Homework3

Laha Ale

February 20, 2019

Note: This file is produced by RMarkdown , and the lines start with ## are the outputs of R codes.

```
library(geoR)
library(spBayes)
```

Excerise 6

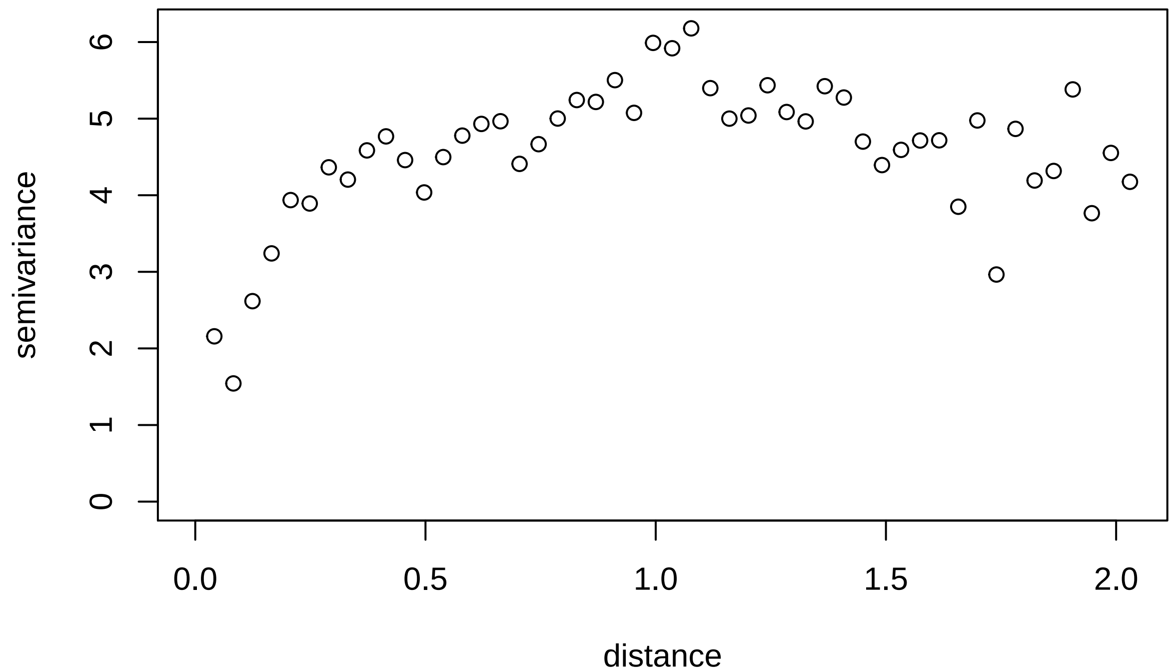
step 1:load data and plot variogram

```
url <- "https://www.counterpointstat.com/uploads/1/1/9/3/119383887/myscallops.txt"
myscallops <- read.table(url,header = T)
coords <- as.matrix(myscallops[,c("lat","long")])
lgcatch<- myscallops$lgcatch

bins = 50
max.dist <- 0.7*max(iDist(coords))
myscps.vario <- variog(coords = coords, data = lgcatch,
                       uvec = (seq(0, max.dist, length = bins)))
```

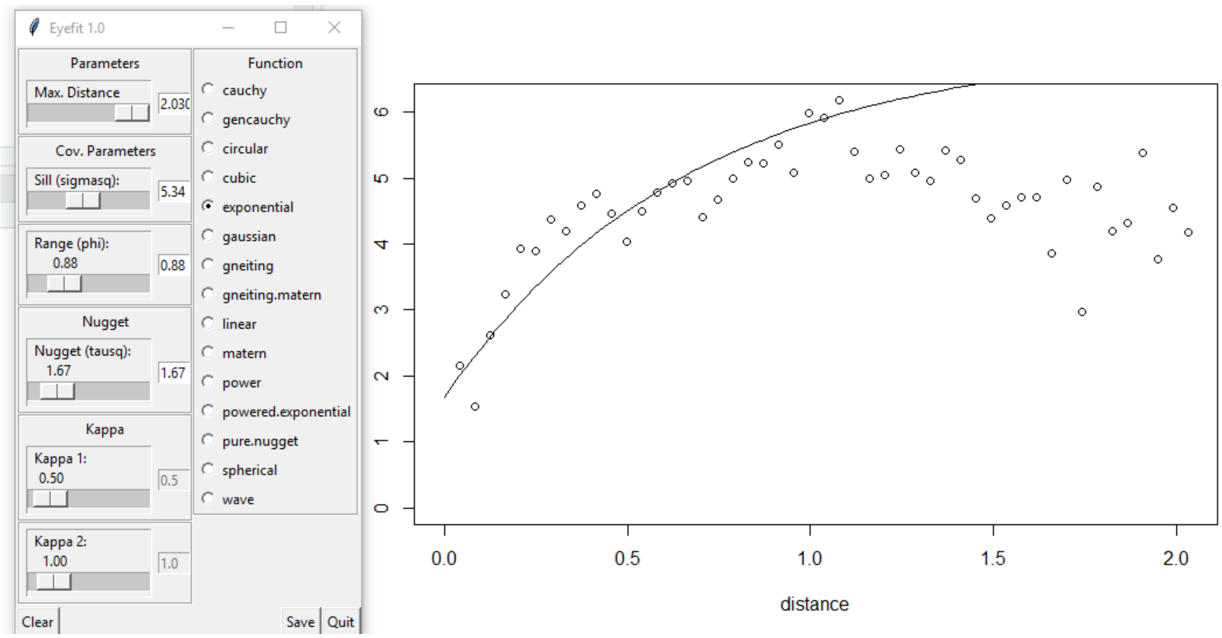
```
## variog: computing omnidirectional variogram
```

```
plot(myscps.vario)
```



step 2: adjust paramters with eyefit

```
#adjust paramters with eyefit
eyefit(myscps.vario,silent=TRUE)
```



step 3: computing varfit

According step 2 results, the following parameters should setting as $\sigma^2 = 5.34, \Phi = 0.88$ and nugget=1.67. And *exponential* model seems works fine.

```
fit.lgcatch<- variofit(myscps.vario,cov.model="exponential",
                       fix.nugget=FALSE, ini.cov.pars=c(5.34,0.88),
                       nugget=1.67)
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```
fit.lgcatch
```

```
## variofit: model parameters estimated by WLS (weighted least squares):
## covariance model is: exponential
## parameter estimates:
##   tausq sigmasq    phi
##  0.3802  4.6326  0.1767
## Practical Range with cor=0.05 for asymptotic range: 0.5292095
##
## variofit: minimised weighted sum of squares = 2649.161
```

step 4: Kriging

According step 3 results, the following parameters should setting as $\sigma^2 = 4.6326, \Phi = 0.1767$ and nugget=0.3802. Although the value $\Phi = 0.1767$ is not making sense in the above results if we compare to results from step 2, we will use this value to compute the last point of the lgcatch data.

```
point<-krige.conv(coords = coords, data = lgcatch,loc=c(length(lgcatch),1),
                  krige=krige.control(cov.pars=c(4.6326,0.1767),
                                      cov.model="exponential",
                                      nugget=0.3802))
```

```
## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood
```

```
point
```

```
## $predict
##     data
## 2.644499
##
## $krige.var
## [1] 5.263914
##
## $beta.est
##     beta
## 2.644499
##
## $distribution
## [1] "normal"
##
## $message
## [1] "krige.conv: Kriging performed using global neighbourhood"
##
## $call
## krige.conv(coords = coords, data = lgcatch, locations = c(length(lgcatch),
##     1), krige = krige.control(cov.pars = c(4.6326, 0.1767), cov.model = "exponential",
##     nugget = 0.3802))
##
## attr(,"sp.dim")
## [1] "2d"
## attr(,"prediction.locations")
## c(length(lgcatch), 1)
## attr(,"parent.env")
## <environment: R_GlobalEnv>
## attr(,"data.locations")
## coords
## attr(,"class")
## [1] "kriging"
```

```
pred_low <-point$predict - 2*sqrt(point$krige.var)
pred_high <-point$predict + 2*sqrt(point$krige.var)
print(paste("The PI is between",pred_low,"and",pred_high))
```

```
## [1] "The PI is between -1.94414491080096 and 7.23314348012374"
```

step 5: Summary Results

As we can see from above, the predict and varirance are 2.64 and 5.26; therefore, the PI with 95% confident is beteewn $2.64 - 2 \times \sqrt{5.26} = -1.94$ and $2.64 + 2 \times \sqrt{5.26} = 7.23$, more accurate numbers have been printed above.

Excerise 7

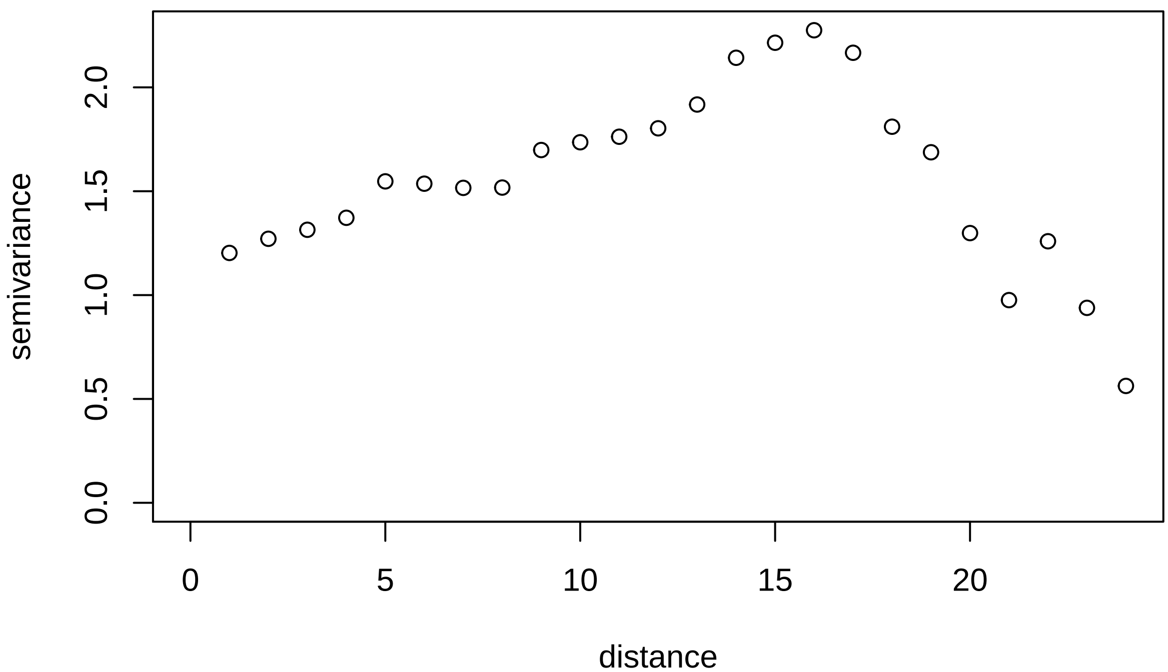
step 1:load data and plot variogram

```
url_coal <- "https://www.counterpointstat.com/uploads/1/1/9/3/119383887/coal.ash.txt"
coalash <- read.table(url_coal,header = T)


coords_coal <- as.matrix(coalash[,c("x","y")])
coal <- coalash$coal
vario.coal <- variog(coords = coords_coal, data = coal, uvec = (seq(0, length = bins)))
```

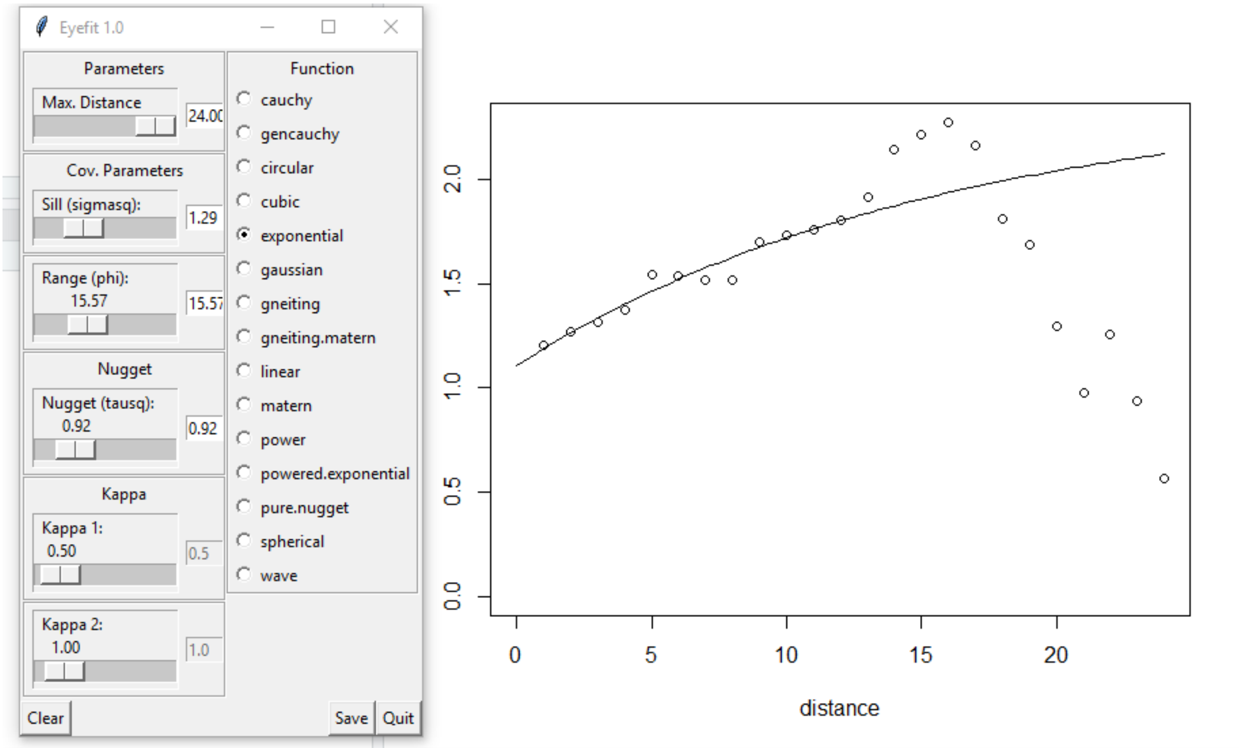
```
## variog: computing omnidirectional variogram
```

```
plot(vario.coal)
```



step 2: adjust paramters with eyefit

```
eyefit(vario.coal ,silent=TRUE)
```



step 3: computing varfit

According step 2 results, the following parameters should setting as $\sigma^2 = 1.29, \Phi = 15.57$ and nugget=0.92. However, the value shown in eyefit results is a little strange, it may more make sense if we choose about $\frac{1}{17}$ base on variogram. In the following code, let's try both values of Φ and select the one produces more accurate results in predicting τ^2 and σ^2.

And *exponential* model seems works fine.

```
fit.coal<- variofit(vario.coal,cov.model="exponential",
                    fix.nugget=FALSE,
                    ini.cov.pars=c(1.29,1/17), nugget=0.92)
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```
fit.coal
```

```
## variofit: model parameters estimated by WLS (weighted least squares):
## covariance model is: exponential
## parameter estimates:
##   tausq sigmasq     phi
##  0.6712  0.9437  2.0996
## Practical Range with cor=0.05 for asymptotic range: 6.289926
##
## variofit: minimised weighted sum of squares = 1183.811
```

```r
fit.coal<- variofit(vario.coal,cov.model="exponential",
                    fix.nugget=FALSE,
                    ini.cov.pars=c(1.29,15.57), nugget=0.92)
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```r
fit.coal
```

```
## variofit: model parameters estimated by WLS (weighted least squares):
## covariance model is: exponential
## parameter estimates:
##   tausq sigmasq     phi
##  1.0372  1.2026 11.1435
## Practical Range with cor=0.05 for asymptotic range: 33.38281
##
## variofit: minimised weighted sum of squares = 498.1197
```

step 4: Kriging Prediction

According step 3 results, Even though set $\phi = \frac{1}{17}$ is make more sense but the model can produce more accurate values of τ^2 and σ^2 by setting $\phi = 15.57$. Therefore, we will adopt the results from the second option. And the following parameters should setting as $\sigma^2 = 1.2026, \Phi = 11.1435$ and $\tau^2 = 1.0372$.

```r
point<-krige.conv(coords = coords_coal, data = coal,loc=c(length(coal),1),
                  krige=krige.control(cov.pars=c(1.2026,11.1435),cov.model="exponential",nugget=1.0372))
```

```
## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood
```

```r
point
```

```
## $predict
##     data
## 9.663041
##
## $krige.var
## [1] 2.761329
##
## $beta.est
##     beta
## 9.663041
##
## $distribution
## [1] "normal"
##
## $message
## [1] "krige.conv: Kriging performed using global neighbourhood"
##
## $call
## krige.conv(coords = coords_coal, data = coal, locations = c(length(coal),
##     1), krige = krige.control(cov.pars = c(1.2026, 11.1435),
##     cov.model = "exponential", nugget = 1.0372))
##
## attr(,"sp.dim")
## [1] "2d"
```

```
## attr(,"prediction.locations")
## c(length(coal), 1)
## attr(,"parent.env")
## <environment: R_GlobalEnv>
## attr(,"data.locations")
## coords_coal
## attr(,"class")
## [1] "kriging"
```

```
pred_low <-point$predict - 2*sqrt(point$krige.var)
pred_high <-point$predict + 2*sqrt(point$krige.var)
print(paste("The PI is between",pred_low,"and",pred_high))
```

```
## [1] "The PI is between 6.33959120662333 and 12.9864901385506"
```

step 5: Summary Results

As we can see from above, the predict and varirance are 9.66 and 2.76; therefore, the PI with 95% confident is beteewn $9.66 - 2 \times \sqrt{2.76} = 6.34$ and $9.66 + 2 \times \sqrt{2.76} = 12.98$, more accurate numbers have been printed above.

Using Matrix Method

Exercise 6

Given same previous work and choose paramters as below:

```
# assign the coords to given_coords except last row

given_coords <- coords[1:dim(coords)[1]-1,]
myscallops.covmat<-varcov.spatial(coords = given_coords,
                                  cov.model = "exponential",
                                 nugget = 1.67,
                                 cov.pars = c(5.34,0.88))
# just print myscallops.covmat will be too long!
# myscallops.covmat
# setup covariance matrix with point for prediction
gamma_all<-varcov.spatial(coords = coords,
                          cov.model = "exponential",
                          nugget = 1.67,
                          cov.pars = c(5.34,0.88))
# we are interested in last column
gamma <- gamma_all$varcov[,ncol(gamma_all$varcov)]
cov <- gamma[length(gamma)]
gamma <- gamma[-length(gamma)]

z <- lgcatch[-length(lgcatch)]
mu <- mean(z)
m <- rep(mu, length(z))
m <- matrix(m,nrow=length(z),ncol=1)
z <- matrix(z,nrow=length(z),ncol=1)


y_pred<- mu +t(gamma)%*%solve(myscallops.covmat$varcov)%*%(z-m)
y_pred
```

```
##           [,1]
## [1,] 2.960819
```

```
# now compute variance

cov
```

```
## [1] 7.01
```

```
var_pred<-cov -t(gamma)%*%solve(myscallops.covmat$varcov)%*%gamma
var_pred
```

```
##           [,1]
## [1,] 2.369833
```

```
pred_low <- y_pred - 2*sqrt(var_pred)
pred_high <- y_pred + 2*sqrt(var_pred)
print(paste("The PI is between",pred_low,"and",pred_high))
```

```
## [1] "The PI is between -0.11803299551682 and 6.03967116464465"
```

As we can see, from above results, the matrix method results is almost the same as the API function results.

Exercise 7

Given same previous work and choose paramters as below:

```
# assign the coords to given_coords except last row

given_coords <- coords_coal[1:dim(coords_coal)[1]-1,]
coal.covmat<-varcov.spatial(coords = given_coords,
                                  cov.model = "exponential",
                                 nugget = 1.0372,
                                 cov.pars = c(1.2026,11.1435))
# just print myscallops.covmat will be too long!
# myscallops.covmat
# setup covariance matrix with point for prediction
gamma_all<-varcov.spatial(coords = coords_coal,
                           cov.model = "exponential",
                           nugget = 1.0372,
                           cov.pars = c(1.2026,11.1435))
# we are interested in last column
gamma <- gamma_all$varcov[,ncol(gamma_all$varcov)]
cov <- gamma[length(gamma)]
gamma <- gamma[-length(gamma)]

z <- coal[-length(coal)]
mu <- mean(z)
m <- rep(mu, length(z))
m <- matrix(m,nrow=length(z),ncol=1)
z <- matrix(z,nrow=length(z),ncol=1)


y_pred<- mu +t(gamma)%*%solve(coal.covmat$varcov)%*%(z-m)
y_pred
```

```
##           [,1]
## [1,] 8.612182
```

```r
# now compute variance

cov
```

```
## [1] 2.2398
```

```r
var_pred<-cov -t(gamma)%*%solve(coal.covmat$varcov)%*%gamma
var_pred
```

```
##           [,1]
## [1,] 1.391571
```

```r
pred_low <- y_pred - 2*sqrt(var_pred)
pred_high <- y_pred + 2*sqrt(var_pred)
print(paste("The PI is between",pred_low,"and",pred_high))
```

```
## [1] "The PI is between 6.25288475730364 and 10.971479547674"
```

As we can see, from above results, the matrix method results is clost to the API function results to some extend.