spmoran: An R package for Moran's eigenvector-based spatial regression analysis

Daisuke Murakami
Department of Data Science, Institute of Statistical Mathematics
10-3 Midori-cho, Tachikawa, Tokyo, Japan
E-mail: dmuraka@ism.ac.jp

Table of Contents

Latest update: 2018/04

- The resf_vc function that estimates a spatially varying coefficient (SVC) model is accelerated. Besides, the accelerated function automatically selects SVCs to stabilize the estimates.
- Functions for spatial interpolation are implemented.

1. Introduction

Eigenvector spatial filtering (ESF; e.g., Griffith, 2003), which is also known as Moran's eigenvector mapping (MEM; e.g., Dray et al., 2006), is a regression approach to estimate and infer regression coefficients in the presence of spatial dependence. Recently, ESF is extended to random effects ESF (RE-ESF; Murakami and Griffith, 2015). RE-ESF increases the estimation accuracy of regression coefficients and their standard errors with shorter computational time. RE-ESF is also extended to spatially varying coefficient (SVC) modeling (Murakami et al., 2017). The package "spmoran" provides R functions for fast estimation of ESF and RE-ESF models with/without SVCs.

This tutorial applies ESF and RE-ESF to a land price analysis of flood hazard. The target area is Ibaraki prefecture, Japan. Explained variables are logged land prices in 2015 (JPY/m^2; sample size: 647; Figure 1). Explanatory variables are as listed in Table 1. All these variables are downloaded from the National Land Numerical Information download service (NLNI; http://nlftp.mlit.go.jp/ksj-e/index.html).

The following is a data image, in which "px" and "py" are spatial coordinates:

```
> data    <- read.csv( "data.csv" )
> data[ 1:6, ]
        px          py     ln_price     station        tokyo   city   flood
1  19235.25   -4784.562   10.126631   4.0109290     43.38504      1     1.5
2  16450.37   -8782.851   10.835652   0.8977986     43.38504      1     0.0
3  17673.30   -8351.802   10.633449   0.5596742     43.38504      1     0.0
4  17824.50   -7704.343    9.878170   0.8504618     43.38504      0     0.0
5  67334.31   58001.724   10.122623   3.1660661    140.95839      1     0.0
6  68929.42   55028.751    9.952278   2.5008292    140.95839      1     1.5
```

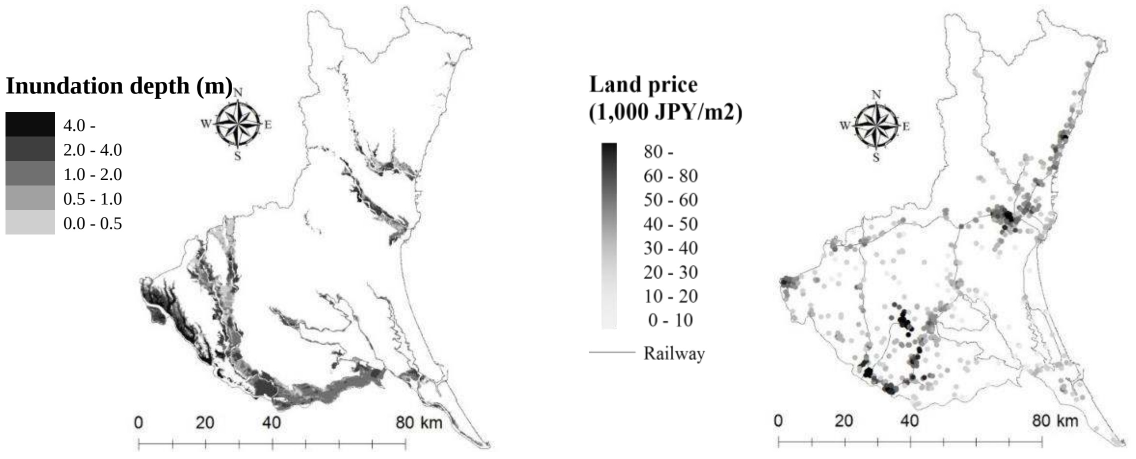

Figure 1. Anticipated inundation depth (left) and officially assessed land prices in 2015 (right) in the Ibaraki prefecture

Table 1. Explanatory variables

Variables	Description
tokyo	Logarithm of the distance from the nearest railway station to Tokyo Station [km]
station	Logarithm of the distance to the nearest railway station [km]
flood	Anticipated inundation depth [m]
city	1 if the site is in an urban promotion land and 0 otherwise

ESF/RE-ESF are implemented in the following two steps:

- Extraction of Moran's eigenvectors (see Section 2);
- Parameter estimation of the ESF/RE-ESF model (see Sections 3 and 4).

Sections 2, 3 and 4 explain implementation of these steps, while Section 5 explains how to accelerate the computation.

2. Extraction of Moran's eigenvectors

Consider a doubly-centered spatial connectivity matrix, $\mathbf{MCM}$, where $\mathbf{C}$ is a symmetric spatial proximity matrix whose diagonals are zeros, $\mathbf{M} = \mathbf{I} - \mathbf{11}'/N$ is a centering operator, where $\mathbf{I}$ is an identity matrix, and $\mathbf{1}$ is a vector of ones, and N is the sample size. The eigenvectors, $\mathbf{E} = \{\mathbf{e}_1,\ldots, \mathbf{e}_N\}$, of $\mathbf{MCM}$ furnish all possible distinct map pattern descriptions of latent spatial dependence, with each level being indexed by the Moran coefficient (MC; Griffith, 2003; Tiefelsdorf and Griffith, 2007). Eigenvectors corresponding to large positive eigenvalue describe map patterns with greater positive spatial dependence (i.e. greater positive MC), whereas eigenvectors corresponding to negative eigenvalue describe map patterns with negative spatial dependence. As positive spatial dependence is dominant in most real-world cases, only eigenvectors with positive eigenvalues are considered in many applied studies.

The function meigen extracts eigenvectors corresponding to positive eigenvalue (i.e. $\lambda_l > 0$, where λ_l is the l-th eigenvalue)[1]. The command is as follows:

```
> coords  <- data[ ,c( "px", "py" ) ]
> meig    <- meigen( coords = coords )
```

Calculated eigenvectors and eigenvalues are displayed by commanding meig$sf and meig$ev, respectively. By default, $\mathbf{C}$ is given by the matrix whose (i, j)-th element equals $\exp(-d_{i,j}/r)$, where

[1] For the distance-based $\mathbf{C}$, it is standard to set the threshold by $\lambda_l > 0$, which attempts to consider all elements describing positive spatial dependence.

$d_{i,j}$ is the Euclidean distance between sites i and j, and r is the longest distance in the minimum spanning tree covering the sample sites (Dray et al., 2006; Murakami and Griffith, 2015).

The distance-based **C** may be replaced with other types of spatial connectivity matrix. In this case, user must construct the matrix *a priori*. For example, the following command employs the 4-nearest-neighbor-based **C**:

```
> library( spdep )
> col.knn <- knearneigh( coordinates( coords ), k = 4 )
> cmat    <- nb2mat( knn2nb( col.knn ), style = "B" )
> meigB  <- meigen( cmat = cmat )
```

If the spatial connectivity matrix is not symmetric like the 4-nearest neighbor-based **C**, meigen symmetrizes it by taking{**C** + t(**C**)}/2. In cases with binary connectivity-based **C** (e.g. proximity-based **C**; k-nearest-neighbor-based **C**), $\lambda_l\ /\lambda_1 > 0.25$ is a standard threshold for the eigenvector extraction[2]. The thresholding is implemented by the following command:

```
> meigB  <- meigen( cmat = cmat, threshold = 0.25 )
```

The eigen-decomposition can be very slow for large samples. To accelerate the computation, the function meigen_f approximates the eigenvectors by applying the Nystrom extension, which is a dimension reduction technique (Murakami and Griffith, 2017)[3]. The command is as follows:

```
> meig_f <- meigen_f( coords = coords )
```

Just like meigen, meig_f$sf and meig_f$ev return approximated eigenvectors and eigenvalues, respectively. By default, the first 200 eigenvectors are approximated[4]. While meigen takes 243.79 seconds for the exact eigen-decomposition, meigen_f takes only 0.38 seconds (see Section 5 for further details).

[2] The threshold $\lambda_l\ /\lambda_1 > 0.25$ attempts to capture roughly 5% of the variance in explained variables attributable to positive spatial dependence (Griffith and Chun, 2014).

[3] This approximation is available only for the distance-based **C**.

[4] Consideration of 200 eigenvectors is recommended because Murakami and Griffith (2017) show that the approximation error in regression coefficients is quite small when 200 (or more) eigenvectors are considered while the error increases in cases with fewer than 200 eigenvectors.

3. ESF and RE-ESF models

3.1. ESF model

The linear ESF model is formulated as follows:

$$\mathbf{y} = \mathbf{X\beta} + \mathbf{E\gamma} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}),$$

where **E** is a matrix whose l-th column is the l-th eigenvector, $\mathbf{e}_l$, and $\boldsymbol{\gamma}$ is a vector of coefficients. The term $\mathbf{E\gamma}$ eliminates residual spatial dependence to estimate and infer regression coefficients appropriately while avoiding the Type I error. This model is identical to the standard linear regression model.

The ESF model is estimated using the following steps: (i) eigenvectors whose eigenvalue exceeds a threshold are extracted from **MCM**; (ii) stepwise eigenvector selection is performed; (iii) the ESF model with selected eigenvectors is estimated by ordinary least squares.

The following command estimates the linear ESF model. In step (i), following many ESF studies (Griffith, 2003; Tiefelsdorf and Griffith, 2007), eigenvectors whose eigenvalue fulfills $\lambda_l / \lambda_1 >$ 0.25 are extracted from a binary connectivity-based **C** (4-nearest-neighbor-based **C**; see Section 2):

```
> y       <- data[ ,"ln_price" ]                                # Explained variables
> x       <- data[ ,c( "station", "tokyo", "city", "flood " ) ]  # Explanatory variables
> meig    <- meigB                                              #Moran's eigenvectors (knn-based C)
> e_res   <- esf( y = y, x = x, meig = meig, vif = 10, fn = "r2" )
```

To cope with possible multicollinearity, eigenvectors are selected so that the variance inflation factor (VIF), which is an indicator of multicollinearity, does not exceed 10. It is implemented by setting vif = 10, whereas VIF is not considered by default. The eigenvector selection is performed by the adjusted R^2 maximization (fn = "r2"; default). Akaike information criterion (AIC) minimization (fn = "aic") or Bayesian Information criterion (BIC) minimization (fn = "bic"). Alternatively, all eigenvectors are considered without selecting them by setting fn = "all".

When fn = "r2", the coefficient estimates yield:

```
> e_res$b
                 Estimate            SE        t_value        p_value
 (Intercept)  9.932080e+00  0.0587240255  169.13146372  0.000000e+00
 station     -6.911515e-02  0.0065601988  -10.53552610   5.070594e-24
 tokyo       -2.846888e-05  0.0004214075   -0.06755664   9.461599e-01
 city         6.738630e-01  0.0360500253   18.69244166   2.121536e-62
 flood        2.795299e-02  0.0142681894    1.95911280   5.053884e-02
```

Station (-) and city (+) are statistically significant at the 0.1% level. It is verified that urban areas with good access to a railway station have a higher land price than other areas. We can see that flood is positively significant at the 10% level. This result suggests that influence from flood disaster, which is expected to be negative, is not appropriately reflected to land price.

VIF values are displayed by the following command:

```
> e_res$vif
                VIF
 station   1.367917
 tokyo     1.225594
 city      1.282930
 flood     1.208189
 sf4       1.167728
 sf9       1.017697
 sf12      1.142611
 sf31      1.084662
 sf33      1.032077
 sf45      1.035118
 sf32      1.095973
 sf26      1.012234
 sf6       1.059948
 sf20      1.016059
```

The following command displays error statistics, including residual standard error (residual_SE), adjusted R^2 (adjR2), log-likelihood (logLik), AIC, BIC, and degrees of freedom (DF):

```
> e_res$e
                    stat
 resid_SE      0.3542671
 adjR2         0.6987400
 logLik     -239.0702859
 AIC         510.1405718
 BIC         581.6981125
```

While we assumed ESF with binary **C**, which is popular in regional science, ESF with distance-based **C**, which is popular in ecology, is implemented as follows:

```
> meig    <- meigen( coords=coords )                    #Moran's eigenvectors (distance-based C)
> e_res   <- esf( y=y, x=x, meig=meig, fn = "r2" )
```

The distance-based ESF is often referred to as MEM or a principal coordinate neighborhood matrix (PCNM) (see Legendre and Legendre, 2012).

A major disadvantage of ESF is the computational cost. To cope with this problem, Murakami and Griffith (2017) develops a fact approximation. It is implemented by the following command:

```
> meig_f  <- meigen_f( coords = coords )
> e_res   <- esf( y = y, x = x, meig = meig_f, fn = "all" )
```

Here, all eigenvectors in meig_f are considered without selecting them by setting fn = "all". It is sufficient for thousands or more samples (Murakami and Griffith, 2017).

3.2.RE-ESF model

The RE-ESF model is formulated as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\gamma} \sim N\left(\mathbf{0}, \sigma_{\gamma}^{2}\boldsymbol{\Lambda}(\alpha)\right), \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^{2}\mathbf{I}).$$

Unlike ESF, $\boldsymbol{\gamma}$ is given by a vector of random coefficients: $\boldsymbol{\gamma} \sim N(\mathbf{0}, \sigma_{\gamma}^{2}\boldsymbol{\Lambda}(\alpha))$. $\boldsymbol{\Lambda}(\alpha)$ is a diagonal matrix whose elements are the eigenvalues, which are multiplied by α. σ_{γ}^{2} and α represent the variance and the scale of the spatially dependent component; large α implies global-scale spatial variation, while small α implies local variation. These parameters act as shrinkage parameters controlling variance inflation. The RE-ESF model is a low rank approximation of the Gaussian process model, which is popular in geostatistics.

The RE-ESF model is estimated by the following steps: (i) eigenvectors whose eigenvalue exceeds a threshold are extracted from **MCM**; (ii) parameters are estimated by the maximum likelihood (ML) method or the restricted maximum likelihood (REML) method. REML estimation is preferable because it accounts for the degrees of freedom lost by estimating the regression coefficients.

The REML estimation is implemented by the following command:

```
> meig    <- meigen( coords = coords )                  #Moran's eigenvectors (distance-based C)
> r_res   <- resf( y = y, x = x, meig = meig, method = "reml" )
```

ML is implemented by replacing method = "reml" with method = "ml".

Estimated coefficients are displayed as follows:

```
> r_res$b
                  Estimate          SE       t_value        p_value
 (Intercept)  9.9902998898  0.169833051  58.8242385  0.000000e+00
 station     -0.0792859163  0.009598674  -8.2600901   8.881784e-16
 tokyo       -0.0003715008  0.001795810  -0.2068709   8.361807e-01
 city         0.6857752216  0.036926493  18.5713608  0.000000e+00
 flood       -0.0043670379  0.014784271  -0.2953841   7.678025e-01
```

Just like the estimates for ESF, station (-) and city (+) are statistically significant, and tokyo is not. In contrast, unlike ESF, flood is not statistically significant. Because RE-ESF tends to outperform ESF in terms of the estimation accuracy of regression coefficients and their standard errors (Murakami and Griffith, 2015), the results of RE-ESF might be more reliable. Error statistics are extracted by the following command:

```
> r_res$e
                        stat
 resid_SE          0.3116825
 adjR2(cond)       0.7649824
 rlogLik        -262.9627231
 AIC             543.9254462
 BIC             584.1765628
```

where adjR2(cond) is the adjusted conditional R^2, and rlogLik is the restricted log-likelihood. rlogLik is replaced with loglik, which denotes log-likelihood, if method = "ml". It is important to note that, when REML is used, AIC and BIC are comparable only with models with the same explanatory variables. resf also returns the estimated shrinkage parameters as follows:

```
> r_res$s
                        par
 shrink_sf_SE     0.4337118
 shrink_sf_alpha  0.2449076
```

where shrink_sf_SE and shrink_sf_alpha are σ_γ and α, respectively. The standard error of the spatially dependent component (shrink_sf_SE = 0.4337118)[5] is greater than the residual standard error (resid_SE = 0.3116825). In other words, substantial spatial dependent variations, which are ignored if the linear regression model is estimated, are captured by $\mathbf{E}\boldsymbol{\gamma}$. shrink_sf_alpha is smaller than one. This implies that coefficients on each eigenvector are shrunk comparatively equally, irrespective of their corresponding eigenvalues. The resulting $\mathbf{E}\boldsymbol{\gamma}$ has local-scale spatial variations relative to $\mathbf{E}\boldsymbol{\gamma}$ with large shrink_sf_alpha.

resf performs the computationally efficient ML/REML estimation of Murakami and Griffith (2018). The command is as follows:

```
> meig_f  <- meigen_f( coords = coords )
> r_res    <- resf( y = y, x = x, meig = meig_f, method = "reml" )
```

4. Extended models

4.1.Spatially varying coefficient (SVC) model

Murakami et al. (2017) showed that a RE-ESF-based SVC modeling approach outperforms the geographically weighted regression (GWR) approach that is a standard approach for SVC modeling, in terms of coefficient estimation accuracy and computational time.

The SVC model is formulated as follows:

$$\mathbf{y} = \sum_k \mathbf{x}_k \otimes \boldsymbol{\beta}_k + \mathbf{E}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\gamma} \sim N\left(\mathbf{0}, \sigma_{\boldsymbol{\gamma}}^2 \boldsymbol{\Lambda}(\alpha)\right), \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}),$$

$$\boldsymbol{\beta}_k = \beta_{k,0} 1 + \mathbf{E}\boldsymbol{\gamma}_k, \qquad \boldsymbol{\gamma}_k \sim N\left(\mathbf{0}, \sigma_{\boldsymbol{\gamma},k}^2 \boldsymbol{\Lambda}(\alpha_k)\right),$$

where $\boldsymbol{\beta}_k$ is the vector of SVCs on the kth explanatory variables that , $\mathbf{x}_k$. $\boldsymbol{\beta}_k$ consists of the constant component, $\beta_{k,0}\mathbf{1}$, and the spatially varying component, $\mathbf{E}\boldsymbol{\gamma}_k$. The latter is modeled by Moran's eigenvectors, $\mathbf{E}$, and their random coefficients, $\boldsymbol{\gamma}_k \sim N(\mathbf{0}, \sigma_{\gamma,k}{}^2\boldsymbol{\Lambda}(\alpha_k))$. $\boldsymbol{\Lambda}(\alpha_k)$ is a diagonal matrix whose elements are the eigenvalues, which are multiplied by α_k. $\sigma_{\gamma,k}{}^2$ denotes the variance of the spatially dependent component, $\mathbf{E}\boldsymbol{\gamma}_k$, whereas α_k denotes the spatial scale of the component; large/small α_k implies global/local-scale spatial variation explained by $\mathbf{E}\boldsymbol{\gamma}_k$. These parameters act as shrinkage parameters controlling variance inflation. An interesting point is that, unlike GWR, the RE-ESF-based approach estimates the spatial scale of each SVC using α_k. Furthermore, the RE-ESF

[5] The following relationship holds: $Var[\mathbf{E}\boldsymbol{\gamma}]= \mathbf{E}\boldsymbol{\gamma}\boldsymbol{\gamma}'\,\mathbf{E}' = \sigma_\gamma{}^2\mathbf{E}\boldsymbol{\Lambda}(\alpha)\,\mathbf{E}' = \sigma_\gamma^2 \hat{\mathbf{C}}_M^\alpha$, where $\hat{\mathbf{C}}_M^\alpha$ is $\mathbf{MC}^\alpha\mathbf{M}$ approximated by the eigenvectors in $\mathbf{E}$. Hence, $\sigma_\gamma{}^2$ denotes the variance of the spatially dependent component.

approach enables us selecting SVCs to stabilize the estimates computationally efficiently.

In this tutorial, coefficients on tokyo, station, city, and flood are allowed to vary across geographical space whereas coefficients on px (x-coordinate) and py (y-coordinate) are not. The command for the SVC modeling is as

```
> xv      <- x[ , c( "tokyo", "station", "city", "flood" ) ]  #x with spatially varying coefficients
> xconst  <- x[ , c( "px", "py") ]                            #x with constant coefficients
> meig    <- meigen( coords = coords )                        #Moran's eigenvectors (distance-based C)
> rv_res  <- resf_vc( y = y, x = xv, xconst = xconst, meig = meig )
```

To stabilize the estimates, the resf_vc function selects SVCs by a Bayesian information criterion (BIC) minimization by default (penalty = "bic"). The selection can be replaced with an Akaike information criterion (AIC) minimization-based selection by specifying penalty = "aic" in the resf_vc function.

The constant coefficient estimates for px and py are returned by the following command:

```
> rv_res$b
          Estimate           SE      t_value      p_value
px   -1.164779e-06  4.476629e-06  -0.2601911   0.79492527
py    7.977915e-06  4.447399e-06   1.7938385   0.07401748
```

With regard to the SVCs, the BIC-based SVC selection result are displayed as follows:

```
> rv_res$vc
                      (Intercept)   tokyo   station   city   flood
varying coefficients            1       0         1      1       1
```

The result shows that coefficients on tokyo are estimated constant, and coefficients on the others are estimated to have spatial variations. Estimated SVCs and their *p*-values are displayed by the following command:

```
> rv_res$b_vc[ 1:6, ]
   (Intercept)       tokyo      station       city        flood
1    9.875385  -0.00513785  -0.06311678  0.5690735  0.006637360
2   10.278009  -0.00513785  -0.11503321  0.8255947  0.005446833
3   10.173544  -0.00513785  -0.10025270  0.7743310  0.006120595
```

```
4  10.138267  -0.00513785  -0.09395701   0.7445319   0.006262945
5  10.207279  -0.00513785  -0.10122246   0.5212322  -0.058020901
6  10.258219  -0.00513785  -0.08688370   0.5006614  -0.059386765

> rv_res$p_vc[ 1:6, ]
   (Intercept)      tokyo      station         city      flood
1            0  0.2354304  0.288107321 1.324063e-06 0.76500129
2            0  0.2354304  0.006000605 6.344747e-11 0.79813709
3            0  0.2354304  0.012536479 6.735013e-10 0.77200809
4            0  0.2354304  0.019306686 3.431317e-10 0.76597759
5            0  0.2354304  0.058577563 5.667240e-05 0.04008828
6            0  0.2354304  0.124107522 1.405513e-03 0.15243568
```

Coefficients on tokyo are constant as estimated. The estimated SVCs can be summarized as follows (coefficients on tokyo are omitted):

```
> summary( rv_res$b_vc[ , -2 ] )
          (Intercept)              station                  city                 flood
  Min.   :  8.909   Min.   :-0.21020   Min.   :-0.02115   Min.   :-0.066797
  1st Qu.:  9.831   1st Qu.:-0.15448   1st Qu.: 0.57226   1st Qu.:-0.049578
  Median : 10.062   Median :-0.12184   Median : 0.68319   Median :-0.013046
  Mean   : 10.061   Mean   :-0.11572   Mean   : 0.67039   Mean   :-0.021668
  3rd Qu.: 10.242   3rd Qu.:-0.07764   3rd Qu.: 0.81286   3rd Qu.: 0.003591
  Max.   : 10.946   Max.   : 0.06522   Max.   : 1.06872   Max.   : 0.009442

> summary( rv_res$p_vc[ , -2 ] )
          (Intercept)              station                  city                 flood
  Min.   :        0   Min.   :0.000001   Min.   :0.0000000   Min.   :0.003934
  1st Qu.:        0   1st Qu.:0.001426   1st Qu.:0.0000001   1st Qu.:0.086623
  Median :        0   Median :0.010549   Median :0.0000068   Median :0.585556
  Mean   :        0   Mean   :0.123792   Mean   :0.0171369   Mean   :0.495177
  3rd Qu.:        0   3rd Qu.:0.175201   3rd Qu.:0.0006345   3rd Qu.:0.853780
  Max.   :        0   Max.   :0.945239   Max.   :0.9582583   Max.   :0.995845
```

The result suggests that the spatially varying intercept and SVCs on city are positively significant across the target area. station is negatively significant in many sample sites, and flood is statistically insignificant in most sample sites.

Figure 2 displays the estimated coefficients and their statistical significance. Estimated SVCs on station demonstrate that the distance to a railway station has a significant influence on land price in areas along railways. SVCs on city are positively significant across the target area. SVCs on flood suggest that flood risk is negatively significant around Mito city, which is the prefectural capital. Mito city has a long history as a castle town. The negative sign on flood might be because Mito city has adapted to flood disaster in its long history.

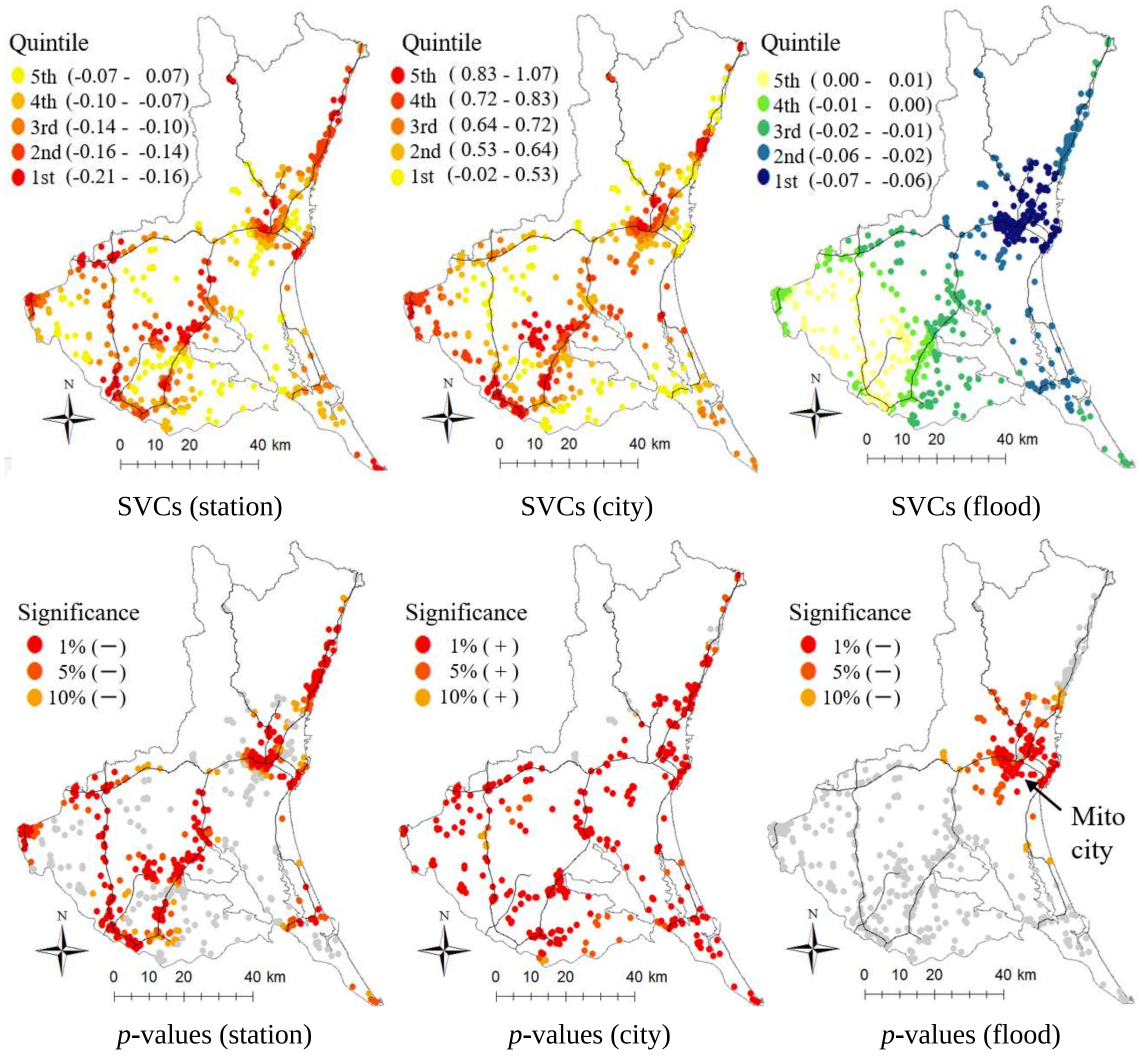

Figure 2. Estimated SVCs and their *p*-values (the spatially varying intercept is omitted)

Just like resf, resf_vc returns shrinkage parameter estimates for SVCs. In our case, the estimates are as follows:

```
> rv_res$s
                 (Intercept)       station        city       flood
 Shrink_sf_SE      0.4562311  0.0820578615  0.30293756  0.04153263
 Shrink_sf_alpha   0.1472938  0.0001043149  0.04450748  1.59444026
```

Shrink_sf_SE summarizes the estimated standard errors, $\sigma_{\gamma,k}$, for each SVC, and Shrink_sf_alpha summarizes the estimated α_k parameters. Large α_k values imply strong shrinkage for local variations. For example, SVCs on flood have a global map pattern due to the large α_k value while SVCs on station have a local pattern due to the small α_k value. Thus, the α_k parameter controls the spatial scale of the *k*-th SVCs.

Error statistics for the SVC model are displayed by the following command although logLik and AIC are reference values because we apply the REML estimation:

```
> r_res$e
                     stat
 resid_SE       0.2637017
 adjR2(cond)    0.8312410
 rlogLik     -230.4469132
 AIC          482.8938264
 BIC          532.0896357
```

4.2.Spatially filtered unconditional quantile regression (SF-UQR)

While the conventional conditional quantile regression (CQR) estimates the influence of x_k on the τ-th "conditional" quantile of y, $q_\tau(y|x_k)$, the unconditional quantile regression (UQR; Firpo et al., 2009) estimates the influence of x_k on the "unconditional" quantile of y, $q_\tau(y)$.

Suppose y and x_k represent land price and accessibility, respectively. UQR estimates the influence of accessibility on land price in each price range. This interpretation does not hold for CQR, because it quantifies the influence of accessibility on land prices conditional on x_k (see, **Figure 3**). Thus, UQR coefficients are more interpretable than CQR coefficients.

In this context, Murakami and Seya (2017) developed the spatial filter UQR (SF-UQR). The SF-UQR model is formulated as follows:

$$\mathbf{r}_\tau = \mathbf{X}\boldsymbol{\beta}_\tau + \mathbf{E}\boldsymbol{\gamma}_\tau + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\gamma} \sim N\left(\mathbf{0}, \sigma^2_{\gamma,\tau}\boldsymbol{\Lambda}(\alpha_\tau)\right), \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2_\tau \mathbf{I}),$$

where $\mathbf{r}_\tau$ is a vector whose *i*-th element equals the re-centered influence function (RIF) for the *i*-th explained variable, y_i. The SF-UQR is a UQR considering spatial dependence.

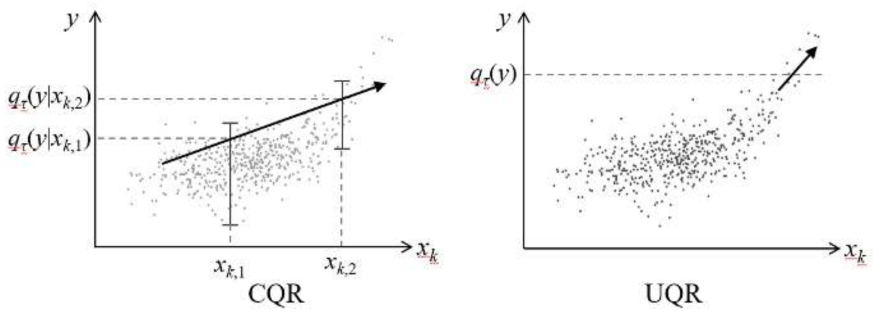

Figure 3: CQR and UQR coefficients. Allows illustrate their coefficients on the 0.9 quantile. CQR coefficient equals $\partial q_\tau(y|x_k)/\partial x_k$ whereas UQR coefficient equals $\partial q_\tau(y)/\partial x_k$.

The spmoran package provides the resf_qr function to estimate the SF-UQR model. The command is as follows:

```
> qr_res    <- resf_qr( y = y, x = x, meig = meig, boot = T )
```

If boot = T, a semiparametric bootstrapping is performed to estimate the standard errors of UQR coefficients, and these are not calculated if boot = F. This function returns parameters estimated at 0.1, 0.2, ..., 0.9 quantiles by default. The quantile(s) can be specified by using an argument tau; for example, parameters at the 0.22 quantile are estimated by assigning tau = 0.22.

The computational complexity for the bootstrap iterations does not depend on the sample size, *N* (see, Murakmai and Griffith, 2018), but it depends on the number of eigenpairs, *L*, which grows as *N* increases. Hence, for large samples, it is useful to restrict *L* as follows:

```
> meig     <- meig( coords, enum = 200 )
> qr_res   <- resf_qr( y = y, x = x, meig = meig, boot = T )
```

For very large *N*, which prohibits the eigen-decomposition, the following eigen-approximation would be useful:

```
> meig     <- meig_f( coords )    #It approximates the first 200 eignen-pairs by default
```

UQR coefficients estimated by the resf_qr function can be visualized by the plot_qr function. The commands to plot estimated coefficients for the first five explanatory variables are as follows:

```
> plot_qr( qr_res, 1 )
> plot_qr( qr_res, 2 )
> plot_qr( qr_res, 3 )
> plot_qr( qr_res, 4 )
> plot_qr( qr_res, 5 )
```

The numbers 1 to 5 specify which regression coefficients are plotted (1: intercept). The resulting plots are as follows:

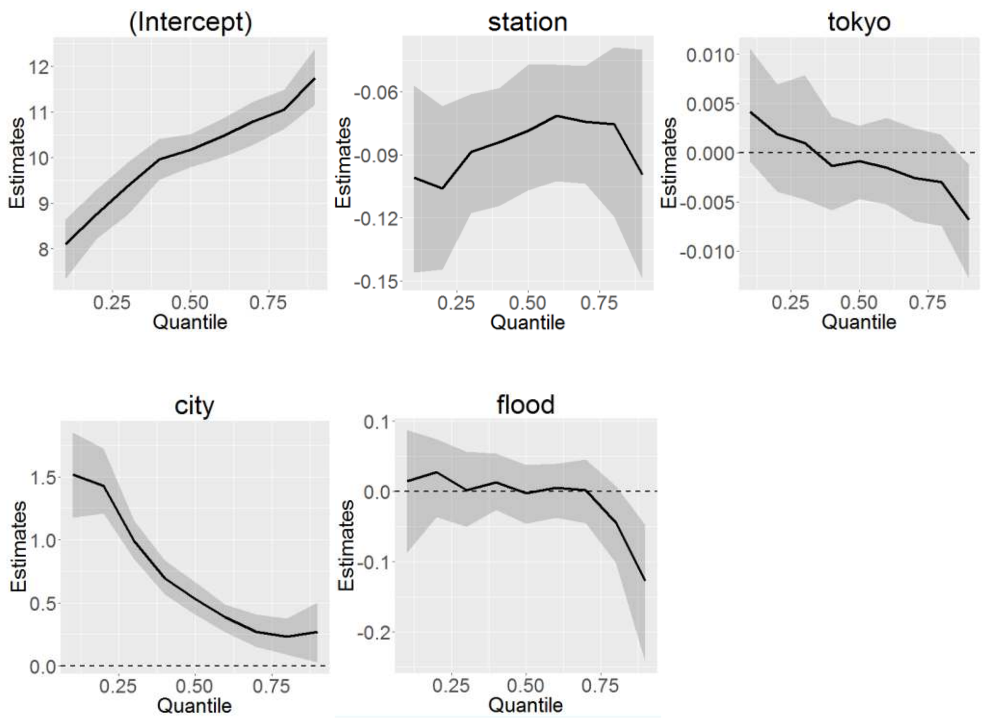

Figure 4. Outputs from the plot_qr function (regression coefficients). Solid lines are coefficient estimates and gray areas are their 95% confidential intervals.

On the other hand, the standard errors for the residual spatial dependent component (shrink_sf_SE) are plotted by assigning pnum = 1 and par = "s", while the scale (degree) parameters for the component (shrink_sf_alpha) are plotted by assigning pnum = 2 and par = "s". The commands and the outcomes are as follows:

```
> plot( qr_res, 1, “s” )
> plot( qr_res, 2, “s” )
```

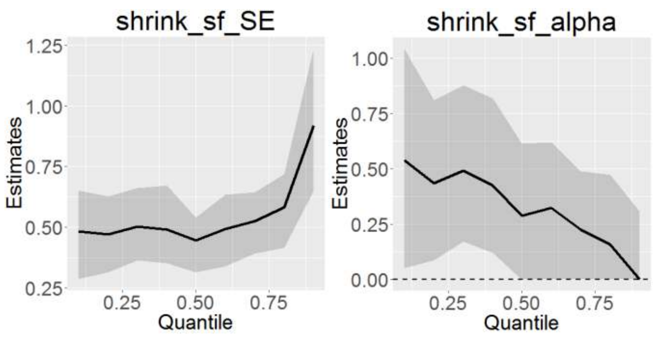

Figure 5. Outputs from the plot_qr function (shrinkage (variance) parameters). Solid lines are coefficient estimates and gray areas are their 95 % confidential intervals.

Parameter estimates are displayed by the following commands:

```
> res$b
> res$s
```

When boot = T, parameter estimates, lower and upper bounds for their 95% confidential intervals, and p-values are returned by the following command:

```
> res$B
> res$S
```

Error statistics, including the residual standard error and the adjusted quasi conditional R^2, are displayed as follows:

```
> res$e
                    tau=0.1    tau=0.2    tau=0.3    …    tau=0.9
 resid_SE           0.93164    0.67819    0.58475    …     1.0025
 quasi_adjR2(cond)  0.43749    0.57931    0.57318    …     0.4258
```

5. Spatial interpolation

Despite ESF is well-known as an explanatory tool in ecology, it is less clear how to apply it for (out-of-sample) spatial prediction, which is a representative objective of explanatory spatial data analysis (Note: in ecology, ESF is called Moran's eigenvector maps (MEMs)). To overcome this limitation, the spmoran package provides functions for ESF/RE-ESF-based spatial interpolation based on a minimization of expected error. The Nystrom extension, which is an eigen-approximation technique that is popular in a machine learning literature, is used for the expected error minimization.

Note that RE-ESF can be viewed as a low rank approximation of the Gaussian process (GP) model whose spatial prediction is well-known as kriging (see, Murakami and Griffith, 2015). ESF is also a special case. In other words, because the ESF and RE-ESF models are approximations, their spatial predictions might be less accurate relative to kriging. Instead, they are available for large samples as I will explain in section 6.

In this tutorial, the land price data is randomly divided into two, and one is assumed as data at observed sites (dd), and another is assumed as data at unobserved sites (d0).

```
> samp<-sample( length( d[, 1] ), 300 ) # Random sampling

> dd          <- d[ samp, ]              # Data at observed sites
> coords      <- dd[ , 1:2 ]
> y           <- dd[ ,3 ]
> x           <- dd[ ,4:7]
> xconst      <- dd[ ,1:2]

> md          <- d[-samp, ]              # Data at unobserved sites
> coords0     <- md[ , 1:2 ]
> y0          <- md[ ,3 ]
> x0          <- md[ ,4:7 ]
> xconst0     <- md[ ,1:2 ]
```

Before the prediction, Moran's eigenvectors must be evaluated both for observed and unobserved sites. meigen or meigen_f is used for the former while meigen0 is used for the latter:

```
> meig        <- meigen( coords = coords )
> meig0       <- meigen0( meig = meig, coords0 = coords0 )
```

For ESF-based spatial interpolation, the ESF model is estimated as before. Then, data at unobserved sites are predicted using the predict0 function. The command is as follows:

```
> mod_e      <-esf( y = y, x = x, meig = meig )                    ## Model estimation
> pred_e     <-predict0( mod = mod_e, x0 = x0, meig0 = meig0 )   ## Spatial prediction
> pred_e[1:6,]
              pred          xb          sf
1         10.42826   10.430779   -0.00252353
2         10.76887   10.598882    0.16999289
3         10.01560    9.930919    0.08467572
4         10.22738   10.333108   -0.10572316
5         10.43264   10.458080   -0.02543832
6         10.41613   10.375074    0.04106011
```

The outputs are the predicted explained variables (pred; **y**), trend (xb; **Xβ**), and residual spatial component (sf; **E**γ). RE-ESF-based spatial interpolation is implemented in the same way:

```
> mod_re     <-resf( y = y, x = x, meig = meig )                   ## Model estimation
> pred_re    <-predict0( mod = mod_re, x0 = x0, meig0=meig0 )    ## Spatial prediction
> pred_re[1:6,]
              pred          xb          sf
1         10.43984   10.438097    0.00174815
2         10.79472   10.629825    0.16489057
3         10.02705    9.941318    0.08573132
4         10.21008   10.318316   -0.10823100
5         10.41722   10.455091   -0.03787292
6         10.39429   10.364245    0.03004664
```

If x0 is not provided, the predict0 function interpolates the spatial component only.

```
> sf_re      <-predict0( mod = mod_re, meig0=meig0 )
> sf_re[1:6]
[1]  0.001748157  0.164890570  0.085731320  -0.108231008  -0.037872927  0.030046643
```

The SVC model is also available for spatial prediction. The predict0_vc function predicts, explanatory variables and SVCs. The command is as follows:

```
> rv_res<-resf_vc( y = y, x = x, xconst = xconst, meig = meig )          ## Model estimation
> pred_vc    <- predict0_vc( mod = rv_res, x0 = x0, xconst0 = xconst0, meig0 = meig0 )
                                                                          ## Spatial prediction
```

The predicted explained variables are returned as

```
> pred_vc$pred[ , 1:6 ]
              pred          xb            sf
1         10.44104   10.420895    0.02014816
2         10.77090   10.579498    0.19139730
3         10.01037    9.898628    0.11174645
4         10.20561   10.375823   -0.17020794
5         10.41315   10.496904   -0.08375796
6         10.40018   10.431076   -0.03090005
```

The outputs are the predicted explained variables (pred; $\mathbf{y}$), trend (xb; $\sum_k \mathbf{x}_k \otimes \boldsymbol{\beta}_k$), and residual spatial component (sf; $\mathbf{E}\boldsymbol{\gamma}$). The predicted SVCs are returned as follows:

```
> pred_vc$b_vc[ 1:6, ]
  (Intercept)         tokyo        station         city          flood
 1   10.07619   -0.00513785    -0.06439647    0.5620327   -0.003150771
 2   10.42358   -0.00513785    -0.10874117    0.8603790    0.006905524
 3   10.33535   -0.00513785    -0.09560769    0.8012729    0.000362355
 4   10.30509   -0.00513785    -0.09011763    0.7661289   -0.004951810
 5   10.27922   -0.00513785    -0.10336049    0.5352906    0.002864006
 6   10.35502   -0.00513785    -0.08970204    0.5099307   -0.041507693
```

The standard errors, *t*-values, and *p*-values of the SVCs are returned by pred_vc$bse_vc, pred_vc$t_vc, and pred_vc$p_vc, respectively. It is worth noting that SVCs are predicted even if x0 and xconst0 are missing. The command is as follows:

```
> pred_vc    <- predict0_vc( mod = rv_res, meig0 = meig0 )
> pred_vc$b_vc[ 1:6, ]
```

6. Tips for fast computation

6.1.Eigen-decomposition

As discussed, meigen_f performs a fact eigen-approximation, and extracts the first 200 eigenvectors by default. The computation is further accelerated by reducing number of approximated eigenvectors. It is achieved by setting enum by a positive integer less than 200. For example, in the case with 5000 samples and enum = 200 (default), 100, and 50, computational times are as follows:

```
> coords_test                   <- cbind( rnorm( 5000 ), rnorm( 5000 ) )
```

```
-----------------CP time (without approximation) -----------------
> system.time( meig_test        <- meigen( coords = coords_test ) )
     user     system     elapsed
  242.28        1.44      243.79
```

```
-----------------CP time (with approximation) ----------------------
> system.time( meig_test200  <- meigen_f( coords = coords_test )
     user     system     elapsed
     0.37       0.00         0.38
> system.time( meig_test100  <- meigen_f( coords = coords_test, enum = 100 ) )
     user     system     elapsed
     0.15       0.00         0.16
> system.time( meig_test50    <- meigen_f( coords = coords_test, enum = 50 ) )
     user     system     elapsed
     0.08       0.00         0.08
```

Figure 3 maps the calculated 1st, 10th, and 100th eigenvectors. It is important to note that, while approximated and exact eigenvectors can have different map patterns respectively, both of them describe patterns in similar spatial scales. In other words, in both cases, 1st eigenvectors describe global map patterns, 10th medium-scale patterns, and 100th local patterns.

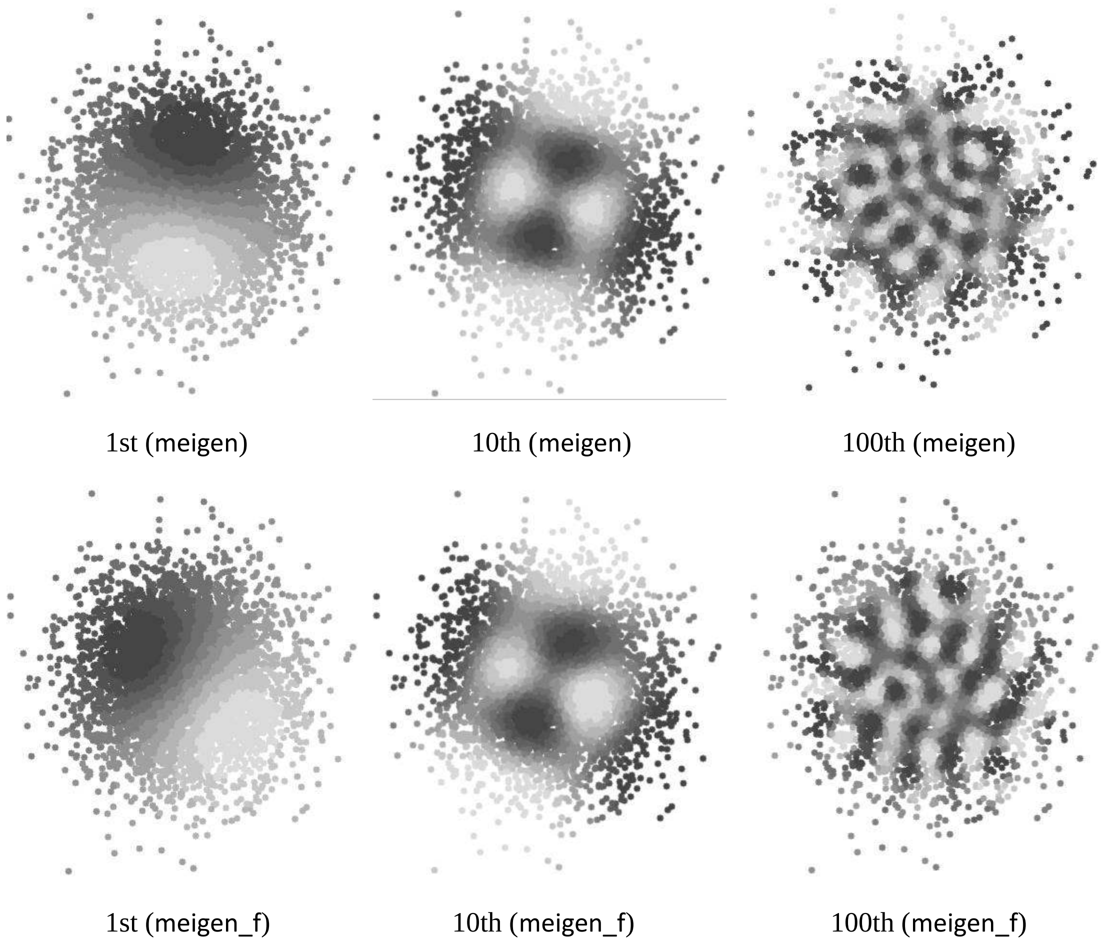

Figure 3. The 1st, 10th, and 100th eigenvectors extracted from meigen and meigen_f

6.2.Parameter estimation

The basic ESF model is estimated computationally efficiently by setting fn = "all" in the function esf. The RE-ESF model is estimated by small computational cost by the function resf, by default.

In the current version, the RE-ESF-based SVC model is also estimated computationally efficiently by the resf_vc function. The SVC modeling requires an inversion of a (K_{const} + K + LK) $\times$ (K_{const} + K + LK) matrix. K_{const} and K are the number of explanatory variables in xconst and x, respectively. L is the number of eigen-pairs that is below enum = 200 by default. To avoid slow computation, resf_vc constraints (K_{const} + K + LK) to not to exceed sizelimit whose default value is 2,000 (in this case, an inversion of a 2,000 × 2,000 matrix is the possible heaviest computation in the estimation step). Note that sizelimit is effective only when (K_{const} + K + LK) < sizelimit; roughly speaking, it becomes true when SVC is more than 10.

The following code is an example of the fast SVC model estimation:

```
> meig    <- meigen_f( coords = coords )                    # fast approximation
> rv_res  <- resf_vc( y = y, x = xv, xconst = xconst, meig = meig )
```

The SF-UQR model requires a bootstrapping to estimate confidential intervals for the coefficients. However, computational cost for the iteration does not dependent on sample size, but only on the number of eigenvectors in meig (see, Murakami and Seya, 2017). That is, the SF-UQR is applicable to large data if only meig is defined just as mentioned above.

7. Future directions

I will increase functions relating Moran's eigenvector-based spatial regression modeling gradually.

References

- Dray, S., Legendre, P., and Peres-Neto, P.R. (2006) Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM). *Ecological Modelling*, 196 (3), 483-493.
- Firpo, S., Fortin, N.M., and Lemieux, T. (2009) Unconditional quantile regressions. *Econometrica*, 77 (3), 953-973.
- Griffith, D.A. (2003) *Spatial autocorrelation and spatial filtering: gaining understanding through theory and scientific visualization*. Springer Science & Business Media.
- Griffith, D., & Chun, Y. (2014). Spatial autocorrelation and spatial filtering. In: *Handbook of regional science*, pp. 1477-1507. Springer.
- Legendre, P. and Legendre, L.F. (2012) *Numerical Ecology*. Elsevier.
- Murakami, D. and Griffith, D.A. (2015) Random effects specifications in eigenvector spatial filtering: a simulation study. *Journal of Geographical Systems*, 17 (4), 311-331.
- Murakami, D. and Griffith, D.A. (2018) Eigenvector spatial filtering for large data sets: fixed and random effects approaches. *Geographical Analysis*, doi: 10.1111/gean.12156.
- Murakami, D. and Seya, H. (2017) Spatially filtered unconditional quantile regression, *Arxiv*.1706.07705.
- Murakami, D., Yoshida, T., Seya, H., Griffith, D.A., and Yamagata, Y. (2017) A Moran coefficient-based mixed effects approach to investigate spatially varying relationships. *Spatial Statistics*, 19, 68-89.
- Tiefelsdorf, M., and Griffith, D. A. (2007). Semiparametric filtering of spatial autocorrelation: the eigenvector approach. *Environment and Planning A*, 39 (5), 1193-1221.