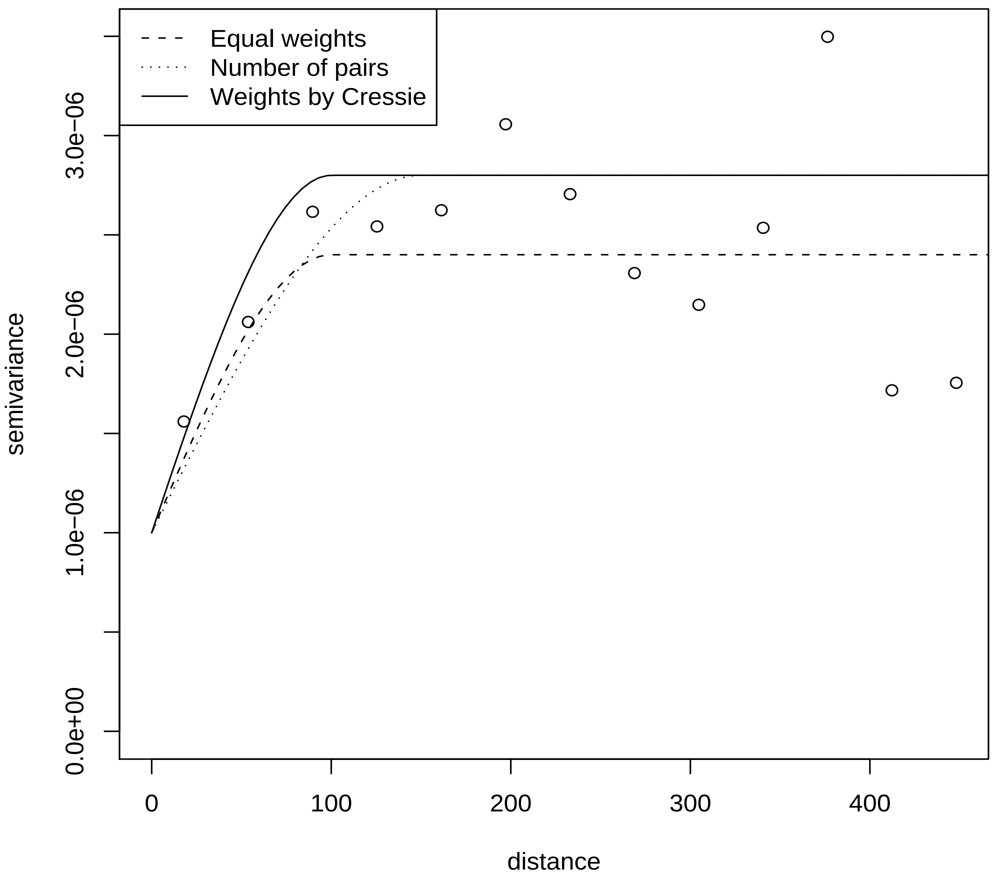
- Equal weights.
- Weights=number of pairs.
- Using Cressie's weighted least squares.

Here are the R commands:

```
> library(geoR)
> a <- read.table("http://www.stat.ucla.edu/~nchristo/statistics_c173_c273/
  nc_sids.txt", header=TRUE)
> a1 <- cbind(a$x, a$y, a$sids/a$birth)
> b <- as.geodata(a1)
> var1 <- variog(b)
> plot(var1)

> initial.values <- expand.grid(seq(0.000001, 0.000003, by=0.0000004),
  seq(100, 350, by=50))
> fit1 <- variofit(var1, cov.model="sph", ini.cov.pars=initial.values,
  fix.nugget=FALSE, nugget=0.000001, wei="equal")

> fit2 <- variofit(var1, cov.model="sph", ini.cov.pars=initial.values,
  fix.nugget=FALSE, nugget=0.000001, wei="npairs")

> fit3 <- variofit(var1, cov.model="sph", ini.cov.pars=initial.values,
  fix.nugget=FALSE, nugget=0.000001, wei="cressie")

#  Below we see the results (the estimated parameters with the three methods):

> fit1$cov.pars
[1] 1.4e-06 1.0e+02
> fit1$nugget
[1] 1e-06
> fit2$cov.pars
[1] 1.8e-06 1.5e+02
> fit2$nugget
[1] 1e-06
> fit3$cov.pars
[1] 1.8e-06 1.0e+02
> fit3$nugget
[1] 1e-06
```

Here is the plot:

```
> plot(var1)
> lines(fit1, type="l", lty=2)
> lines(fit2, type="l", lty=3)
> lines(fit3, type="l")
> legend("topleft", legend=c("Equal weights", "Number of pairs",
  "Weights by Cressie"), lty=c(2,3,1))
```



Exercise:
Use the data from the *Maas* river

```
a <- read.table("http://www.stat.ucla.edu/~nchristo/statistics_c173_c273/soil.txt",
  header=TRUE)
```

to answer the following questions:

a. Compute the sample variogram. Try different directions and angular tolerance until you find something reasonable.

b. Fit theoretical variogram models using `equal, npairs, cressie` weights to the sample variogram and place all the fitted variograms together with the sample variogram on one plot (see example on page 5). For the `ini.cov.pars` argument in the `variofit` function use a grid as shown on page 1.

c. Print the estimates of the parameters for each variogram model and for each type of weights.

d. Print the two plots.