

# Spatial Experiments Doc 3

Antony Sikorski

Checking the ability of fields to calculate Maximum Likelihood Estimates

A bit of setup:

```
suppressMessages(library( fields))

#Test 1 synthetic data
matrixRoot<- function( A, inv=TRUE){
  hold<- eigen(A, symmetric=TRUE)
  if( inv){
    DHalf<- sqrt(1/hold$values)
  }
  else{
    DHalf<- sqrt(hold$values)
  }
  hold$vectors%*%diag(DHalf)%*%t( hold$vectors)
}
set.seed(224)
aRange<- 3.5
M<- 100
s<- cbind( 1:M)
scaledD<- rdist( s,s)/aRange
A<- (1+ scaledD)* exp( - scaledD)
B<- matrixRoot(A, inv=FALSE)
f<- B%*%rnorm( M)
f6<- list(s=s, y=f)
```

We first calculate the likelihood by hand using the MLE values of the aRange,  $\sigma^2$ , and  $\tau^2$  from the spatial process function:

```

#performing the fit
Z<- f6$y
s<- f6$s
fit0<- spatialProcess( s,Z,smoothness = 1.5,mKrig.args = list( m=0)
)
lLike0<- fit0$summary["lnProfileLike.FULL"]

#grabbing the params
aRange <- fit0$summary["aRange"]
sigma2 <- fit0$summary["sigma2"]
tau2 <- (fit0$summary["tau"])^2
lambda <- tau2/sigma2
n <- M

#calculations by hand
H <- (1 + rdist(s,s)/aRange)*exp(-rdist(s,s)/aRange)
CovM <- (sigma2 * H) + diag(tau2, M)

lnLike <- (-n/2)*log(2*pi) - (1/2)*log(det(CovM)) - (1/2)*(t(Z))%*%solve(CovM)%*%Z
lnLike <- lnLike[1,1]

#checking validity
lnLike

```

```
## [1] 25.61477
```

```
test.for.zero(lnLike, lLike0)
```

```
## PASSED test at tolerance 1e-08
```

We now reduce the likelihood function to only be in terms of  $\lambda$  and aRange (maximizing over  $\sigma^2$ ), and calculate the likelihood value again:

```

RSS <- t(Z)%*%solve(H + diag(lambda, n))%*%Z
sigma2Est <- RSS/n

lnLikeSig <- (-n/2)*log(2*pi) - (n/2)*log(sigma2Est) - (1/2)*log(det(H + diag(lambda, n))) - (n/2)
lnLikeSig <- lnLikeSig[1,1]
lnLikeSig

```

```
## [1] 25.61477
```

```
test.for.zero(lnLikeSig, lLike0)
```

```
## PASSED test at tolerance 1e-08
```

We now add some white noise to the data and compute the log likelihood “by hand”.

```
set.seed(431)
tau <- .05
e <- tau*rnorm(100)
Z1 <- Z + e
fit1 <- spatialProcess( s,Z1,smoothness = 1.5,mKrig.args = list( m=0))
lLike1 <- fit1$summary["lnProfileLike.FULL"]

aRange <- fit1$summary["aRange"]
sigma2 <- fit1$summary["sigma2"]
tau2 <- (fit1$summary["tau"])^2
lambda <- tau2/sigma2
n <- M

H <- (1 + rdist(s,s)/aRange)*exp(-rdist(s,s)/aRange)
CovM <- (sigma2 * H) + diag(tau2, M)

lnLike1 <- (-n/2)*log(2*pi) - (1/2)*log(det(CovM)) - (1/2)*(t(Z1))%*%solve(CovM)%*%Z1
lnLike1 <- lnLike1[1,1]

lnLike1
```

```
## [1] 11.60587
```

```
test.for.zero(lnLike1, lLike1)
```

```
## PASSED test at tolerance 1e-08
```

We expect the log likelihood to be smaller and not match the fields function if we used the true values of the covariance parameters because the MLE estimates maximize the log likelihood function, so using any other parameters will result in a smaller log likelihood value.

We now add a linear regression component into our spatial model, and perform the calculations again.

```

fit2<- spatialProcess( s,Z1,smoothness = 1.5,gridN=25)
#to check if I did this correctly
lLike2<- fit2$summary["lnProfileLike.FULL"]

aRange <- fit2$summary["aRange"]
sigma2 <- fit2$summary["sigma2"]
tau2 <- (fit2$summary["tau"])^2
lambda <- tau2/sigma2
n <- M

H <- (1 + rdist(s,s)/aRange)*exp(-rdist(s,s)/aRange)

#covariances get smaller
CovM <- (sigma2 * H) + diag(tau2, M)

X <- cbind(1,s)
bHat <- (solve(t(X) %*% solve(CovM) %*% X)) %*% t(X) %*% solve(CovM) %*% Z1

#solving for our log Likelihood
lnLike2 <- (-n/2)*log(2*pi) - (1/2)*log(det(CovM)) - (1/2)*(t(Z1 - X%*%bHat))%*%solve(CovM)%*%(Z1 - X%*%bHat)
lnLike2 <- lnLike2[1,1]

lnLike2

```

```
## [1] 13.56994
```

```

#it appears I did it right
test.for.zero(lnLike2, lnLike2)

```

```
## PASSED test at tolerance 1e-08
```

Let's see if the difference made by the linear component is significant by comparing it to the given Chi square (.95 quantile of  $\chi^2(df = 2)$ ) value.

```

diff <- lnLike2 - lnLike1
chisqVal <- 0.5 * qchisq(0.95,2)

cat("The difference value is ", diff, " and the Chisq Value is ", chisqVal)

```

```
## The difference value is 1.964068 and the Chisq Value is 2.995732
```

It appears that the difference made by the linear component is not statistically significant.

## Creating a MLE Log Likelihood Contour Plot for aRange and Lambda values

Setup:

```

data(ozone2)
s<- ozone2$lon.lat
# day 16
Z<- ozone2$y[16,]
good<- !is.na( Z)
s<- s[good,]
Z<- Z[good]
fitOzone<- spatialProcess( s,Z,smoothness=.5)

gridN <- 80

lam1 <- seq(.001, 1, length.out = gridN)
a1 <- seq(.25, 4, length.out = gridN)
surfaceGrid <- matrix(NA, nrow = gridN, ncol = gridN)

#estimating MLE over variety of aRanges and Lambdas
for (i in 1:gridN){
  for (j in 1:gridN){
    fit<- spatialProcess( s,Z,smoothness=.5,lambda=lam1[i], aRange=a1[j])
    surfaceGrid[i,j] <- fit$summary["lnProfileLike.FULL"]
  }
}

```

Here we plot a log likelihood surface as an image plot with aRange and  $\lambda$  on our axis. We indicate the maximum value and it's location, and we also add a 75% confidence as a contour line for easy visualization.

```

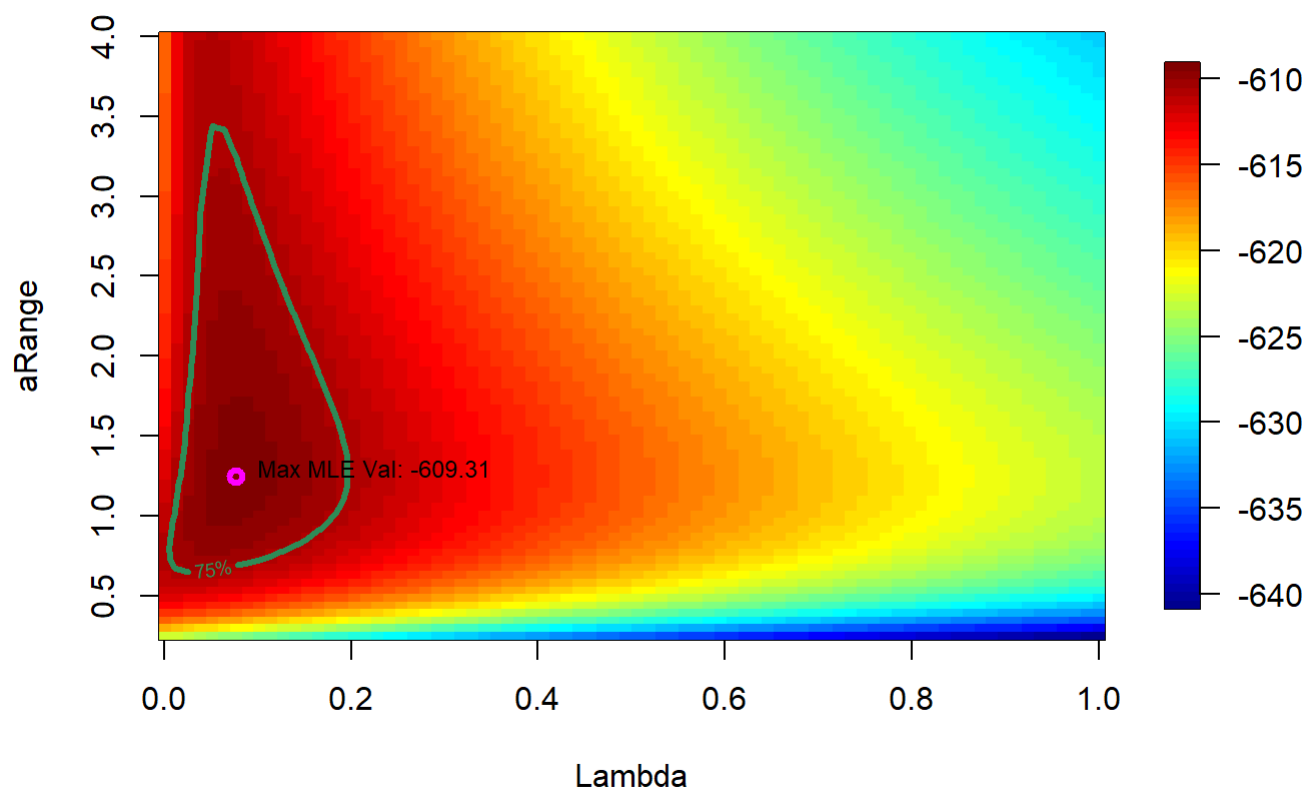
#Part 1 (plotting)
imagePlot(lam1, a1, surfaceGrid, xlab = "Lambda", ylab = "aRange", main = "MLE Log Likelihood Co
ntour Plot")
maxX <- which.max.matrix(surfaceGrid)[1]
maxY <- which.max.matrix(surfaceGrid)[2]
maxVal <- max(surfaceGrid)

#Part 3 (contour line)
contour(lam1, a1, surfaceGrid, level = maxVal - (0.5*qchisq(.75,2)), labels = "75%", col = "seag
reen", add = TRUE, lwd = 3, lex = 3)

#Part 2 is below part 3 so that the text is placed above the contour line
#Part 2 (indicating maximum value)
points(lam1[maxX], a1[maxY], col = "magenta", lwd = 3)
text(lam1[maxX], a1[maxY], adj = c(0,0), paste0("    Max MLE Val: ", round(maxVal, 2)), cex = 0.7
5)

```

## MLE Log Likelihood Contour Plot



## Creating a parametric bootstrap to set a confidence set for the ozone data above.

We first do the 200 simulations and capture the bootstrap replicates.

```
nSims <- 200

#empty lists for vals
lambdaVals <- c()
aRangeVals <- c()
MLEVals <- c()
TrueMLEVals <- c()

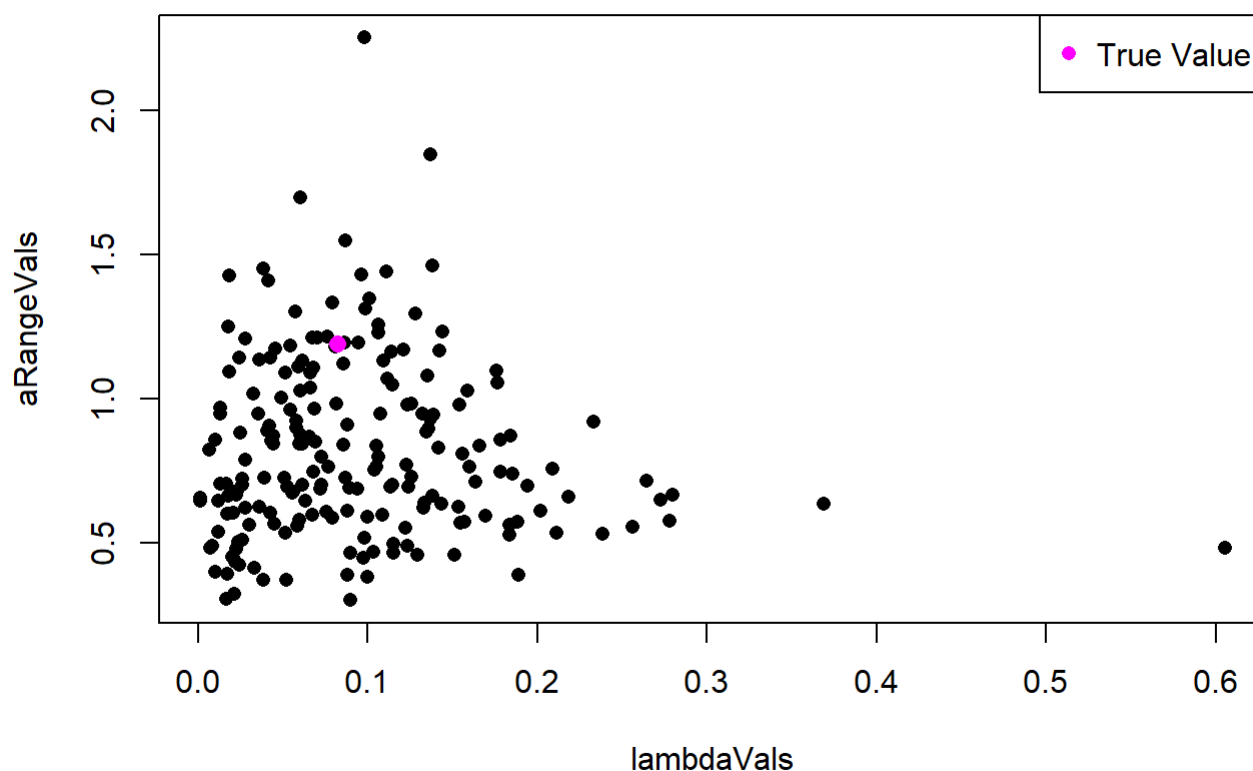
#200 simulations
set.seed(532)
Zsim <- simSpatialData(fitOzone, M = 200)

#for loop to capture vals from the simulations, along with the true vals for 3(d)
for (i in 1: nSims){
  tmp <- spatialProcess(s, Zsim[,i], smoothness = 0.5)
  lambdaVals <- append(lambdaVals, tmp$summary["lambda"])
  aRangeVals <- append(aRangeVals, tmp$summary["aRange"])
  MLEVals <- append(MLEVals, tmp$summary["lnProfileLike.FULL"])
  spatialAgain <- spatialProcess(s, Zsim[,i], smoothness = 0.5,
                                lambda = fitOzone$summary["lambda"],
                                aRange = fitOzone$summary["aRange"])
  TrueMLEVals <- append(TrueMLEVals, spatialAgain$summary["lnProfileLike.FULL"])
}
```

Then we plot the MLE values of log lambda and aRange and the true value in magenta on a scatterplot.

```
plot(lambdaVals, aRangeVals, pch = 16, main = "Simulated Lambda vs aRange Values (n=200)")
points(fitOzone$summary["lambda"], fitOzone$summary["aRange"], col = "magenta", cex = 1.2, pch = 16)
legend("topright", legend = c("True Value"), col = c("magenta"), pch = 16)
```

## Simulated Lambda vs aRange Values (n=200)



Here we take the 95% confidence interval for both parameters.

```
lambdaVals <- sort(lambdaVals)
lambdaCI <- c(lambdaVals[5], lambdaVals[195])

aRangeVals <- sort(aRangeVals)
aRangeCI <- c(aRangeVals[5], aRangeVals[195])
```

```
## True Lambda Value: 0.08226039
```

```
## Lambda 95% CI: 0.00862096 0.2646218
```

```
## True aRange Value: 1.190598
```

```
## aRange 95% CI: 0.3719689 1.45287
```

It appears that the true values fall within the confidence intervals, so they do well enough.

### Assessing differences between MLE vals at true and estimated parameters (large sample theory)

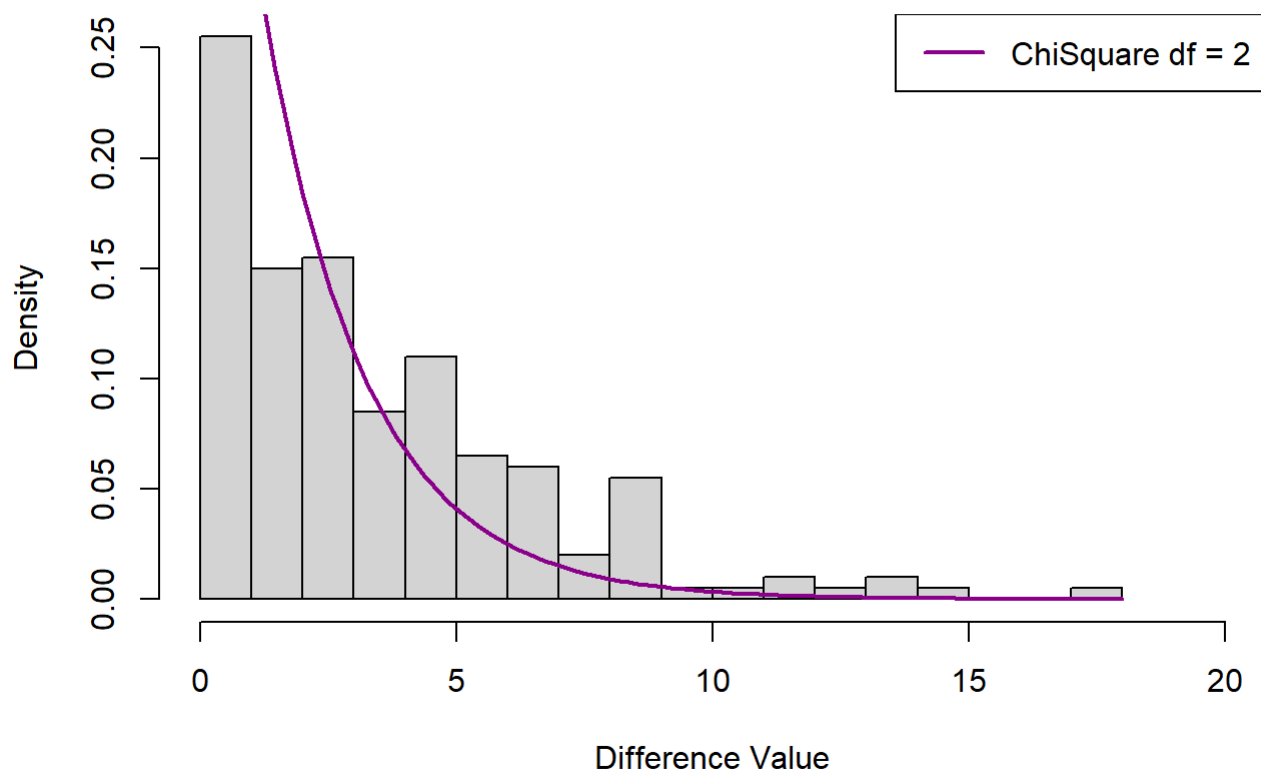
Calculations for this are already done in the for loop above.



```
MLEValDiffs <- 2*abs(MLEVals - TrueMLEVals)
hist(MLEValDiffs, probability = TRUE, xlab = "Difference Value", main = "Difference between log
likelihood at Simulated MLEs and True Values", xlim = c(0,20), breaks = 18)
legend("topright", legend = c("ChiSquare df = 2"), col = c("darkmagenta"), lwd = 2)

chiGrid <- seq(0, 18, length.out = 100)
lines(chiGrid, dchisq(chiGrid, 2), col = "darkmagenta", lwd = 2)
```

## Difference between log likelihood at Simulated MLEs and True Values



After looking at the histogram, it seems that the theory is close, but it could be significantly better. I suspect that the histogram looks nicer than the other possible plots that I could make. Maybe the fit of the curve to the histogram would look nicer with a larger sample.