Random processes project

Jiahui Xin

2022-06-18

1 Basic Parts

1.1 3.9

Exercise

Let $x(t)$ be a stationary Gaussian process with $E(x(t)) = 0$, covariance function $r_x(t)$, and spectral density $f_x(\omega)$. Find the covariance function for

$$y(t) = x^2(t) - r_x(0),$$

and show that it has the spectral density

$$f_y(\omega) = 2\int_{-\infty}^{\infty} f_x(\mu) f_x(\omega - \mu)\mathrm{d}\mu.$$

Solution

We first calculate the moment $E\left(u^2v^2\right)$ for bivariate normal variables u, v. It can be found from the coefficient of $s^2t^2/4$! in the series expansion of the moment generating function (or the characteristic function),

$$\begin{aligned} E\left(e^{su+tv}\right) = 1 + sE(u) + tE(v) + \frac{s^2E\left(u^2\right) + 2stE(uv) + t^2E\left(v^2\right)}{2!} \\ + \ldots + \frac{\ldots + 6s^2t^2E\left(u^2v^2\right) + \ldots}{4!} + \ldots \end{aligned}$$

If u, v are bivariate Gaussian variables, with mean 0, variance 1, and correlation coefficient ρ, then the moment generating function is

$$E\left(e^{su+tv}\right) = e^{\frac{1}{2}\left(s^2+2\rho st+t^2\right)} = \ldots + \frac{\ldots + 2s^2t^2 + 4\rho^2s^2t^2 + \ldots}{8} + \ldots$$

giving $E\left(u^2v^2\right) = 1 + 2\rho^2$. Thus, $r_y(\tau) = Cov\left(x^2(0), x^2(\tau)\right)$ is equal to $E\left(x^2(0)x^2(\tau)\right) - r_x^2(0) = (1 + 2\rho_\tau^2)r_x^2(0) - r_x^2(0) = 2r_x^2(\tau)$.

The second part follows immediately from the fact that multiplication of two covariance functions (here $r_x(\tau)$ and $r_x(\tau)$) corresponds to convolution of the corresponding spectral densities. Note the analogy between convolution of probability densities and multiplication of the characteristic functions ($f_1 * f_2 \leftrightarrow r_1 \times r_2$).

As a result, we have

$$f_y(w) = 2f_x(w) * f_x(w) = 2\int_{-\infty}^{\infty} f_x(\mu) f_x(w - \mu) d\mu.$$

1.2 Ornstein–Uhlenbeck process

Further, assume that the process in the title is an OU process that satisfies the following stochastic differential equation:

$$ax(t) + x'(t) = \sqrt{2a}w'(t), w'(t) \text{ where } w'(t) \text{ is Gaussian white noise.}$$

1.2.1 Analytical nature of the process, applicable scenarios, and programming to sample from the process (sample path function display).

The general form of Ornstein–Uhlenbeck process satisfies:

$$dx_t = \theta\left(\mu - x_t\right)dt + \sigma dW_t,$$

here W_t is Wiener process (i.e., dW_t is Gaussian white noise).

The stochastic differential equation for x_t can be formally solved by variation of parameters. Writing

$$f\left(x_t, t\right) = x_t e^{\theta t}$$

we get

$$\begin{aligned} df\left(x_t, t\right) &= \theta x_t e^{\theta t} dt + e^{\theta t} dx_t \\ &= e^{\theta t}\theta\mu dt + \sigma e^{\theta t} dW_t. \end{aligned}$$

Integrating from 0 to t we get

$$x_t e^{\theta t} = x_0 + \int_0^t e^{\theta s}\theta\mu ds + \int_0^t \sigma e^{\theta s} dW_s$$

whereupon we see

$$x_t = x_0 e^{-\theta t} + \mu\left(1 - e^{-\theta t}\right) + \sigma\int_0^t e^{-\theta(t-s)} dW_s.$$

Corresponding sample function we can get as follows.

```
r_ou <- function(T,n,mu,theta,sigma,x0){
  # dX(t) = theta * (mu - X(t)) * dt + sigma * dW(t)
  #T: period length, n: number of simulation
  dw  <- rnorm(n, 0, sqrt(T/n))#differences of wiener process
  dt  <- T/n
  x <- c(x0)
  for (i in 2:(n+1)) {
    x[i]  <-  x[i-1] + theta*(mu-x[i-1])*dt + sigma*dw[i-1]
  }
  return(x[-1]);
}
```

With the original denotion, $\theta = a, \mu = 0, \sigma = \sqrt{2a}$. Furthermore we set $x_0 = 0$

```
a=1
x<-r_ou(T=10,n=100,mu=0,theta=a,sigma=sqrt(2*a),x0=0)
head(x,10)#plot(x,type="l")
```

```
##  [1] -0.389952196 -0.203606693 -0.001888576  0.324714597  0.823559097
##  [6]  0.619903046  0.504852063  0.374166776  0.256329316  0.943489067
```

1.2.2 The spectral representation of the process, including covariance function, spectral density (or periodogram function), discuss the properties of the process parameters on the above two functions, and and the effects of the nature of the stochastic process itself.

About x_t

First consider the general form, then plug a into it.

$$\mathrm{E}\left(x_t\right) = x_0 e^{-\theta t} + \mu\left(1 - e^{-\theta t}\right).$$

$$
\begin{aligned}
\operatorname{cov}\left(x_s, x_t\right) &= \mathrm{E}\left[\left(x_s-\mathrm{E}\left[x_s\right]\right)\left(x_t-\mathrm{E}\left[x_t\right]\right)\right] \\
&= \mathrm{E}\left[\int_0^s \sigma e^{\theta(u-s)} dW_u \int_0^t \sigma e^{\theta(v-t)} dW_v\right] \\
&= \sigma^2 e^{-\theta(s+t)} \mathrm{E}\left[\int_0^s e^{\theta u} dW_u \int_0^t e^{\theta v} dW_v\right] \\
&= \frac{\sigma^2}{2\theta} e^{-\theta(s+t)}\left(e^{2\theta \min(s,t)}-1\right) \\
&= \frac{\sigma^2}{2\theta}\left(e^{-\theta|t-s|}-e^{-\theta(t+s)}\right).
\end{aligned}
$$

Let $t, s \to \infty$, the covariance function of x_t is $r_x(t) = \frac{\sigma^2}{2\theta} e^{-\theta|t|}$. Plug $\theta = a, \sigma = \sqrt{2a}$ into it,

$$
r_x(t) = e^{-a|t|}.
$$

The corresponding spectral density is

$$
f_x(w) = \frac{1}{2\pi}\int_{\mathbb{R}} e^{-iwt} r_x(t) dt = \frac{a}{\pi(a^2+w^2)}.
$$

About $y_t = x_t^2 - r_x(0)$

By the conclusion before, we have

$$
r_y(t) = 2r_x^2(t) = 2e^{-2a|t|}
$$

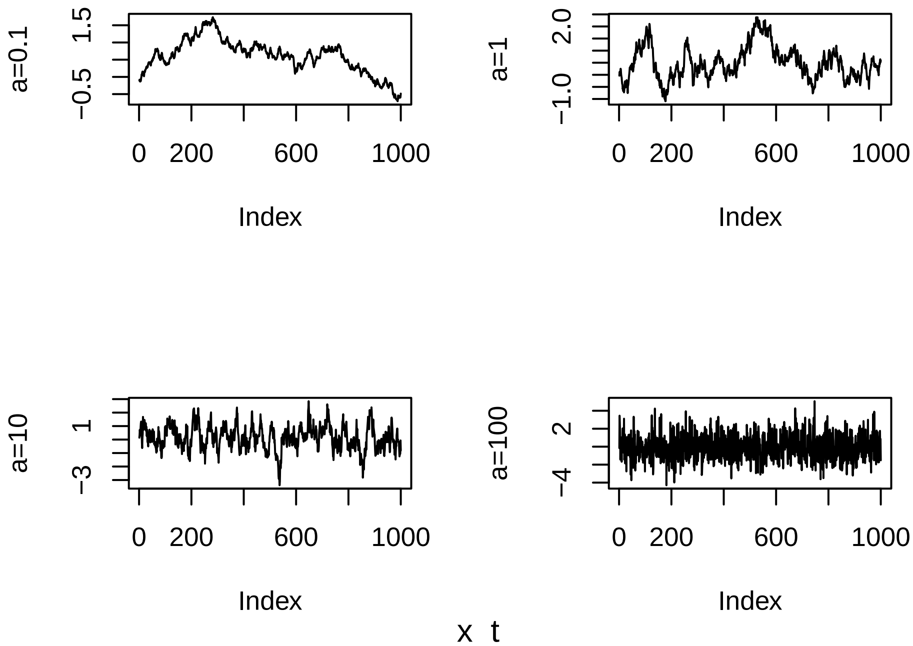
and

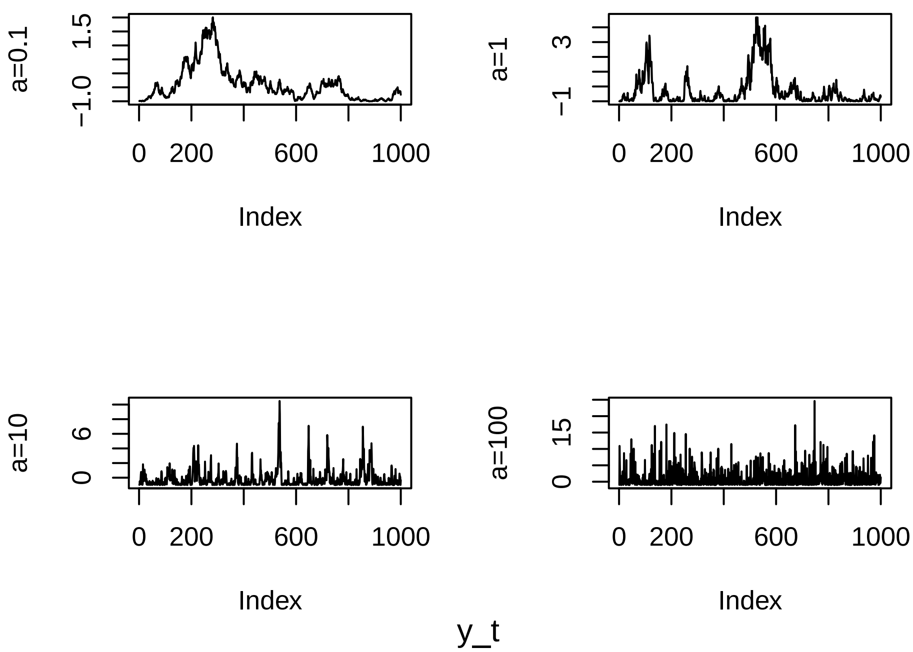
$$
\begin{aligned}
f_y(w) &= \frac{1}{2\pi}\int_{\mathbb{R}} e^{-iwt} r_y(t) dt \\
&= 2\frac{1}{2\pi}\int_{\mathbb{R}} e^{-iwt} r_x(t, 2a) dt, \text{ (Note that } r_y(t,a) = 2r_x(t,2a)) \\
&= 2\frac{2a}{(2a)^2+w^2} \\
&= \frac{4a}{4a^2+w^2}.
\end{aligned}
$$

About Parameter a

The parameter a can be seen as the dependence between sample points. For large $a(a \to \infty)$, the covariance function falls off very rapidly around $t = 0$ and the correlation between $x(s)$ and $x(t)$ becomes negligible when $s \neq t$. With increasing a, the spectral density becomes increasingly flatter at the same time as $f(w) \to 0$. But whatever a, $r_x(0) = 1$ which means the process keeps the same variance. Actually, $\lim_{a\to\infty} x_t$ is Gaussian white noise and $\lim_{a\to 0} x_t$ is constant process.

y_t has similar covariance function and spectral density.

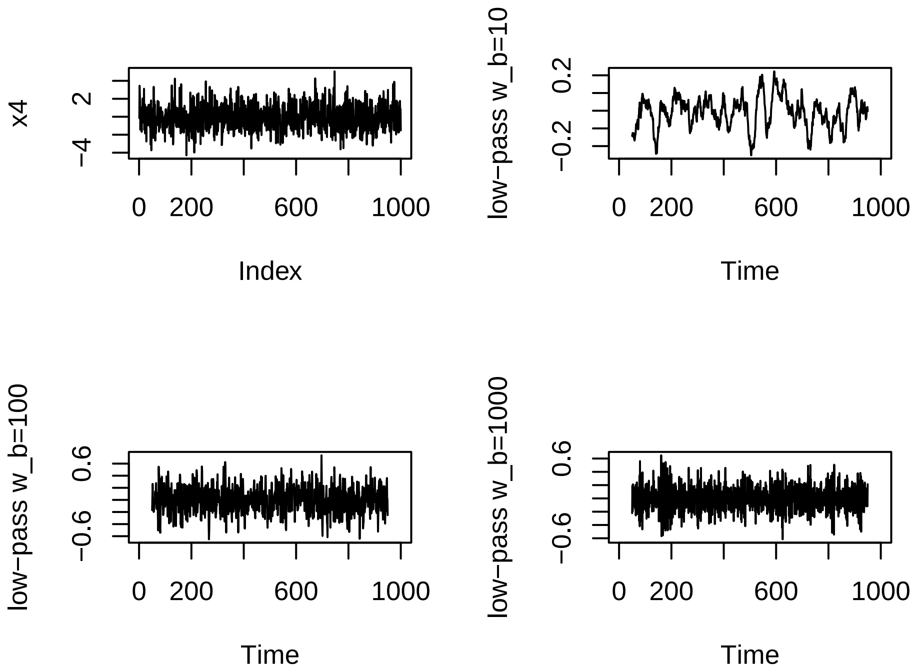




1.2.3 Perform linear time-invariant filtering (filters) on the process to verify the effects of different filters.

$$\hat{x}(t) = \int_{-\infty}^{\infty} h(u)x(t-u)\mathrm{d}u = \int_{-\infty}^{\infty} h(t-u)x(u)\mathrm{d}u$$

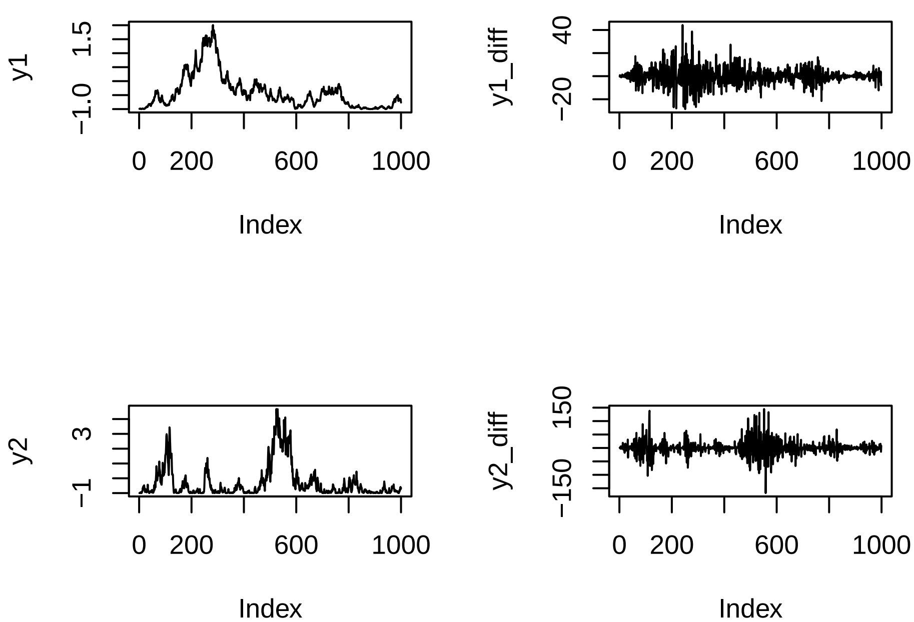
Low-pass filter Low-pass filter with $g(w) = \frac{1}{w_b}$ when $|w| < w_b$, and 0 otherwise, which has $h(u) = \frac{\sin(w_b u)}{2\pi u}$.



x4 low−pass filtered with w_b

deviation filter (Increasing high-frequency part)

$$\hat{x}(t) = x'(t) \text{ with impulse response } \delta_0'(u) \text{ and frequency response } iw.$$



y filtered by differential operator

1.2.4 Generate several samples of the random process, state the calculation steps of KL decomposition, and program to realise the process and display the results.

Karhunen-Loève expansion

Generate 1000 sample paths with 100 points per path.

Use pca to get coefficients and rotations (eigenfunctions) of KL expansion:

- step 1, calculate empirical covariance S.
- step 2, Do *singular value decomposition* onto $S = UDU'$ to get D (component variance) and U (rotation matrix).
- step 3, transform original data matrix $X_{n\times p}$ to $Y_{n\times p} = X_{n\times p}U$ and donote Y as coefficient matrix.
- step 4, keep the first n_{trun} columns of U and Y as U_{trun} and Y_{trun} respectively and calculate the n_{trun}-order approximation $X_{trun} = Y_{trun}U'_{trun}$.

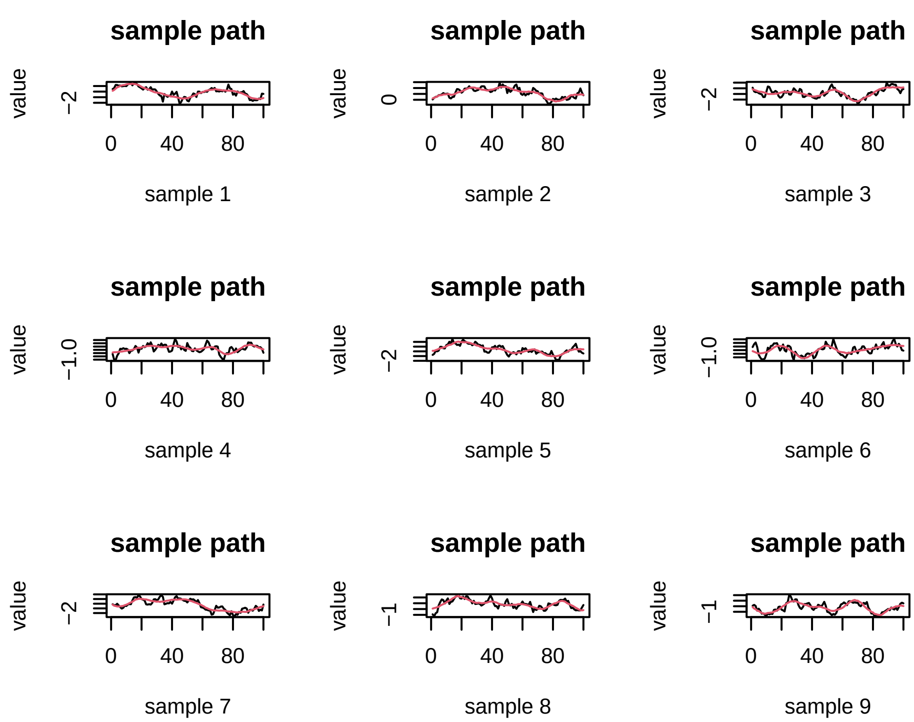
```
pca<-prcomp(data,center=F)

coeff=pca$x
rotat=pca$rotation
```

Test normalality and magnitude of coefficient of eigenfunctions (p-value of nomality test).

```
## [1] "coefficent 1 with variance 20.2776978105827 has p-value 0.689614728710347"
## [1] "coefficent 2 with variance 14.1990985401619 has p-value 0.894557753927704"
## [1] "coefficent 3 with variance 11.9274212919427 has p-value 0.621895175904131"
```

It is ok to fit a trend with only the first 10 eigenfunctions.



2 Application

Use Whittle-Matern Family (see Example 4.11) to fit an real data set, briefly state the purpose and process of the analysis, and show the numerical result.

2.1 Use R package *gstat* and *sp* to implement krigging algorithm (fitting variogram with Matern family)

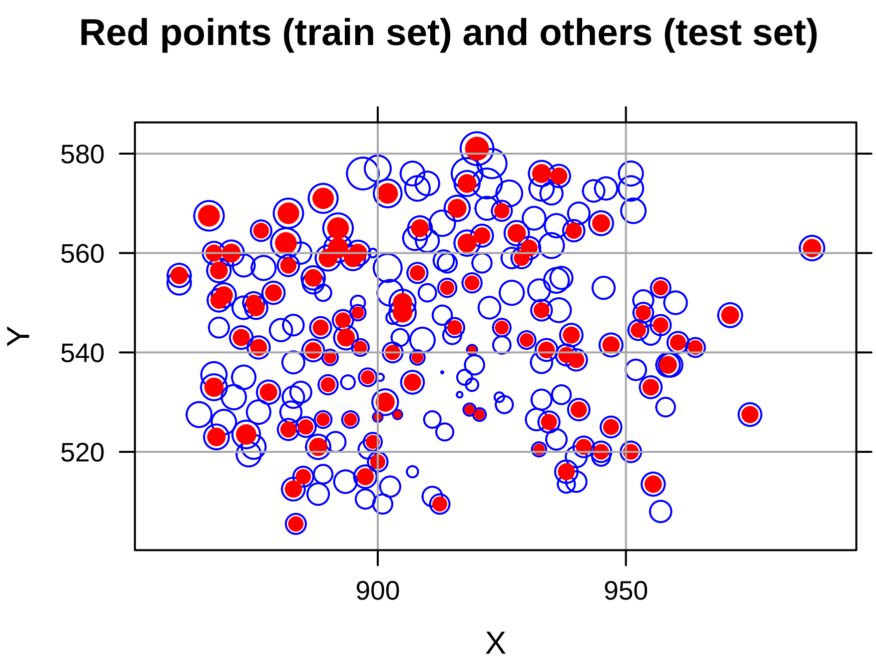
Data is *baltimore* in the package *spData*. I used *krigging* algorithm to predict unobserved point values in the field and in the step of fitting variogram, I used *Mattern* family.

Here I only used the price and the coordinates to construct a rough spatial model.

```
names(baltimore)
```

```
##  [1] "STATION" "PRICE"   "NROOM"   "DWELL"   "NBATH"   "PATIO"   "FIREPL"
##  [8] "AC"      "BMENT"   "NSTOR"   "GAR"     "AGE"     "CITCOU"  "LOTSZ"
## [15] "SQFT"    "X"       "Y"
```

2.2 Split train set and test set (Vaildation & Cross validation)

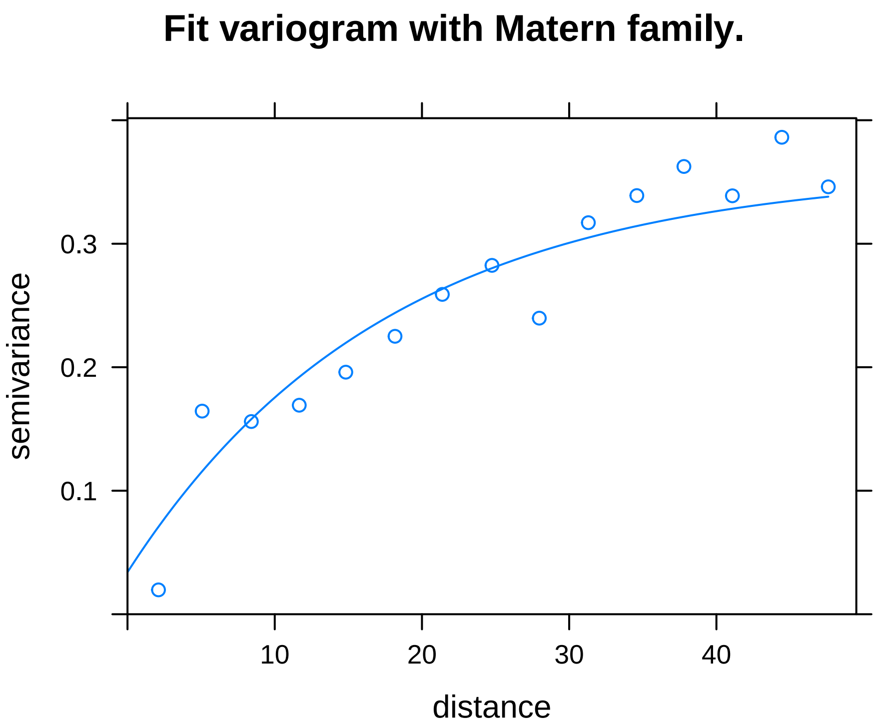


Fitted variogram with *Matern* family on the train set.

```
vgm.train = variogram(lgpr~1, baltimore.train)
fit.train = fit.variogram(vgm.train, model = vgm(1, "Mat", 30, 1))
fit.train
```

```
##   model      psill    range kappa
## 1   Nug 0.03403776  0.00000   0.0
## 2   Mat 0.32573970 17.56796   0.5
```

```
plot(vgm.train, fit.train, main="Fit variogram with Matern family.")
```



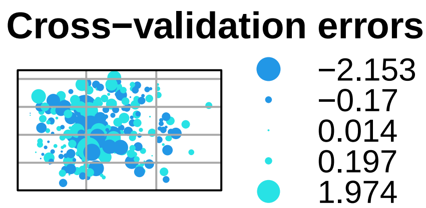
Then use the fitted variogram to do *krigging* on the test set.

Also used the fitted variogram on the train set to do krigging (cross-validation version) on the whole set. Compare the error graphs as follows.

```
k <- krige(lgpr ~ 1, baltimore.train, baltimore.test, fit.train)
```

```
## [using ordinary kriging]
```

```
cv.o <- krige.cv(lgpr ~ 1, baltimore, model=fit.train, nfold=5,
verbose=FALSE)
#summary(cv.o)
```



The fitted model seems good with small MSE and MAE.

```
summary(cv.o$observed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.253   3.432   3.689   3.652   3.984   5.106
```

```
sqrt(mean((cv.o$residual)^2))#mse
```

```
## [1] 0.4521721
```

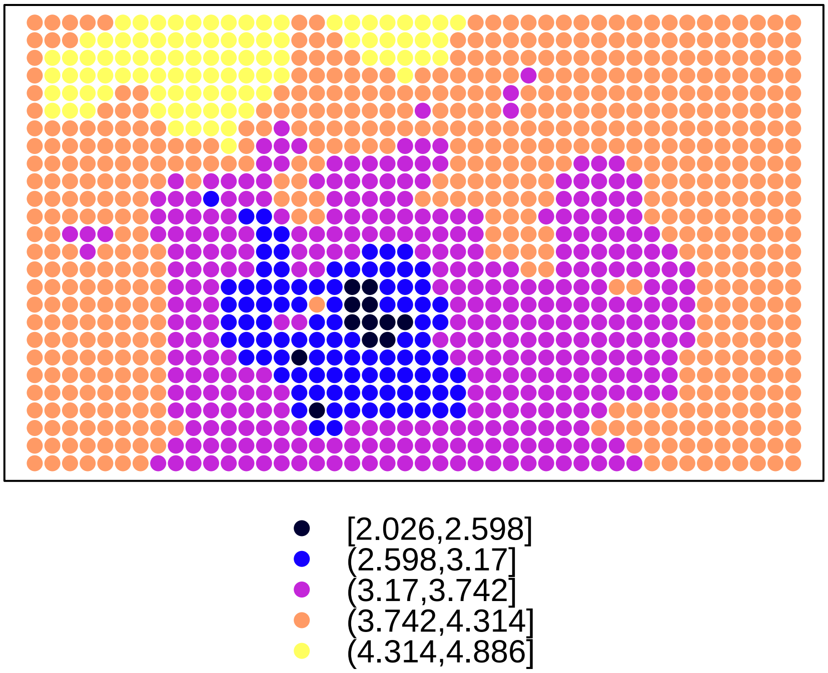
```
sqrt(mean(abs(cv.o$residual)))#mae
```

```
## [1] 0.5399519
```

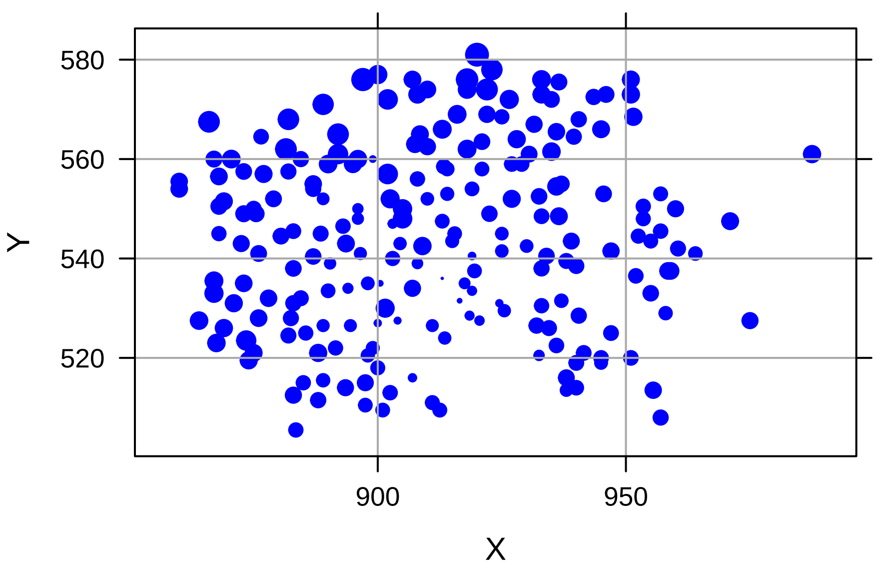
At last, I generated many grid points and predicted the price at these points with the fitted model on the train set.

```
## [using ordinary kriging]
```

Prediction valude at grid (cellsize=3)



True distribution (point size indicates value)



The prediction recovers the pattern which several annular bands locates around a low price center.