Spatio Temporal Data Analysis

Week 2: Lielihood-based Inference for Geostatistics

이번주는 시공간 데이터에 대해서 모델을 세우고 모수에 대해서 추론하는 과정에 대해서 배워본다. 시공간 데이터는 현재 likelihood-based, baysian approach의 방법으로 모델링할 수 있다. 두 방법 중, bayesian 방법이 더 복잡한 모형을 쉽게 모델링할 수 있어서 널리 쓰인다고 한다.

n개의 시공간 데이터를 관측했다고 하자.

$$\mathbf{Y} = (Y(s_1), \cdots, Y(s_n))$$

이 데이터를 가지고, $Y(s_1), \cdots, Y(s_n)$이 어떻게 발생했는지, 그 process에 대해서 추론하는 것이 이번 포스팅의 목적이다. 결국 이는, model을 specify하고 모수를 추정한다는 의미이다. 또한, 적절한 모형을 적합했으면 관측하지 않은 위치, s_0에서의 Y 값 또한 구할 수 있을 것이다. 즉, $Y(s_0)$를 예측하는 것도 적합한 모형을 이용해서 할 수 있다. 실제로 어떻게 이용할 수 있을까? 예를 들어, 미세먼지 관측소가 신촌역과 합정역에 있고 그 사이에는 비용 문제로 인해서 관측소가 없다고 하자. 홍대역에 있는 사람이 이곳의 미세먼지 정도를 알고 싶지만 관측소가 신촌역과 합정역에만 있기 때문에, 여태까지는 이 둘의 평균을 하는 등의 방법을 취했을 것이다. 하지만 신촌, 합정, 홍대의 위치 정보를 이용하여 홍대의 미세먼지를 모델링한다면, 그 값은 단순 평균보다는 유의미한 모델에 의해서 나온 것이며 통계적 추론도 할 수 있을 것이다.

이번 포스팅에서 사용할 데이터는 zinc concentration 데이터이다. 이 데이터는 특정 위치와 여기서의 topsoil concentration, 그리고 몇몇 변수를 함께 제공한다. 어떤 지역에 대한 value가 아니라, 특정 지점에서의 value이기 때문에 poin referenced data, 또는 geostatistical data 이다.

```
knitr::opts_chunk$set(comment=NA, fig.width=3, fig.height=3,fig.align='center',message=FALSE)
library(sp)
library(gstat)
library(nlme)
library(classInt)
library(fields)
data(meuse)
sprintf('meuse는 총 %s개의 데이터가 있음.', length(meuse))
```

```
## [1] "meuse는 총 14개의 데이터가 있음."
```

```
head(meuse)
```

```
##        x      y cadmium copper lead zinc  elev       dist   om ffreq soil lime
## 1 181072 333611    11.7     85  299 1022 7.909 0.00135803 13.6     1    1    1
## 2 181025 333558     8.6     81  277 1141 6.983 0.01222430 14.0     1    1    1
## 3 181165 333537     6.5     68  199  640 7.800 0.10302900 13.0     1    1    1
## 4 181298 333484     2.6     81  116  257 7.655 0.19009400  8.0     1    2    0
## 5 181307 333330     2.8     48  117  269 7.480 0.27709000  8.7     1    2    0
## 6 181390 333260     3.0     61  137  281 7.791 0.36406700  7.8     1    2    0
##   landuse dist.m
## 1      Ah     50
## 2      Ah     30
## 3      Ah    150
## 4      Ga    270
## 5      Ah    380
## 6      Ga    470
```

x, y 칼럼이 spatial location을 나타내고, zinc가 target 변수이다.

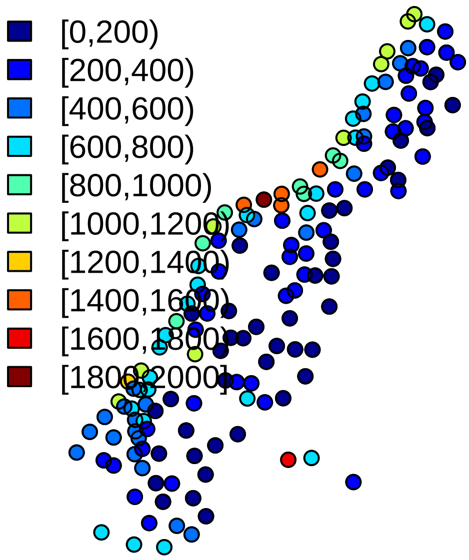
R에서 평범한 dataframe을 spatial object로 바꿔주는 함수에 대해서 살펴보자. 이는 coordiantes 함수를 이용한다.

```
coordinates(meuse) = c('x', 'y')
class(meuse)


[1] "SpatialPointsDataFrame"
attr(,"package")
[1] "sp"


plot.point.ref <- function(spatialdata, vals) {
pal <- tim.colors(10)
ints <- classIntervals(vals, n = 8, style = "pretty") # Determine breakpoints
intcols <- findColours(ints, pal) # vector of colors
par(mar = rep(0, 4))
plot(spatialdata, col = intcols, pch = 19)
points(spatialdata, pch = 1)
legend("topleft", fill = attr(intcols, "palette"), legend = names(attr(intcols, "table")), bty = '
}


plot.point.ref(meuse, meuse$zinc)
```



위 그림은 spatial 정보를 이용해서 meuse를 zinc의 색을 categorize하여 시각화한 것이다. 눈으로 보았을 때, 비슷한 색이 클러스터를 이룸을 대강 확인할 수 있다.

위와 같은 point referenced data에 대한 모델을 specify 해보자.

$$
\begin{aligned}
Y(s) &= \mu(s) + e(s) \\
&= \mu(s) + \eta(s) + \epsilon(s)
\end{aligned}
$$

- $\mu(s) = E[Y(s)]$: mean trend이다. Linear Model에서 $X\beta$라고 생각하면 된다. 모두 β를 포함하고 있으므로 $\mu(s;\beta) = X(s)'\beta$라고 표현한다.
  - 설명 변수 $X(s)$는 절편, 위도, 경도, 또는 다른 spatial covariates(온도, 기압 등)을 포함할 수 있다.
- $e(s)$: 평균이 0인 stationary process이다. (보통 Gaussian Process)
  - $\eta(s)$는 spatially correlated process이다.
  - $\epsilon(s)$는 correlation이 없는 white noise이다. (nugget or measurement error)
  - $\eta(s)$와 $\epsilon(s)$는 독립이라고 가정한다.
  - $e(s) \sim N(0, \sigma^2\Gamma(\rho) + \tau^2 I)$

Estimation through Variogram

classical geostatistical approach는 아래와 같다.

Step 1. spatial dependence를 고려하지 않고, β를 추정한다.

Step 2. residual을 이용하여 variogram을 추정한다.

Step 3. variogram이 spatial dependence가 있음을 나타낸다면, spatial dependence를 고려하여 β를 다시 추정한다.

이 방법은 통계적 관점에서 optimal하지는 않지만, EDA로는 좋은 방법이다. 여기서는 각 단계를 R에서 함께 해보도록 한다. 사용할 데이터는 위에서 살펴본 meuse 데이터이다. 타겟 변수는 zinc, 설명변수는 eleve, dist, om을 사용한다. zinc와 dist는 변수를 변환하여 사용한다.

```
meuse$logzinc <- log(meuse$zinc) # model log concentrations
meuse$sqrtdist <- sqrt(meuse$dist)
```

Step 1.

먼저 spatial dependence를 고려하지 않고 β를 추정한다. 이는 β에 대한 OLS를 구하는 것과 동일하다.

$$\hat{\beta}_{ols} = (X^{'}X)^{-1}X^{'}Y,\ \hat{\epsilon} = Y - X\hat{\beta}_{ols}$$

이는 R에서 간단하게 lm 함수를 써서 한다.

```
linmod <- lm(logzinc ~ elev + sqrtdist + om, data = meuse)
summary(linmod) # ignore standard errors!



Call:
lm(formula = logzinc ~ elev + sqrtdist + om, data = meuse)

Residuals:
     Min       1Q   Median       3Q      Max
-0.83163 -0.22055 -0.01086  0.23716  0.84062

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.03704    0.27276  29.465  < 2e-16 ***
elev        -0.23063    0.03210  -7.185 2.99e-11 ***
sqrtdist    -1.46316    0.18962  -7.716 1.59e-12 ***
om           0.05028    0.01144   4.394 2.10e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.3582 on 149 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.7584,Adjusted R-squared:  0.7536
F-statistic: 155.9 on 3 and 149 DF,  p-value: < 2.2e-16
```

Step 2.
이제 variogram을 추정한다. variogram의 population 버전이 무엇이었는지 생각해보자.

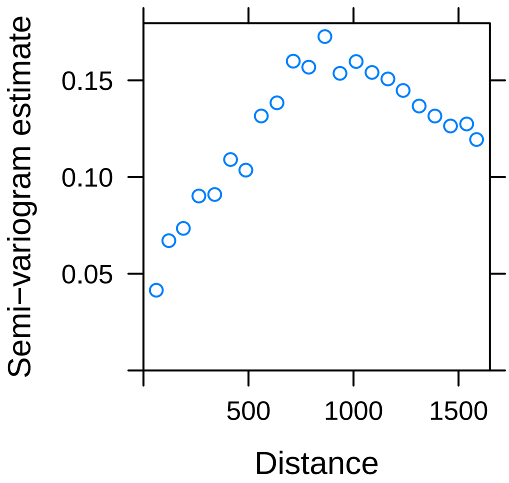
$$
\begin{aligned}
2\gamma(h) &= Var\left[e(s+h) - e(s)\right] \\
&= E\left[(e(s+h) - e(s))^2\right] \quad \because E\left[e(s)\right] = 0
\end{aligned}
$$

$A = e(s+h) - e(s)$라고 두면, 이는 A의 second moment이고, moment에 대한 대표적인 estimator는 Method of Moment Estimator가 있다. 이때 각 h에 대해 replication이 없으므로 binning을 해야한다. $H_1, \cdots, H_k$를 가능한 lags의 partition으로, h_u를 H_u에 대한 representative member라고 하자. MME를 사용하여 $\gamma(h)$을 아래와 같이 추정한다.

$$
\hat{\gamma}(h_u) = \frac{1}{2\#\{s_i - s_j \in H_u\}} \sum_{s_i - s_j \in H_u} \left[\hat{e}(s_i) - \hat{e}(s_j)\right]^2
$$

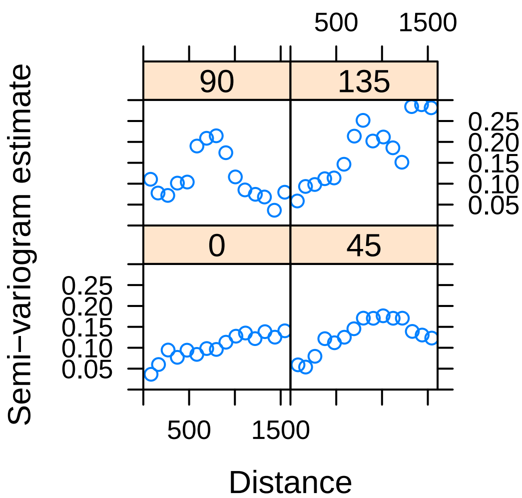
R에서는 아래와 같이 variogram을 추정한다. variogram() 함수는 데이터로부터 sample variogram을 계산해준다. 즉, MME를 계산해주는 것이다.

```r
fitted <- predict(linmod, newdata = meuse, na.action = na.pass)
ehat <- meuse$logzinc - fitted # residuals
meuse$ehat <- ehat
meuse.sub <- meuse[!is.na(ehat),] # Remove lines with missing data
vg <- variogram(ehat ~ 1, data = meuse.sub, width=75)
plot(vg, xlab = "Distance", ylab = "Semi-variogram estimate", width=5)
```



추정된 variogram을 이용하여 anisotropy 가정을 확인한다.

```r
vgangle <- variogram(ehat ~ 1, data = meuse.sub, alpha = c(0, 45, 90, 135))
plot(vgangle, xlab = "Distance", ylab = "Semi-variogram estimate")
```



subjective하지만, 눈으로 판단해보면 네 그림의 패턴이 크게 다르지 않은 것으로 보인다. 이렇게 rotating해도 패턴이 크게 바뀌지 않을 때, Geometric anisotropy를 만족한다고 하고, 이는 variogram이 coordinate space의 linear transformation에 대해 isotropic함을 의미한다. 정리해보면 meuse 데이터에 non-parametric estimate인 MME를 구해보았고 MME가 anisotropy를 만족하는 것으로 파악됐다. 그런데, anisotropy는 곧, isotropic함을 의미한다. 전통적으로 isotropic variogram (이 데이터가 isotropic 함을 밝힘)의 non-parametric estimate(이 데이터에서는 MME)은 wls을 이용하여, $\gamma(h)$에

대한 parametric model을 추정하는데 사용된다고 한다. 최소화하고자 하는 목적식은 아래와 같다.

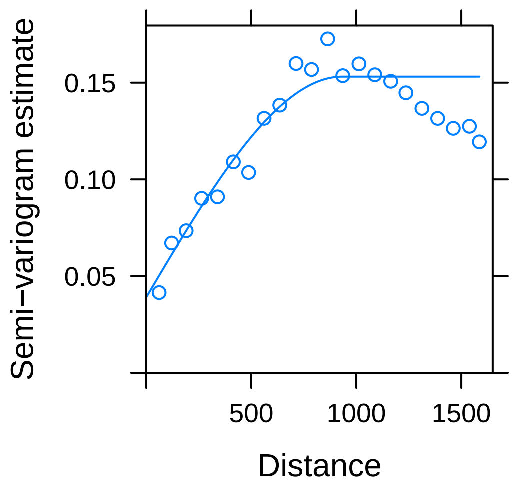
$$\sum_u \frac{n_u}{\gamma(h_u;\theta)} \left[\hat{\gamma}(h_u) - \gamma(h_u;\theta)\right]^2$$

이제, parametric model을 사용해야 하는데, 대표적으로 spherical variogram이 있다. non-parametric estimate을 이용하여 spherical variogram는 R에서 아래와 같이 추정한다.

```
# second argument has starting values
fitvg <- fit.variogram(vg, vgm(1, "Sph", 500, 0.05))
print(fitvg)


  model      psill    range
1   Nug 0.03903854   0.0000
2   Sph 0.11408603 932.8558


s2.hat <- fitvg$psill[2]
rho.hat <- fitvg$range[2]
tau2.hat <- fitvg$psill[1]
plot(vg, fitvg, xlab = "Distance", ylab = "Semi-variogram estimate")
```



gls 결과로, 각 parameter에 대한 추정치와 p-value, 변수 간의 correlation 구조, nugget parameter를 확인할 수 있다.

위 그림은 variogram에 대한 parametric model로, spherical variogram을 적합한 결과이다. 코드 부분에서, fit.variogram을 살펴보자. R documents을 보면 argument로 sample variogram, 즉 variogram

함수를 이용하여 얻은 output을 받고 model로는 variogram model, 즉 vgm 함수를 이용하여 얻은 output을 받는다고 나와있다. 여기서는 vgm(1, "Sph", 500, 0.05)을 사용했는데, 이는 spherical model을 의미한다.

그렇다면 그냥 non-parametric model을 쓰면 될텐데, 왜 굳이 이를 이용해서 parametric model까지 사용할까? covariance function이 충종해야할 몇 가지 요소가 있고 covariance functions은positive definite해야하는데 non-parametric model은 이러한 제약 조건을 충족하기 힘들다고 한다.

이제 β를 gls를 이용하여 다시 추정한다.

$$\hat{\beta}_{gls} = (X^{'}\Sigma^{-1}X)^{-1}X^{'}\Sigma^{-1}Y,\ \hat{\Sigma}_{ij} = C(s_i - s_j, \hat{\theta}_{wls})$$

```
gls.fit <- gls(logzinc ~ elev + sqrtdist + om, data = meuse.sub,
corSpher(value = c(range = rho.hat,
nugget = tau2.hat/(tau2.hat+s2.hat)),
nugget = TRUE, form=~x+y, fixed = TRUE))
summary(gls.fit)

Generalized least squares fit by REML
  Model: logzinc ~ elev + sqrtdist + om
  Data: meuse.sub
       AIC      BIC    logLik
  97.47863 112.4984 -43.73932

Correlation Structure: Spherical spatial correlation
 Formula: ~x + y
 Parameter estimate(s):
      range      nugget
932.8557713   0.2549463

Coefficients:
                Value  Std.Error   t-value p-value
(Intercept)  8.039659 0.29579249 27.180064       0
elev        -0.247831 0.02948049 -8.406615       0
sqrtdist    -1.274175 0.28571449 -4.459611       0
om           0.054219 0.01083674  5.003264       0

 Correlation:
         (Intr) elev   sqrtds
```

```
elev     -0.745
sqrtdist -0.350 -0.206
om       -0.536  0.061  0.489

Standardized residuals:
       Min         Q1        Med         Q3        Max
-1.9881813 -0.4542813  0.0399102  0.6089699  1.8710727

Residual standard error: 0.4148434
Degrees of freedom: 153 total; 149 residual
```

전체적인 순서는 아래와 같다.

$$\hat{\beta}_{ols} \to \hat{\gamma}(h) \to \hat{\theta}_{wls} \to \Sigma(\hat{\theta}_{wls}) \to \hat{\beta}_{gls}\ (Empirical\ BLEU)$$

Likelihood-based Inference for Geostatistics

이전까지 variogram을 사용하여 gls을 구하는 과정을 살펴보았다. 이 과정에서 variogram의 추정치를 이용하는데 non-parametric estimate로 MME를 사용하고, 이를 이용하여 variogram의 parametric model을 추정한다. 하지만 이 과정에서 estimates의 분산을 구할 수 없고 어느 variogram을 사용하는 것이 좋을지 판단할 수 없다. 즉, model comparison을 할 수 없다. 이러한 이유로 시공간 데이터 분석에서는 bayesian 방법이 가장 인기가 있다고 한다. 베이지안 방법은 다음주에 알아보도록 하고, 이번에는 Likelihood-based Inference에 대해서 알아본다.

Kriging

kriging은 시공간 데이터에서, best linear unbiased predictor (BLUP) 이다. kriging은 아래 조건을 만족한다.

$$Linearity:\ \hat{\mathbf{Y}}(s_0) = \lambda_0 + \lambda^{'}\mathbf{Y}$$

$$Unbiasedness:\ E\left[\hat{\mathbf{Y}}(s_0) - \mathbf{Y}(s_0)\right] = 0$$

$$minimizes\ Var\left[\hat{\mathbf{Y}}(s_0) - \mathbf{Y}(s_0)\right]$$

간단한 예시로, n개의 데이터가 주어졌을 때 1개의 관측되지 않은 값 s_0에서의 prediction, $Y(s_0)$을 예측하는 문제를 생각해보자. (Simple Kriging) 아래의 joint 분포를 고려한다.

$$\begin{pmatrix} \mathbf{Y} \\ Y(s_0) \end{pmatrix} \sim D\left(\begin{pmatrix} \mathbf{m} \\ m_0 \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma} & \gamma \\ \gamma^{'} & \sigma^2 \end{pmatrix}\right)$$

유도 과정을 거치면, 아래의 결과를 얻을 수 있다.

$$\hat{\mathbf{Y}}(s_0) = m_0 + \gamma^{'}\mathbf{\Sigma}^{-1}(\mathbf{Y} - \mathbf{m})$$

$$Var\left[\hat{\mathbf{Y}}(s_0) - \mathbf{Y}(s_0)\right] = \sigma^2 - \gamma^{'}\mathbf{\Sigma}^{-1}\gamma$$

simple kriging에서는 모든 모수를 안다고 가정하고 논의를 진행한다.
이제 $E\left[Y(s)\right] = X(s)^{'}\beta$ 라고 가정하자. β 는 unknown, θ 는 여전히 known이다. 이러한 frame을 universal kriging이라고 하는데, β 에 대한 gls를 구하고 이를 kriging predictor에 대입한다.

$$\hat{\mathbf{Y}}(s_0) = X(s_0)^{'}\hat{\beta}_{gls} + \gamma^{'}\mathbf{\Sigma}^{-1}(\mathbf{Y} - X(s_0)^{'}\hat{\beta}_{gls})$$

그런데 사실, 실제 데이터에서는 θ 도 모른다. 하나의 옵션은 variogram estimation을 θ 의 estimate로 넣는 것이다. 이 추정치는 empirical BLUP라고 알려져 있는데, 많은 단점이 있다. 이러한 이유로 데이터가 Gaussian Process로부터 생성되었다고 가정하고, Likelihood를 세운 뒤, 통계적 추론을 하는 방법이 제안되었다. (물론 베이지안 방법이 더 좋다.)

Profile Likelihood for Spatial Data

먼저 관측된 n 개의 데이터 $\mathbf{Y} = (Y(s_1), \cdots, Y(s_n))^{'}$ 에 대해서 mean vector $\mathbf{X}\beta$ 와 $n \times n$ 공분산 행렬 $\mathbf{\Sigma}(\theta)$ 를 가정하자. 또한 Likelihood를 세우기 위해 분포 가정을 해야하는데, 가장 많이 쓰이는 Gaussian Process를 가정하자. 시공간 데이터에서는 데이터 간의 dependency를 허용하기 때문에 공분산 행렬의 off diagonal element도 생각을 해야 한다.

$$\mathbf{\Sigma}(\theta)_{ij} = Cov(Y(s_i), Y(s_j)) = C(s_i, s_j; \theta)$$

분포 형태로 적으면

$$\mathbf{Y} \sim N(\mathbf{X}\beta, \mathbf{\Sigma}(\theta))$$

모수는 β, θ 이므로 likelihood function을 적으면

$$L(\beta, \theta) = (2\pi)^{-n/2}|\mathbf{\Sigma}|^{-1/2}exp\left\{-\frac{1}{2}(\mathbf{Y} - \mathbf{X}\beta)^{'}\mathbf{\Sigma}(\theta)^{-1}(\mathbf{Y} - \mathbf{X}\beta)\right\}$$

보통, θ 를 fix하고 β 에 대한 maximizer를 구한 뒤, 이를 plug in해서 다시 θ 에 대한 maximizer를 구한다. 이러한 방법을 profiling이라고 하고, 이는 β, θ 에 대해 동시에 maximize하는 것과 수학적으로 동일하다고 한다. θ 를 fix하고 구한 β 에 대한 maximizer는

$$\hat{\beta}(\theta) = (\mathbf{X}^{'}\mathbf{\Sigma}(\theta)^{-1}\mathbf{X})^{-1}\mathbf{X}^{'}\mathbf{\Sigma}(\theta)^{-1}\mathbf{Y}$$

이를 likelihood에 대입하고 θ 에 대한 maximizer를 찾으면 된다.