Likelihood of Heirarchical Matern Gaussian Random Field

We model trait values of individuals as independent normally distributed variables around a mean $\bar{z}$ with variance v. In turn the mean $\bar{z}$ is a function of location $\boldsymbol{x} = (x_1, x_2)^\top$ and follows a stationary Gaussian random field with spatially homogeneous mean $\tilde{z}$ and Matern covariance $C(d)$ with collocated variance $V = C(0)$, characteristic length ξ and smoothness parameter $\nu = 1$. Hence, observed individual trait values z_1 and z_2 at locations $\boldsymbol{x}_1, \boldsymbol{x}_2$ separated at a distance d will be bivariate normally distributed with the following statistical properties:

$$\mathbb{E}z_1 = \mathbb{E}z_2 = \tilde{z},\ \mathbb{V}z_1 = \mathbb{V}z_2 = v + V, \mathbb{C}(z_1, z_2) = C(d),$$

$$\mathbb{E}(z_i|\bar{z}) = \bar{z}(\boldsymbol{x}_i),\ \mathbb{V}(z_i|\bar{z}) = v,\ \mathbb{C}(z_1, z_2|\bar{z}) = 0.$$

Since we only get a single sample of this random field, the likelihood function is just

$$L(\tilde{z}, v, V, \xi|\boldsymbol{z}) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left(-\frac{1}{2}(\boldsymbol{z} - \tilde{\boldsymbol{z}})\Sigma^{-1}(\boldsymbol{z} - \tilde{\boldsymbol{z}})^\top\right),$$

where $\boldsymbol{z} = (z_1, \dots, z_n)^\top$ is the vector of observed individual trait values (here there are n individuals) observed at locations $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, $\tilde{\boldsymbol{z}}$ is the vector of expected trait values $(\tilde{z}, \dots, \tilde{z})^\top$ and Σ is a covariance matrix with $v + V$ on the diagonal and $C(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$ on the i, jth off-diagonal entry.

To maximize likelihood, we first compute gradients of the log-likelihood $\ell = \ln L$ with respect to model parameters. We then solve for the parameter values when those gradients are zero:

$$\frac{\partial \ell}{\partial \tilde{z}} = -\frac{1}{2}\frac{\partial}{\partial \tilde{z}}\left[(\boldsymbol{z} - \tilde{\boldsymbol{z}})\Sigma^{-1}(\boldsymbol{z} - \tilde{\boldsymbol{z}})^\top\right] = -\frac{1}{2}\frac{\partial}{\partial \tilde{z}}\sum_{ij}(\sigma^{-1})_{ij}(z_i - \tilde{z})(z_j - \tilde{z})$$

$$= \sum_{ij}(\sigma^{-1})_{ij}\left(\tfrac{z_i + z_j}{2} - \tilde{z}\right) = \sum_{ij}\tfrac{z_i + z_j}{2}(\sigma^{-1})_{ij} - \tilde{z}\sum_{ij}(\sigma^{-1})_{ij},$$

$$\implies \hat{\tilde{z}} = \frac{\sum_{ij}\frac{z_i + z_j}{2}(\sigma^{-1})_{ij}}{\sum_{ij}(\sigma^{-1})_{ij}}.$$

Unfortunately, the rest is analytically intractable. So we proceed via numerical optimization.

```
require(MASS)
```

```
## Loading required package: MASS
```

```
require(rSPDE)
```

```
## Loading required package: rSPDE
```

```
## Loading required package: Matrix

##
## Attaching package: 'rSPDE'

## The following object is masked from 'package:stats':
##
##     simulate
```

```
require(clusterGeneration)
```

```
## Loading required package: clusterGeneration
```

```
Euc = function(x1,x2){
  val = sqrt(sum((x1-x2)^2))
}

# making the matern in
# my preferred parameterization
C = function(d,V,xi){
  matern.covariance(d,sqrt(2)*xi,1,sqrt(V))

}

Cmat = function(x,V,xi,v){
  n = dim(x)[1]
  val = matrix(0,n,n)
  for(i in 1:n){
    for(j in 1:n){
      dd = Euc(x[i,],x[j,])
      val[i,j] = C(dd,V,xi)
    }
  }
  return(val + v*diag(n))
}

hattildez = function(z,S){
  val = 0
  n = length(z)
  Sinv = solve(S)
  for(i in 1:n){
    for(j in 1:n){
      val = val + Sinv[i,j]*(z[i]+z[j])/2
    }
  }
  return(val/sum(Sinv))
}

hatSigtau = function(z,hatSig){
  ztilde = hattildez(z,hatSig)
  val = (z-ztilde)%*%t(z-ztilde)
  return(val)
```

```
}

# mean
ztilde = 0

# char scale
xi = 1

# marginal var
V = 3

# within pop var
v = 20

# number of individuals measured
n = 10

# their locations
x = mvrnorm(n,c(0,0),diag(2))

S = Cmat(x,V,xi,v)

obs = mvrnorm(1,rep(ztilde,n),S)

hatSigtau(obs,S)
```

```
##            [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
##  [1,]  17.523049 -15.127337  1.9948502  23.799596 -21.142694   9.283590
##  [2,] -15.127337  13.059162 -1.7221187 -20.545769  18.252112  -8.014359
##  [3,]   1.994850  -1.722119  0.2270967   2.709382  -2.406916   1.056858
##  [4,]  23.799596 -20.545769  2.7093817  32.324327 -28.715754  12.608861
##  [5,] -21.142694  18.252112 -2.4069160 -28.715754  25.510030 -11.201253
##  [6,]   9.283590  -8.014359  1.0568578  12.608861 -11.201253   4.918382
##  [7,] -12.948493  11.178204 -1.4740759 -17.586489  15.623196  -6.860022
##  [8,]  -1.892284   1.633575 -0.2154204  -2.570078   2.283163  -1.002519
##  [9,] -13.454997  11.615460 -1.5317371 -18.274416  16.234326  -7.128365
## [10,]  10.315626  -8.905297  1.1743463  14.010560 -12.446471   5.465147
##            [,7]       [,8]       [,9]      [,10]
##  [1,] -12.948493 -1.8922840 -13.454997  10.315626
##  [2,]  11.178204  1.6335752  11.615460  -8.905297
##  [3,]  -1.474076 -0.2154204  -1.531737   1.174346
##  [4,] -17.586489 -2.5700775 -18.274416  14.010560
##  [5,]  15.623196  2.2831632  16.234326 -12.446471
##  [6,]  -6.860022 -1.0025190  -7.128365   5.465147
##  [7,]   9.568168  1.3982856   9.942444  -7.622635
##  [8,]   1.398286  0.2043445   1.452982  -1.113967
##  [9,]   9.942444  1.4529821  10.331361  -7.920809
## [10,]  -7.622635 -1.1139668  -7.920809   6.072696
```

```
minThis = function(Y){
  V = exp(Y[1])
  xi = exp(Y[2])
  v = exp(Y[3])
```

```r
  n = dim(x)[1]
  hS = Cmat(x,V,xi,v)
  hSt = hatSigtau(obs,hS)
  val = Euc(hS,hSt)
  return(val)
}

ests = exp(optim(rexp(3),minThis)$par)

Cmat(x,ests[1],ests[2],ests[3])
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
##  [1,] 1.219953e+01 4.396897e-12 7.300799e-10 1.016052e-06 3.343600e-03
##  [2,] 4.396897e-12 1.219953e+01 7.335931e-04 5.019133e-06 4.155154e-08
##  [3,] 7.300799e-10 7.335931e-04 1.219953e+01 2.104130e-02 7.038902e-06
##  [4,] 1.016052e-06 5.019133e-06 2.104130e-02 1.219953e+01 5.775055e-03
##  [5,] 3.343600e-03 4.155154e-08 7.038902e-06 5.775055e-03 1.219953e+01
##  [6,] 5.028294e-05 4.982807e-09 3.492560e-05 3.396567e-02 1.074788e-02
##  [7,] 1.082283e+00 1.337997e-11 6.690306e-10 6.390634e-07 3.619402e-03
##  [8,] 3.467600e-07 8.892887e-13 2.075696e-13 2.205190e-11 4.885766e-08
##  [9,] 3.276165e-04 5.314757e-07 4.210043e-05 1.629017e-02 1.940822e+00
## [10,] 4.771360e-07 1.544253e-06 1.161879e-07 1.536152e-06 8.785521e-05
##              [,6]         [,7]         [,8]         [,9]        [,10]
##  [1,] 5.028294e-05 1.082283e+00 3.467600e-07 3.276165e-04 4.771360e-07
##  [2,] 4.982807e-09 1.337997e-11 8.892887e-13 5.314757e-07 1.544253e-06
##  [3,] 3.492560e-05 6.690306e-10 2.075696e-13 4.210043e-05 1.161879e-07
##  [4,] 3.396567e-02 6.390634e-07 2.205190e-11 1.629017e-02 1.536152e-06
##  [5,] 1.074788e-02 3.619402e-03 4.885766e-08 1.940822e+00 8.785521e-05
##  [6,] 1.219953e+01 9.862236e-06 1.570569e-11 3.794167e-03 5.179761e-08
##  [7,] 9.862236e-06 1.219953e+01 8.523866e-06 5.572508e-04 5.957155e-06
##  [8,] 1.570569e-11 8.523866e-06 1.219953e+01 4.692504e-08 2.545711e-05
##  [9,] 3.794167e-03 5.572508e-04 4.692504e-08 1.219953e+01 4.907338e-04
## [10,] 5.179761e-08 5.957155e-06 2.545711e-05 4.907338e-04 1.219953e+01
```