spmodel: Spatial Statistical Modeling and Prediction in **R** – Supplementary Material

Michael Dumelle [1] [*], Matt Higham [2] , Jay M. Ver Hoef [3]

1 United States Environmental Protection Agency, 200 SW 35th St, Corvallis, OR, 97333
2 St. Lawrence University Department of Math, Computer Science, and Statistics
3 National Oceanic and Atmospheric Administration Alaska Fisheries Science Center, Marine Mammal Laboratory, Seattle, WA, 98115

* Corresponding author: Dumelle.Michael@epa.gov

Abstract

`spmodel` is an **R** package used to fit, summarize, and predict for a variety spatial statistical models. Parameters are estimated using various methods. Additional modeling features include anisotropy, random effects, partition factors, big data approaches, and more. Model-fit statistics are used to summarize, visualize, and compare models. Predictions at unobserved locations are readily obtainable. This manuscript corresponds to `spmodel` version 0.1.0.

Introduction

This vignette covers technical details regarding the functions in `spmodel` that perform computations. We first provide a notation guide and then describe relevant details for each function.

If you use `spmodel` in a formal publication or report, please cite it. Citing `spmodel` lets us devote more resources to it in the future. To view the `spmodel` citation, run

```
citation(package = "spmodel")
```

```
#>
#> To cite spmodel in publications use:
#>
#>   Michael Dumelle, Matt Higham, and Jay M. Ver Hoef (2022). spmodel:
#>   Spatial Statistical Modeling and Prediction. R package version 0.1.0.
#>
#> A BibTeX entry for LaTeX users is
#>
#>   @Manual{,
#>     title = {spmodel: Spatial Statistical Modeling and Prediction},
#>     author = {Michael Dumelle and Matt Higham and Jay M. {Ver Hoef}},
#>     year = {2022},
#>     note = {R package version 0.1.0},
#>   }
```

In addition to this document on the technical details of `spmodel`, there are three other vignettes:

- An overview of basic features in `spmodel`: `vignette("basics", "spmodel")`
- A detailed guide to `spmodel`: `vignette("guide", "spmodel")`

Notation Guide

n = Sample size

$\mathbf{y}$ = Response vector

$\boldsymbol{\beta}$ = Fixed effect parameter vector

$\mathbf{X}$ = Design matrix of known explanatory variables (covariates)

p = The number of linearly independent columns in $\mathbf{X}$

$\mathbf{Z}$ = Design matrix of known random effect variables

$\boldsymbol{\theta}$ = Covariance parameter vector

$\boldsymbol{\Sigma}$ = Covariance matrix evaluated at $\boldsymbol{\theta}$

$\boldsymbol{\Sigma}^{-1}$ = The inverse of $\boldsymbol{\Sigma}$

$\boldsymbol{\Sigma}^{1/2}$ = The square root of $\boldsymbol{\Sigma}$

$\boldsymbol{\Sigma}^{-1/2}$ = The inverse of $\boldsymbol{\Sigma}^{1/2}$

$\boldsymbol{\Theta}$ = General parameter vector

$\ell(\boldsymbol{\Theta})$ = Log-likelihood evaluated at $\boldsymbol{\Theta}$

$\boldsymbol{\tau}$ = Spatial (dependent) random error

$\mathbf{A}^* = \boldsymbol{\Sigma}^{-1/2}\mathbf{A}$ for a general matrix $\mathbf{A}$ (this is known as whitening $\mathbf{A}$)

A hat indicates the parameters are estimated (i.e., $\hat{\boldsymbol{\beta}}$) or evaluated at a relevant estimated parameter vector (e.g., $\hat{\boldsymbol{\Sigma}}$ is evaluated at $\hat{\boldsymbol{\theta}}$). When $\ell(\hat{\boldsymbol{\Theta}})$ is written, it means the log-likelihood evaluated at its maximum, $\hat{\boldsymbol{\Theta}}$. When the covariance matrix of $\mathbf{A}$ is $\boldsymbol{\Sigma}$, we say $\mathbf{A}^*$ "whitens" $\mathbf{A}$ because

$$\text{Cov}(\mathbf{A}^*) = \text{Cov}(\boldsymbol{\Sigma}^{-1/2}\mathbf{A}) = \boldsymbol{\Sigma}^{-1/2}\text{Cov}(\mathbf{A})\boldsymbol{\Sigma}^{-1/2} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1/2} = (\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}^{1/2})(\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{-1/2})$$

See Section for a discussion on obtaining $\boldsymbol{\Sigma}^{1/2}$.

Additional notation is used in Section (`predict()`):

$\mathbf{y}_o$ = Observed response vector

$\mathbf{y}_u$ = Unobserved response vector

$\mathbf{X}_o$ = Design matrix of known explanatory variables at observed response variable locations

$\mathbf{X}_u$ = Design matrix of known explanatory variables at unobserved response variable locations

$\boldsymbol{\Sigma}_o$ = Covariance matrix of $\mathbf{y}_o$ evaluated at $\boldsymbol{\theta}$

$\boldsymbol{\Sigma}_u$ = Covariance matrix of $\mathbf{y}_u$ evaluated at $\boldsymbol{\theta}$

$\boldsymbol{\Sigma}_{uo}$ = A matrix of covariances between $\mathbf{y}_u$ and $\mathbf{y}_o$ evaluated at $\boldsymbol{\theta}$

`AIC()` and `AICc()`

The `AIC()` and `AICc()` functions in `spmodel` are defined for restricted maximum likelihood and maximum likelihood estimation, which maximize a likelihood. They

follow [1], defining spatial AIC and AICc as

$$\text{AIC} = -2\ell(\hat{\boldsymbol{\Theta}}) + 2(|\hat{\boldsymbol{\Theta}}|)$$
$$\text{AICc} = -2\ell(\hat{\boldsymbol{\Theta}}) + 2n(|\hat{\boldsymbol{\Theta}}|)/(n - |\hat{\boldsymbol{\Theta}}| - 1),$$

where $|\hat{\boldsymbol{\Theta}}|$ is the cardinality of $\hat{\boldsymbol{\Theta}}$. For restricted maximum likelihood, $\hat{\boldsymbol{\Theta}} \equiv \{\hat{\boldsymbol{\theta}}\}$. For maximum likelihood, $\hat{\boldsymbol{\Theta}} \equiv \{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}\}$ The discrepancy arises because restricted maximum likelihood integrates the fixed effects out of the likelihood, and so the likelihood does not depend on $\boldsymbol{\beta}$.

AIC comparisons between a model fit using restricted maximum likelihood and a model fit using maximum likelihood are meaningless, as the models are fit with different likelihoods. AIC comparisons between models fit using restricted maximum likelihood are only valid when the models have the same fixed effect structure. In contrast, AIC comparisons between models fit using maximum likelihood are valid when the models have different fixed effect structures.

anova()

Test statistics from `anova()` are formed using the general linear hypothesis test. Let $\mathbf{L}$ be an $l \times p$ contrast matrix and l_0 be an $l \times 1$ vector. The null hypothesis is that $\mathbf{L}\hat{\boldsymbol{\beta}} = l_0$ and the alternative hypothesis is that $\mathbf{L}\hat{\boldsymbol{\beta}} \neq l_0$. Usually, l_0 is the zero vector (in `spmodel`, this is assumed). The test statistic is denoted $Chi2$ and is given by

$$Chi2 = [(\mathbf{L}\hat{\boldsymbol{\beta}} - l_0)^\top (\mathbf{L}(\mathbf{X}^\top \hat{\boldsymbol{\Sigma}} \mathbf{X})^{-1} \mathbf{L}^\top)^{-1} (\mathbf{L}\hat{\boldsymbol{\beta}} - l_0)]$$

By default, $\mathbf{L}$ is chosen such that each variable in the data used to fit the model is tested marginally (i.e., controlling for the other variables) against $l_0 = \mathbf{0}$. If this default is not desired, the `Terms` and `L` arguments can be used to pass user-defined $\mathbf{L}$ matrices to `anova()`. They must be constructed in such a way that $l_0 = \mathbf{0}$.

It is notoriously difficult to determine appropriate p-values for linear mixed models based on the general linear hypothesis test. lme4, for example, does not report p-values by default. A few reasons why obtaining p-values is so challenging:

- The first (and often most important) challenge is that when estimating $\boldsymbol{\theta}$ using a finite sample, it is usually not clear what the null distribution of $Chi2$ is. In certain cases such as ordinary least squares regression or some experimental designs (e.g., blocked design, split plot design, etc.), $Chi2/rank(\mathbf{L})$ is F-distributed with known numerator and denominator degrees of freedom. But outside of these well-studied cases, no general results exist.
- The second challenge is that the standard error of $Chi2$ does not account for the uncertainty in $\hat{\boldsymbol{\theta}}$. For some approaches to addressing this problem, see [2], [3], [4], and [5].
- The third challenge is in determining denominator degrees of freedom. Again, in some cases, these are known – but this is not true in general. For some approaches to addressing this problem, see [6], [7], [8], [5], [9], [10], and [11].

For these reasons, `spmodel` uses an asymptotic (i.e., large sample) Chi-squared test when calculating p-values using `anova()`. This approach addresses the three points above by assuming that with a large enough sample size:

- $Chi2$ is asymptotically Chi-squared (under certain conditions) with $rank(\mathbf{L})$ degrees of freedom when the null hypothesis is true.
- The uncertainty from estimating $\hat{\boldsymbol{\theta}}$ is small enough to be safely ignored.

Because the approximation is asymptotic, degree of freedom adjustments can be ignored (it is also worth noting that an F distribution with infinite denominator degrees of freedom is a Chi-squared distribution scaled by $rank(\mathbf{L})$. This asymptotic approximation implies these p-values are likely unreliable with small samples.

Note that when comparing full and reduced models, the general linear hypothesis test is analogous to an extra sum of (whitened) squares approach [12].

A second approach to determining p-values is a likelihood ratio test. Let $\ell(\hat{\boldsymbol{\Theta}})$ be the log-likelihood for some full model and $\ell(\hat{\boldsymbol{\Theta}}_0)$ be the log-likelihood for some reduced model. For the likelihood ratio test to be valid, the reduced model must be nested in the full model, which means that $\ell(\hat{\boldsymbol{\Theta}}_0)$ is obtained by fixing some parameters in $\boldsymbol{\Theta}$. When the likelihood ratio test is valid, $X^2 = 2\ell(\hat{\boldsymbol{\Theta}}) - 2\ell(\hat{\boldsymbol{\Theta}}_0)$ is asymptotically Chi-squared with degrees of freedom equal to the difference in estimated parameters between the full and reduced models.

For restricted maximum likelihood estimation, likelihood ratio tests can only be used to compare nested models with the same explanatory variables. To use likelihood ratio tests for comparing different explanatory variable structures, parameters must be estimated using maximum likelihood estimation. When using likelihood ratio tests to assess the importance of parameters on the boundary of a parameter space (e.g., a variance parameter being zero), p-values tend to be too large [10,13–15].

`coef()`

`coef()` returns relevant coefficients based on the `type` argument. When `type = "fixed"` (the default), `coef()` returns

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{y}.$$

If the estimation method is restricted maximum likelihood or maximum likelihood, $\hat{\boldsymbol{\beta}}$ is known as the restricted maximum likelihood or maximum likelihood estimator of $\boldsymbol{\beta}$. If the estimation method is semivariogram weighted least squares or semivariogram composite likelihood, $\hat{\boldsymbol{\beta}}$ is known as the empirical generalized least squares estimator of $\boldsymbol{\beta}$. When `type = "spcov"`, the estimated spatial covariance parameters are returned (available for all estimation methods). When `type = "randcov"`, the estimated random effect variance parameters are returned (available for restricted maximum likelihood and maximum likelihood estimation).

`confint()`

`confint()` returns confidence intervals for estimated parameters. Currently, `confint()` only returns confidence intervals for $\boldsymbol{\beta}$. The $(1-\alpha)\%$ confidence interval for β_i is

$$\hat{\beta}_i \pm z^* \sqrt{(\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1}_{i,i}},$$

where $(\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1}_{i,i}$ is the ith diagonal element in $(\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1}$, $\Phi(z^*) = 1 - \alpha/2$, $\Phi(\cdot)$ is the standard normal (Gaussian) cumulative distribution function, and $\alpha = 1 -$ `level`, where `level` is an argument to `confint()`. The default for `level` is 0.95, which corresponds to a z^* of approximately 1.96.

`cooks.distance()`

Cook's distance measures the influence of an observation [16,17]. An influential observation has a large impact on the model fit. The vector of Cook's distances for the

spatial linear model is given by

$$\frac{\mathbf{e}_p^2}{p}\frac{diag(\mathbf{H}_s)}{1-diag(\mathbf{H}_s)}, \tag{1}$$

where $\mathbf{e}_p$ are the Pearson residuals and $diag(\mathbf{H}_s)$ is the diagonal of the spatial hat matrix, $\mathbf{H}_s \equiv \mathbf{X}^*(\mathbf{X}^{*\top}\mathbf{X}^*)^{-1}\mathbf{X}^{*\top}$ [18]. The larger the Cook's distance, the larger the influence.

To better understand the form in Equation 1, recall that the the non-spatial linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ assumes elements of $\boldsymbol{\epsilon}$ are independent and identically distributed (iid) with constant variance. In this context the vector of non-spatial Cook's distances is given by

$$\frac{\mathbf{e}_p^2}{p}\frac{diag(\mathbf{H})}{1-diag(\mathbf{H})},$$

where $diag(\mathbf{H})$ is the diagonal of the non-spatial hat matrix, $\mathbf{H} \equiv \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$. When the elements of $\boldsymbol{\epsilon}$ are not iid or do not have constant variance or both, the spatial Cook's distance cannot be calculated using $\mathbf{H}$. First the linear model must be whitened according to $\mathbf{y}^* = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\epsilon}^*$, where $\boldsymbol{\epsilon}^*$ is the whitened version of the sum of all random errors in the model. Then the spatial Cook's distance follows using the whitened version of $\mathbf{X}$, $\mathbf{X}^*$.

`deviance()`

The deviance of a fitted model is

$$\mathcal{D}_{\boldsymbol{\Theta}} = 2\ell(\boldsymbol{\Theta}_s) - 2\ell(\hat{\boldsymbol{\Theta}}),$$

where $\ell(\boldsymbol{\Theta}_s)$ is the log-likelihood of a "saturated" model that fits every observation perfectly. For normal (Gaussian) random errors,

$$\mathcal{D}_{\boldsymbol{\Theta}} = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

`esv()`

The empirical semivariogram is a moment-based estimate of the theoretical semivariogram. The empirical semivariogram quantifies half of the average squared difference in the response among observations in several distance classes. More formally, the empirical semivariogram is defined as

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|}\sum_{N(h)}(y_i - y_j)^2, \tag{2}$$

where $N(h)$ is the set of observations in $\mathbf{y}$ that are h distance units apart (distance classes) and $|N(h)|$ is the cardinality of $N(h)$ [19]. Often the set $N(h)$ contains observations that are $h \pm c$ apart, where c is some constant. This approach is known as "binning" the empirical semivariogram. The default in `spmodel` is to construct the semivariogram using 15 equally spaced bins where h is contained in $(0, h_{max}]$, and h_{max} is known as a "distance cutoff". Distance cutoffs are commonly used when constructing Equation 2 because there tend to be few pairs with large distances. The default in `spmodel` is to use a cutoff of half the maximum distance (hypotenuse) of the domain's bounding box.

The main purpose of the empirical semivariogram is its use in semivariogram weighted least squares estimation, though it can also be used as a visual diagnostic to assess the fit of a spatial covariance function.

`fitted()`

Fitted values can be obtained for the response, spatial random errors, and random effects. The fitted values for the response (`type = "fixed"`), denoted $\hat{\mathbf{y}}$, are given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}.$$

They are the estimated mean response given the set of explanatory variables for each observation.

Fitted values for spatial random errors (`type = "spcov"`) and random effects (`type = "randcov"`) are linked to best linear unbiased predictors from linear mixed model theory. Consider the standard random effects parameterization

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon},$$

where $\mathbf{Z}$ denotes the random effects design matrix, $\mathbf{u}$ denotes the random effects, and $\boldsymbol{\epsilon}$ denotes independent random error. [20] states that the best linear unbiased predictor (BLUP) of a single random effect $\mathbf{u}$, denoted $\hat{\mathbf{u}}$, is given by

$$\hat{\mathbf{u}} = \sigma_u^2 \mathbf{Z}^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}), \tag{3}$$

where σ_u^2 is the variance of $\mathbf{u}$.

[21] generalize this idea by showing that for a random variable $\boldsymbol{\alpha}$ in a linear model, the best linear unbiased predictor (based on the response, $\mathbf{y}$) of $\boldsymbol{\alpha}$, denoted $\hat{\boldsymbol{\alpha}}$, is given by

$$\hat{\boldsymbol{\alpha}} = \mathrm{E}(\boldsymbol{\alpha}) + \boldsymbol{\Sigma}_\alpha \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}), \tag{4}$$

where $\boldsymbol{\Sigma}_\alpha = \mathrm{Cov}(\boldsymbol{\alpha}, \mathbf{y})$. Evaluating Equation 4 at the plug-in (empirical) estimates of the covariance parameters yields the empirical best linear unbiased predictor (EBLUP) of $\boldsymbol{\alpha}$.

Recall that the spatial linear model with random effects is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\tau} + \boldsymbol{\epsilon},$$

Building from the result in Equation 4, we can find BLUPs for each random term in the spatial linear model ($\mathbf{u}$, $\boldsymbol{\tau}$, and $\boldsymbol{\epsilon}$). For example, the BLUP of $\mathbf{u}$ is found by noting that $\mathrm{E}(\mathbf{u}) = \mathbf{0}$ and

$$\boldsymbol{\Sigma}_u = \mathrm{Cov}(\mathbf{u}, \mathbf{y}) = \mathrm{Cov}(\mathbf{u}, \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\tau} + \boldsymbol{\epsilon}) = \mathrm{Cov}(\mathbf{u}, \mathbf{Z}\mathbf{u}) = \mathrm{Cov}(\mathbf{u}, \mathbf{u})\mathbf{Z}^\top = \sigma_u^2 \mathbf{Z}^\top,$$

where the result follows because the random terms in $\mathbf{y}$ are independent and $\mathrm{Cov}(\mathbf{u}, \mathbf{u}) = \sigma_u^2 \mathbf{I}$. Then it follows that

$$\hat{\mathbf{u}} = \mathrm{E}(\mathbf{u}) + \boldsymbol{\Sigma}_u \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \sigma_u^2 \mathbf{Z}^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}),$$

which matches Equation 3. Similarly, the BLUP of $\boldsymbol{\tau}$ is found by noting that $\mathrm{E}(\boldsymbol{\tau}) = \mathbf{0}$ and

$$\boldsymbol{\Sigma}_{de} = \mathrm{Cov}(\boldsymbol{\tau}, \mathbf{y}) = \mathrm{Cov}(\boldsymbol{\tau}, \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\tau} + \boldsymbol{\epsilon}) = \mathrm{Cov}(\boldsymbol{\tau}, \boldsymbol{\tau}) = \sigma_{de}^2 \mathbf{R},$$

where the result follows because the random terms in $\mathbf{y}$ are independent and $\mathrm{Cov}(\boldsymbol{\tau}, \boldsymbol{\tau}) = \sigma_{de}^2 \mathbf{R}$, and σ_{de}^2 is the variance of $\boldsymbol{\tau}$. Then it follows that

$$\hat{\boldsymbol{\tau}} = \mathrm{E}(\boldsymbol{\tau}) + \boldsymbol{\Sigma}_{de} \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \sigma_{de}^2 \mathbf{R} \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}). \tag{5}$$

Fitted values for $\boldsymbol{\epsilon}$ are obtained using similar arguments. Evaluating these equations at the plug-in (empirical) estimates of the covariance parameters yields EBLUPs.

When partition factors are used, the covariance matrix of all random effects (spatial and non-spatial) can be viewed as the interaction between the non-partitioned covariance matrix and the partition matrix, $\mathbf{P}$. The ijth entry in $\mathbf{P}$ equals one if observation i and observation j share the same level of the partition factor and zero otherwise. For spatial random effects, an adjustment is straightforward, as each column in $\boldsymbol{\Sigma_{de}}$ corresponds to a distinct spatial random effect. Thus with partition factors, $\boldsymbol{\Sigma^*_{de}} = \boldsymbol{\Sigma_{de}} \odot \mathbf{P} = \sigma^2_{de}\mathbf{R} \odot \mathbf{P}$, where $\odot$ denotes the Hadmart (element-wise) product, is used instead used of $\boldsymbol{\Sigma_{de}}$ in Equation 5. Note that $\boldsymbol{\Sigma_{ie}}$ is unchanged as it is proportional to the identity matrix. For non-spatial random effects, however, the situation is more complicated. Applying Equation 3 directly yields BLUPs of random effects corresponding to the interaction between random effect levels and partition levels. Thus a logical approach is to average the non-zero BLUPs for each random effect level across partition levels, yielding a prediction for the random effect level. This does not imply, however, that these estimates are BLUPs of the random effect.

For big data without partition factors, the local indexes act as partition factors. That is, the BLUPs correspond to random effects interacted with each local index. For big data with partition factors, an adjusted partition factor is created as the interaction between each local index and the partition factor. Then this adjusted partition factor is applied to Equation 4.

`hatvalues()`

Hat values measure the leverage of an observation. An observation has high leverage if its combination of explanatory variables is atypical (far from the mean explanatory vector). The spatial leverage (hat) matrix is given by

$$\mathbf{H}_s = \mathbf{X}^*(\mathbf{X}^{*\top}\mathbf{X}^*)^{-1}\mathbf{X}^{*\top}. \tag{6}$$

The diagonal of this matrix yields the leverage (hat) values for each observation [18]. The larger the hat value, the larger the leverage

To better understand the form in Equation 6, recall that the the non-spatial linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ assumes elements of $\boldsymbol{\epsilon}$ are independent and identically distributed (iid) with constant variance. In this context, the leverage (hat) matrix is given by

$$\mathbf{H} \equiv \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top,$$

When the elements of $\boldsymbol{\epsilon}$ are not iid or do not have constant variance or both, the spatial leverage (hat) matrix is not $\mathbf{H}$. First the linear model must be whitened according to $\mathbf{y}^* = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\epsilon}^*$, where $\boldsymbol{\epsilon}^*$ is the whitened version of the sum of all random errors in the model. Then the spatial leverage (hat) matrix follows using the whitened version of $\mathbf{X}$, $\mathbf{X}^*$.

`logLik()`

The log-likelihood is given by $\ell(\hat{\boldsymbol{\Theta}})$.

`loocv()`

k-fold cross validation is a useful tool for evaluating model fits using "hold-out" data. The data are split into k sets. One-by-one, one of the k sets is held out, the model is fit to the remaining $k-1$ sets, and predictions at each observation in the hold-out set are compared to their true values. The closer the predictions are to the true observations,

the better the model fit. A special case where $k = n$ is known as leave-one-out cross validation (loocv), as each observation is left out one-by-one. Computationally efficient solutions exist for leave-one-out cross validation in the non-spatial linear model (with iid, constant variance errors). Outside of this case, however, fitting n separate models can be computationally infeasible. `loocv()` makes a compromise that balances an approximation to the true solution with computational feasibility. First $\boldsymbol{\theta}$ is estimated using all of the data. Then for each of the n model fits, `loocv()` does not re-estimate $\boldsymbol{\theta}$ but does re-estimate $\boldsymbol{\beta}$. This approach relies on the assumption that the covariance parameter estimates obtained using $n-1$ observations are approximately the same as the covariance parameter estimates obtained using all n observations. For a large enough sample size, this is a reasonable assumption.

First define $\boldsymbol{\Sigma}_{-i,-i}$ as $\boldsymbol{\Sigma}$ with the ith row and column deleted, $\boldsymbol{\Sigma}_{i,-i}$ as the ith row of $\boldsymbol{\Sigma}$ with the ith column deleted, $\boldsymbol{\Sigma}_{i,i}$ as the ith row and column of $\boldsymbol{\Sigma}$, $\mathbf{X}_{-i}$ as $\mathbf{X}$ with the ith row deleted, $\mathbf{X}_i$ as the ith row of $\mathbf{X}$, y_{-i} as $\mathbf{y}$ with the ith element deleted, and $\mathbf{y}_i$ as the ith element of $\mathbf{y}$. [22] shows that given $\boldsymbol{\Sigma}^{-1}$, a computationally efficient form for $\boldsymbol{\Sigma}_{-i}^{-1}$ exists. First observe that $\boldsymbol{\Sigma}^{-1}$ can be represented blockwise as

$$\boldsymbol{\Sigma}^{-1} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_{-i,-i} & \tilde{\boldsymbol{\Sigma}}_{i,-i}^\top \\ \tilde{\boldsymbol{\Sigma}}_{i,-i} & \tilde{\boldsymbol{\Sigma}}_{i,i} \end{bmatrix},$$

where the dimensions of each $\tilde{\boldsymbol{\Sigma}}$ match the respective dimensions of relevant blocks in $\boldsymbol{\Sigma}$. Then it follows that

$$\boldsymbol{\Sigma}_{-i,-i}^{-1} = \tilde{\boldsymbol{\Sigma}}_{-i,-i} - \tilde{\boldsymbol{\Sigma}}_{i,-i}^\top \tilde{\boldsymbol{\Sigma}}_{i,i}^{-1} \tilde{\boldsymbol{\Sigma}}_{i,-i}$$

and

$$\boldsymbol{\beta}_{-i} = (\mathbf{X}_{-i}^\top \boldsymbol{\Sigma}_{-i,-i}^{-1} \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top \boldsymbol{\Sigma}_{-i,-i}^{-1} \mathbf{y}_{-i},$$

where $\boldsymbol{\beta}_i$ is the estimate of $\boldsymbol{\beta}$ constructed without the ith observation.

The loocv prediction of y_i is then given by

$$\hat{y}_i = \mathbf{X}_i \hat{\boldsymbol{\beta}}_{-i} + \hat{\boldsymbol{\Sigma}}_{i,-i} \hat{\boldsymbol{\Sigma}}_{-i,-i} (\mathbf{y}_i - \mathbf{X}_{-i} \hat{\boldsymbol{\beta}}_{-i})$$

and the prediction variance of the loocv prediction of y_i is given by

$$\dot{\sigma}_i^2 = \hat{\boldsymbol{\Sigma}}_{i,i} - \hat{\boldsymbol{\Sigma}}_{i,-i} \hat{\boldsymbol{\Sigma}}_{-i,-i}^{-1} \hat{\boldsymbol{\Sigma}}_{i,-i}^\top + \mathbf{Q}_i (\mathbf{X}_{-i}^\top \hat{\boldsymbol{\Sigma}}_{-i,-i}^{-1} \mathbf{X}_{-i})^{-1} \mathbf{Q}_i^\top,$$

$\mathbf{Q}_i = \mathbf{X}_i - \hat{\boldsymbol{\Sigma}}_{i,-i} \hat{\boldsymbol{\Sigma}}_{-i,-i}^{-1} \mathbf{X}_{-i}$. These formulas are analogous to the formulas used to obtain linear unbiased predictions of unobserved data (Equation 7) and prediction variances (Equation 8) in Section . Model fits are evaluated using mean squared prediction error (mspe), formally defined as

$$mspe = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

Generally the lower the *mspe*, the better the model fit.

Big Data

Options for big data leave-one-out cross validation rely on the `local` argument, which is passed to `predict()`. The `local` list for `predict()` is explained in detail in Section , but we provide a short summary of how `local` interacts with `loocv()` here.

For `splm()` and `spautor()` objects, `local` can be `"all"`. When `local = "all"`, all of the data are used for leave-one-out cross validation (i.e., it is implemented exactly as previously described). Parallelization is implemented when setting `parallel = TRUE` in `local`, and the number of cores to use for parallelization is specified via `ncores`.

For `splm()` objects, `local` can be `"covariance"` or `"distance"`. When `local = "covariance"`, then a number of observations (specified via the `size` argument) having the highest covariance with the held-out observation are used in the local neighborhood prediction approach. When `local = "distance"`, then a number of observations (specified via the `size` argument) closest to the held-out observation are used in the local neighborhood prediction approach. When no random effects are used, no partition factor is used, and the spatial covariance function is monotone decreasing, `"covariance"` and `"distance"` are equivalent. The local neighborhood approach only uses the observations in the local neighborhood of the held-out observation to perform prediction, and is thus an approximation to the true solution. Its computational efficiency derives from using $\mathbf{\Sigma}_{l,l}$ (the covariance matrix of the observations in the local neighborhood) instead of $\mathbf{\Sigma}$ (the covariance matrix of all the observations). Parallelization is implemented when setting `parallel = TRUE` in `local`, and the number of cores to use for parallelization is specified via `ncores`.

predict()

interval = "none"

The empirical best linear unbiased predictions (i.e., empirical Kriging predictor) of $\mathbf{y}_u$ are given by

$$\dot{\mathbf{y}}_u = \mathbf{X}_u\hat{\boldsymbol{\beta}} + \hat{\mathbf{\Sigma}}_{uo}\hat{\mathbf{\Sigma}}_o^{-1}(\mathbf{y}_o - \mathbf{X}_o\hat{\boldsymbol{\beta}}). \tag{7}$$

Equation 7 is sometimes called an empirical universal Kriging predictor, a Kriging with external drift predictor, or a regression Kriging predictor.

The covariance matrix of $\dot{\mathbf{y}}_u$

$$\dot{\mathbf{\Sigma}}_u = \hat{\mathbf{\Sigma}}_u - \hat{\mathbf{\Sigma}}_{uo}\hat{\mathbf{\Sigma}}_o^{-1}\hat{\mathbf{\Sigma}}_{uo}^\top + \mathbf{Q}(\mathbf{X}_o^\top\hat{\mathbf{\Sigma}}_o^{-1}\mathbf{X}_o)^{-1}\mathbf{Q}^\top, \tag{8}$$

where $\mathbf{Q} = \mathbf{X}_u - \hat{\mathbf{\Sigma}}_{uo}\hat{\mathbf{\Sigma}}_o^{-1}\mathbf{X}_o$.

When `se.fit = TRUE`, standard errors are returned by taking the square root of the diagonal of $\dot{\mathbf{\Sigma}}_u$ in Equation 8.

interval = "prediction"

The empirical best linear unbiased predictions are returned by evaluating Equation 7. The $(100 \times$ `level`$)\%$ prediction interval for $(y_u)_i$ is $(\dot{y}_u)_i \pm z^*\sqrt{(\dot{\mathbf{\Sigma}}_u)_{i,i}}$, where $\sqrt{(\dot{\mathbf{\Sigma}}_u)_{i,i}}$ is the standard error of $(\dot{y}_u)_i$ obtained from `se.fit = TRUE`, $\Phi(z^*) = 1 - \alpha/2$, $\Phi(\cdot)$ is the standard normal (Gaussian) cumulative distribution function, $\alpha = 1-$ `level`, and `level` is an argument to `predict()`. The default for `level` is 0.95, which corresponds to a z^* of approximately 1.96.

interval = "confidence"

The best linear unbiased estimates of $\mathrm{E}[(y_u)_i]$ ($\mathrm{E}(\cdot)$ denotes expectation) are returned by evaluating $(\mathbf{X}_u)_i\hat{\boldsymbol{\beta}}$ (i.e., fitted values corresponding to $(\mathbf{X}_u)_i$). The $(100 \times$ `level`$)\%$ confidence interval for $\mathrm{E}[(y_u)_i]$ is $(\mathbf{X}_u)_i\hat{\boldsymbol{\beta}} \pm z^*\sqrt{(\mathbf{X}_u)_i(\mathbf{X}_o^\top\hat{\mathbf{\Sigma}}_o^{-1}\mathbf{X}_o)^{-1}(\mathbf{X}_u)_i^\top}$, where $(\mathbf{X}_u)_i$ is the ith row of $\mathbf{X}_u$, $\sqrt{(\mathbf{X}_u)_i(\mathbf{X}_o^\top\hat{\mathbf{\Sigma}}_o^{-1}\mathbf{X}_o)^{-1}(\mathbf{X}_u)_i^\top}$ is the standard error of $(\dot{y}_u)_i$ obtained from `se.fit = TRUE`, $\Phi(z^*) = 1 - \alpha/2$, $\Phi(\cdot)$ is the standard normal (Gaussian) cumulative distribution function, $\alpha = 1-$ `level`, and `level` is an argument to `predict()`. The default for `level` is 0.95, which corresponds to a z^* of approximately 1.96.

`spautor()` extra steps

For spatial autoregressive models, an extra step is required to obtain $\hat{\boldsymbol{\Sigma}}_o^{-1}$, $\hat{\boldsymbol{\Sigma}}_u$, and $\hat{\boldsymbol{\Sigma}}_{uo}$ as they depend on one another through the neighborhood structure of $\mathbf{y}_o$ and $\mathbf{y}_u$. Recall that for autoregressive models, it is $\boldsymbol{\Sigma}^{-1}$ that is straightforward to obtain, not $\boldsymbol{\Sigma}$.

Let $\boldsymbol{\Sigma}^{-1}$ be the inverse covariance matrix of the observed and unobserved data, $\mathbf{y}_o$ and $\mathbf{y}_u$. One approach to obtain $\boldsymbol{\Sigma}_o$ and $\boldsymbol{\Sigma}_{uo}$ is to directly invert $\boldsymbol{\Sigma}^{-1}$ and then subset $\boldsymbol{\Sigma}$ appropriately. This inversion can be prohibitive when $n_o + n_u$ is large. A faster way to obtain $\boldsymbol{\Sigma}_o$ and $\boldsymbol{\Sigma}_{uo}$ exists. Represent $\boldsymbol{\Sigma}^{-1}$ blockwise as

$$\boldsymbol{\Sigma}^{-1} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_o & \tilde{\boldsymbol{\Sigma}}_{uo}^\top \\ \tilde{\boldsymbol{\Sigma}}_{uo} & \tilde{\boldsymbol{\Sigma}}_u \end{bmatrix},$$

where the dimensions of the blocks match the relevant dimensions of $\boldsymbol{\Sigma}$. All of the terms required for prediction can be obtained from this block representation. [22] shows that

$$\begin{aligned} \boldsymbol{\Sigma}_o^{-1} &= \tilde{\boldsymbol{\Sigma}}_o - \tilde{\boldsymbol{\Sigma}}_{uo}^\top (\tilde{\boldsymbol{\Sigma}}_u)^{-1} \tilde{\boldsymbol{\Sigma}}_{uo} \\ \boldsymbol{\Sigma}_u &= (\tilde{\boldsymbol{\Sigma}}_u - \tilde{\boldsymbol{\Sigma}}_{uo} (\tilde{\boldsymbol{\Sigma}}_o)^{-1} \tilde{\boldsymbol{\Sigma}}_{uo}^\top)^{-1} \\ \boldsymbol{\Sigma}_{uo} &= -\boldsymbol{\Sigma}_u \tilde{\boldsymbol{\Sigma}}_{uo} \tilde{\boldsymbol{\Sigma}}_o^{-1} \end{aligned}$$

Evaluating these expressions at $\hat{\boldsymbol{\theta}}$ yields $\hat{\boldsymbol{\Sigma}}_o^{-1}$, and $\hat{\boldsymbol{\Sigma}}_u$, and $\hat{\boldsymbol{\Sigma}}_{uo}$.

A similar result exists for the log determinant of $\boldsymbol{\Sigma}_o$, which is not required for prediction but is required for restricted maximum likelihood and maximum likelihood estimation.

Big Data

When the number of observations in the fitted model (observed data) are large or there are many locations to predict at or both, it is often necessary to implement computationally efficient big data approximations. Big data approximations are implemented in `spmodel` using the `local` argument to `predict()`. When the method in `local` is `"all"`, all of the fitted model data are used to make predictions. In this context, computational efficiency is only gained by parallelizing each prediction. The only available method for `spautor()` fitted models is `"all"`. This is because the neighborhood structure of `spautor()` fitted models does not permit the subsetting used by the `"covariance"` and `"distance"` methods that we discuss next.

When the `local` method is `"covariance"`, $\hat{\boldsymbol{\Sigma}}_{uo}$ is computed between the observation being predicted ($\mathbf{y}_u$) and the rest of the observed data. This vector is then ordered and a number of observations (specified via the `size` argument) having the highest covariance with $\mathbf{y}_u$ are subset, yielding $\check{\boldsymbol{\Sigma}}_{uo}$, which has dimension $1 \times size$. Then similarly $\hat{\boldsymbol{\Sigma}}_o$, $\mathbf{y}_o$, and $\mathbf{X}_u$ are also subset by these `size` observations, yielding $\check{\boldsymbol{\Sigma}}_o$, $\check{\mathbf{y}}_o$, and $\check{\mathbf{X}}_u$, respectively. Equations 7 and 8 can be evaluated at $\check{\boldsymbol{\Sigma}}_{uo}$, $\check{\boldsymbol{\Sigma}}_o$, $\check{\mathbf{y}}_o$, and $\check{\mathbf{X}}_u$. When the local method is `"distance"`, a similar approach is used except a number of observations (specified via the `size` argument) closest (in terms of Euclidean distance) to $\mathbf{y}_u$ are subset instead. When random effects are not used, partition factors are not used, and the spatial covariance function is monotone decreasing, `"covariance"` and `"distance"` are equivalent. This approach of subsetting the observed data by the set of locations closest in covariance or proximity to $\mathbf{y}_u$ is known as the local neighborhood approach. As long as `size` is relatively small (the default is 50), the local neighborhood approach is very computationally efficient, mainly because $\check{\boldsymbol{\Sigma}}_o^{-1}$ is easy to compute. Additional computational efficiency is gained by parallelizing each prediction.

pseudoR2()

The pseudo R-squared is a generalization of the classical R-squared from non-spatial linear models. Like the classical R-squared, the pseudo R-squared measures the proportion of variability in the response explained by the fixed effects in the fitted model. Unlike the classical R-squared, the pseudo R-squared can be applied to models whose errors do not satisfy the iid and constant variance assumption. The pseudo R-squared is given by

$$PR2 = 1 - \frac{\mathcal{D}(\hat{\boldsymbol{\Theta}})}{\mathcal{D}(\hat{\boldsymbol{\Theta}}_0)}.$$

For normal (Gaussian) random errors, the pseudo R-squared is

$$PR2 = 1 - \frac{(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{(\mathbf{y} - \hat{\mu})^\top \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \hat{\mu})},$$

where $\hat{\mu} = (\mathbf{1}^\top \hat{\boldsymbol{\Sigma}}^{-1}\mathbf{1})^{-1}\mathbf{1}^\top \hat{\boldsymbol{\Sigma}}^{-1}\mathbf{y}$. For the non-spatial model, the pseudo R-squared reduces to the classical R-squared, as

$$PR2 = 1 - \frac{(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{(\mathbf{y} - \hat{\mu})^\top \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \hat{\mu})} = 1 - \frac{(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{(\mathbf{y} - \hat{\mu})^\top (\mathbf{y} - \hat{\mu})} = 1 - \frac{\text{SSE}}{\text{SST}} = R2,$$

where SSE denotes the error sum of squares and SST denotes the total sum of squares. The result follows because for a non-spatial model, $\boldsymbol{\Sigma}$ is proportional to the identity matrix.

The adjusted pseudo r-squared adjusts for additional explanatory variables and is given by

$$PR2adj = 1 - (1 - PR2)\frac{n-1}{n-p}.$$

If the fitted model does not have an intercept, the $n-1$ term is instead n.

residuals()

Terminology regarding residual names if often conflicting and confusing. Because of this, next we explicitly define the residual options in `spmodel`. These definitions may be different from others you may have seen in the literature.

When `type = "raw"`, raw residuals are returned:

$$\mathbf{e}_r = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}.$$

When `type = "pearson"`, pearson residuals are returned:

$$\mathbf{e}_p = \hat{\boldsymbol{\Sigma}}^{-1/2}\mathbf{e}_r,$$

If the errors are normal (Gaussian), the pearson residuals should be approximately normally distributed with mean zero and variance one. The result follows when $\hat{\boldsymbol{\Sigma}}^{-1/2} \approx \boldsymbol{\Sigma}^{-1/2}$ because

$$\text{E}(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}_r) = \boldsymbol{\Sigma}^{-1/2}\text{E}(\mathbf{e}_r) = \boldsymbol{\Sigma}^{-1/2}\mathbf{0} = \mathbf{0}$$

and

$$\begin{aligned}
\text{Cov}(\boldsymbol{\Sigma}^{-1/2}\mathbf{e}_r) &= \boldsymbol{\Sigma}^{-1/2}\text{Cov}(\mathbf{e}_r)\boldsymbol{\Sigma}^{-1/2} \\
&\approx \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1/2} \\
&= (\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}^{1/2})(\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{-1/2}) \\
&= \mathbf{I}
\end{aligned}$$

When `type = "standardized"`, standardized residuals are returned:

$$\mathbf{e}_s = \frac{\mathbf{e}_p}{\sqrt{1 - diag(\mathbf{H}^*)}},$$

where $diag(\mathbf{H}^*)$ is the diagonal of the spatial hat matrix, $\mathbf{H}_s \equiv \mathbf{X}^*(\mathbf{X}^{*\top}\mathbf{X}^*)^{-1}\mathbf{X}^{*\top}$. This residual transformation "standardizes" the Pearson residuals. As such, the standardized residuals should also have mean zero and variance

$$\begin{aligned}
\text{Cov}(\mathbf{e}_s) &= \text{Cov}((\mathbf{I} - \mathbf{H}^*)\hat{\boldsymbol{\Sigma}}^{-1/2}\mathbf{y}) \\
&\approx \text{Cov}((\mathbf{I} - \mathbf{H}^*)\boldsymbol{\Sigma}^{-1/2}\mathbf{y}) \\
&= (\mathbf{I} - \mathbf{H}^*)\boldsymbol{\Sigma}^{-1/2}\text{Cov}(\mathbf{y})\boldsymbol{\Sigma}^{-1/2}(\mathbf{I} - \mathbf{H}^*)^\top \\
&= (\mathbf{I} - \mathbf{H}^*)\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1/2}(\mathbf{I} - \mathbf{H}^*)^\top \\
&= (\mathbf{I} - \mathbf{H}^*)\mathbf{I}(\mathbf{I} - \mathbf{H}^*)^\top \\
&= (\mathbf{I} - \mathbf{H}^*),
\end{aligned}$$

because $(\mathbf{I} - \mathbf{H}^*)$ is symmetric and idempotent. Note that the average value of $diag(\mathbf{H}^*)$ is p/n, so $(\mathbf{I} - \mathbf{H}^*) \approx \mathbf{I}$ for large sample sizes.

`spautor()` and `splm()`

Next we discuss technical details for the `spautor()` and `splm()` functions. Many of the details for the two functions are the same, though occasional differences are noted in the following subsection headers. Specifically, `spautor()` and `splm()` are for different data types and use different covariance functions. `spautor()` is for spatial linear models with areal data (i.e., spatial autoregressive models) and `splm()` is for spatial linear models with point-referenced data (i.e., geostatistical models). There are also a few features `splm()` has that `spautor()` does not: semivariogram-based estimation, random effects, anisotropy, and big data approximations.

`spautor()` Spatial Covariance Functions

For areal data, the covariance matrix depends on the specification of a neighborhood structure among the observations. Observations with at least one neighbor (not including itself) are called "connected" observations. Observations with no neighbors are called "unconnected" observations. The autoregressive spatial covariance matrix can be defined as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2_{de}\mathbf{R} & \mathbf{0} \\ \mathbf{0} & \sigma^2_{\xi}\mathbf{I} \end{bmatrix} + \sigma^2_{ie}\mathbf{I},$$

where σ^2_{de} (≥ 0) is the spatially dependent (correlated) variance for the connected observations, $\mathbf{R}$ is a matrix that describes the spatial dependence for the connected observations, σ^2_{ξ} (≥ 0) is the independent (not correlated) variance for the unconnected observations, and σ^2_{ie} (≥ 0) is the independent (not correlated) variance for all observations. As seen, the connected and unconnected observations are allowed different variances. The total variance for connected observations is then $\sigma^2_{de} + \sigma^2_{ie}$ and the total variance for unconnected observations is $\sigma^2_{\xi} + \sigma^2_{ie}$. `spmodel` accommodates two spatial covariances: conditional autoregressive (CAR) and simultaneous autoregressive (SAR), both of which have their $\mathbf{R}$ forms provided in Table 1. For both CAR and SAR covariance functions, $\mathbf{R}$ depends on similar quantities: $\mathbf{I}$, an identity matrix; ϕ, a range parameter, and $\mathbf{W}$, a matrix that defines the neighborhood structure. Often $\mathbf{W}$ is symmetric but it need not be. Valid values for ϕ are in $(1/\lambda_{max}, 1/\lambda_{min})$, where λ_{min} is

Spatial covariance type	$\mathbf{R}$ functional form
`"car"`	$(\mathbf{I} - \phi\mathbf{W})^{-1}\mathbf{M}$
`"sar"`	$[(\mathbf{I} - \phi\mathbf{W})(\mathbf{I} - \phi\mathbf{W})^\top]^{-1}$

Table 1. The forms of $\mathbf{R}$ for each spatial covariance type available in `spautor()`.

the minimum eigenvalue of $\mathbf{W}$ and λ_{max} is the maximum eigenvalue of $\mathbf{W}$. For SAR covariance functions, λ_{min} must be negative and λ_{max} must be positive. For CAR covariances functions, a matrix $\mathbf{M}$ matrix must be provided that satisfies the CAR symmetry condition, which enforces the symmetry of the covariance matrix. The CAR symmetry condition states

$$\frac{\mathbf{W}_{ij}}{\mathbf{M}_{ii}} = \frac{\mathbf{W}_{ji}}{\mathbf{M}_{jj}}$$

for all i and j, where i and j index rows or columns. When $\mathbf{W}$ is symmetric, $\mathbf{M}$ is often taken to be the identity matrix.

The default in `spmodel` is to row-standardize $\mathbf{W}$ by dividing each element by its respective row sum, which decreases variance. If row-standardization is not used for a CAR model, the default in `spmodel` for $\mathbf{M}$ is the identity matrix.

`splm()` Spatial Covariance Functions

For point-referenced data, the spatial covariance is given by

$$\sigma^2_{de}\mathbf{R} + \sigma^2_{ie}\mathbf{I},$$

where σ^2_{de} (≥ 0) is the spatially dependent (correlated) variance, $\mathbf{R}$ is a spatial correlation matrix, σ^2_{ie} (≥ 0) is the spatially independent (not correlated) variance, and $\mathbf{I}$ is an identity matrix. The $\mathbf{R}$ matrix always depends on a range parameter, ϕ (> 0), that controls the behavior of the covariance function with distance. For some covariance functions, the $\mathbf{R}$ matrix depends on an additional parameter that we call the "extra" parameter. Table 2 shows the parametric form for all $\mathbf{R}$ matrices available in `splm()`. In Table 2, the range parameter is denoted as ϕ, the distance divided by the range parameter (h/ϕ) is denoted as η, $\mathbb{1}\{\cdot\}$ is an indicator function equal to one when the argument occurs and zero otherwise, and the extra parameter is denoted as ξ (when relevant).

Model-fitting

Likelihood-based Estimation (`estmethod = "reml"` or `estmethod = "ml"`)

Minus twice a profiled (by $\boldsymbol{\beta}$) Gaussian log-likelihood is given by

$$-2\ell_p(\boldsymbol{\theta}) = \ln|\boldsymbol{\Sigma}| + (\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}) + n\ln 2\pi, \tag{9}$$

where $\tilde{\boldsymbol{\beta}} = (\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y}$. Minimizing Equation 9 yields $\hat{\boldsymbol{\theta}}_{ml}$, the maximum likelihood estimates for $\boldsymbol{\theta}$. Then a closed form solution exists for $\hat{\boldsymbol{\beta}}_{ml}$, the maximum likelihood estimates for $\boldsymbol{\beta}$: $\hat{\boldsymbol{\beta}}_{ml} = \tilde{\boldsymbol{\beta}}_{ml}$, where $\tilde{\boldsymbol{\beta}}_{ml}$ is $\tilde{\boldsymbol{\beta}}$ evaluated at $\hat{\boldsymbol{\theta}}_{ml}$. Unfortunately $\hat{\boldsymbol{\theta}}_{ml}$ can be badly biased for $\boldsymbol{\theta}$ (especially for small sample sizes), which impacts the estimation of $\boldsymbol{\beta}$ [23]. This bias occurs due to the simultaneous estimation of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. To reduce this bias, restricted maximum likelihood estimation (REML) emerged [23–25]. Integrating $\boldsymbol{\beta}$ out of a Gaussian likelihood yields the restricted Gaussian likelihood. Minus twice a restricted Gaussian log-likelihood is given by

$$-2\ell_R(\boldsymbol{\theta}) = -2\ell_p(\boldsymbol{\theta}) + \ln|\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}| - p\ln 2\pi, \tag{10}$$

Spatial covariance type	**R** functional form
`"exponential"`	$e^{-\eta}$
`"spherical"`	$(1 - 1.5\eta + 0.5\eta^3)\mathbb{1}\{h \leq \phi\}$
`"gaussian"`	$e^{-\eta^2}$
`"triangular"`	$(1 - \eta)\mathbb{1}\{h \leq \phi\}$
`"circular"`	$(1 - \frac{2}{\pi}[m\sqrt{1-m^2} + sin^{-1}\{m\}])\mathbb{1}\{h \leq \phi\}, m = min(\eta, 1)$
`"cubic"`	$(1 - 7\eta^2 + 8.75\eta^3 - 3.5\eta^5 + 0.75\eta^7)\mathbb{1}\{h \leq \phi\}$
`"pentaspherical"`	$(1 - 1.875\eta + 1.250\eta^3 - 0.375\eta^5)\mathbb{1}\{h \leq \phi\}$
`"cosine"`	$cos(\eta)$
`"wave"`	$\frac{sin(\eta)}{\eta}\mathbb{1}\{h > 0\} + \mathbb{1}\{h = 0\}$
`"jbessel"`	$B_j(h\phi), B_j$ is Bessel-J
`"gravity"`	$(1 + \eta^2)^{-1/2}$
`"rquad"`	$(1 + \eta^2)^{-1}$
`"magnetic"`	$(1 + \eta^2)^{-3/2}$
`"matern"`	$\frac{2^{(1-\xi)}}{\Gamma(\xi)}\alpha^\xi B_k(\alpha, \xi), \alpha = \sqrt{2\xi\eta}, B_k$ is Bessel-K with order ξ, $\xi \in [1/5, 5]$
`"cauchy"`	$(1 + \eta^2)^{-\xi}$, $\xi > 0$
`"pexponential"`	$exp(-h^\xi/\phi)$, $\xi \in (0, 2]$
`"none"`	0

Table 2. The forms of **R** for each spatial covariance type available in `splm()`. All spatial covariance functions are valid in two dimensions except `"triangular"` and `"cosine"`, which are only valid in one dimension.

where p equals the dimension of $\boldsymbol{\beta}$. Minimizing Equation 10 yields $\hat{\boldsymbol{\theta}}_{reml}$, the restricted maximum likelihood estimates for $\boldsymbol{\theta}$. Then a closed for solution exists for $\hat{\boldsymbol{\beta}}_{reml}$, the restricted maximum likelihood estimates for $\boldsymbol{\beta}$: $\hat{\boldsymbol{\beta}}_{reml} = \tilde{\boldsymbol{\beta}}_{reml}$, where $\tilde{\boldsymbol{\beta}}_{reml}$ is $\tilde{\boldsymbol{\beta}}$ evaluated at $\hat{\boldsymbol{\theta}}_{reml}$.

The covariance matrix can often be written as $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{\Sigma}^*$, where σ^2 is the overall variance and $\boldsymbol{\Sigma}^*$ is a covariance matrix that depends on parameter vector $\boldsymbol{\theta}^*$ with one less dimension than $\boldsymbol{\theta}$. Then the overall variance, σ^2, can be profiled out of Equation 9 and Equation 10. This reduces the number of parameters requiring optimization by one, which can dramatically reduce estimation time. Profiling σ^2 out of Equation 9 yields

$$-2\ell_p^*(\boldsymbol{\theta}^*) = \ln|\boldsymbol{\Sigma}^*| + n\ln[(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})^\top\boldsymbol{\Sigma}^{*-1}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})] + n + n\ln 2\pi/n.$$

After finding $\hat{\boldsymbol{\theta}}^*_{ml}$, a closed form solution for $\hat{\sigma}^2_{ml}$ exists:
$\hat{\sigma}^2_{ml} = [(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})^\top\boldsymbol{\Sigma}^{*-1}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})]/n$. Then $\hat{\boldsymbol{\theta}}^*_{ml}$ is combined with $\hat{\sigma}^2_{ml}$ to yield $\hat{\boldsymbol{\theta}}_{ml}$ and subsequently $\hat{\boldsymbol{\beta}}_{ml}$. A similar result holds for restricted maximum likelihood estimation. Profiling σ^2 out of Equation 10 yields

$$-2\ell_R^*(\boldsymbol{\Theta}) = \ln|\boldsymbol{\Sigma}^*| + (n-p)\ln[(\mathbf{y}-\mathbf{X}\tilde{\boldsymbol{\beta}})^\top\boldsymbol{\Sigma}^{*-1}(\mathbf{y}-\mathbf{X}\tilde{\boldsymbol{\beta}})] + \ln|\mathbf{X}^\top\boldsymbol{\Sigma}^{*-1}\mathbf{X}| + (n-p) + (n-p)\ln 2\pi/(n-$$

After finding $\hat{\boldsymbol{\theta}}^*_{reml}$, a closed form solution for $\hat{\sigma}^2_{reml}$ exists:
$\hat{\sigma}^2_{reml} = [(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})^\top\boldsymbol{\Sigma}^{*-1}(\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}})]/(n-p)$. Then $\hat{\boldsymbol{\theta}}^*_{reml}$ is combined with $\hat{\sigma}^2_{reml}$ to yield $\hat{\boldsymbol{\theta}}_{reml}$ and subsequently $\hat{\boldsymbol{\beta}}_{reml}$. For more on profiling Gaussian likelihoods, see [25].

Both maximum likelihood and restricted maximum likelihood estimation rely on the $n \times n$ covariance matrix inverse. Inverting an $n \times n$ matrix is an enormous computational demand that scales cubically with the sample size. For this reason, maximum likelihood and restricted maximum likelihood estimation have historically been infeasible to implement in their standard form with data larger than a few thousand observations. This motivates the use for the big data approaches outlined in Section .

Semivariogram-based Estimation (`splm()` only)

An alternative approach to likelihood-based estimation is semivariogram-based estimation. The semivariogram of a constant-mean process $\mathbf{y}$ is the expectation of half of the squared difference between two observations h distance units apart. More formally, the semivariogram is denoted $\gamma(h)$ and defined as

$$\gamma(h) = \mathrm{E}[(y_i - y_j)^2]/2,$$

where h is the Euclidean distance between the locations of y_i and y_j. When the process $\mathbf{y}$ is second-order stationary, the semivariogram and covariance function are intimately connected: $\gamma(h) = \sigma^2 - \mathrm{Cov}(h)$, where σ^2 is the overall variance and $\mathrm{Cov}(h)$ is the covariance function evaluated at h. As such, the semivariogram and covariance function rely on the same parameter vector $\boldsymbol{\theta}$. Both of the semivariogram approaches described next are more computationally efficient than restricted maximum likelihood and maximum likelihood estimation because the major computational burden of the semivariogram approaches (calculations based on squared differences among pairs) scales quadratically with the sample size (i.e., not the cubed sample size like the likelihood-based approaches).

Weighted Least Squares (`estmethod = "sv-wls"`) The empirical semivariogram is a moment-based estimate of the semivariogram denoted by $\hat{\gamma}(h)$. Recall it is defined in Equation 2 as

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{N(h)} (y_i - y_j)^2,$$

where $N(h)$ is the set of observations in $\mathbf{y}$ that are h distance units apart (distance classes) and $|N(h)|$ is the cardinality of $N(h)$ [19]. More computational details are provided in Section . One criticism of the empirical semivariogram is that distance bins and cutoffs tend to be arbitrarily chosen (i.e., not chosen according to some statistical criteria).

[26] proposed estimating $\boldsymbol{\theta}$ by minimizing an objective function that involves $\gamma(h)$ and $\hat{\gamma}(h)$ and is based on a weighted least squares criterion. This criterion is defined as

$$\sum_i w_i[\hat{\gamma}(h)_i - \gamma(h)_i]^2, \tag{11}$$

where w_i, $\hat{\gamma}(h)_i$, and $\gamma(h)_i$ are the weights, empirical semivariogram, and semivariogram for the ith distance class, respectively. Minimizing Equation 11 yields $\hat{\boldsymbol{\theta}}_{wls}$, the semivariogram weighted least squares estimate of $\boldsymbol{\theta}$. After estimating $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ estimates are constructed using (empirical) generalized least squares: $\hat{\boldsymbol{\beta}}_{wls} = (\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{y}$.

[26] recommends setting the w_i in Equation 11 as $w_i = |N(h)|/\gamma(h)_i^2$, which gives more weight to distance classes with more observations ($|N(h)|$) and shorter distances ($1/\gamma(h)_i^2$). The default in `spmodel` is to use these w_i, known as Cressie weights, though several other options for w_i exist and are available via the `weights` argument. Table 3 contains all w_i available via the `weights` argument.

The number of $N(h)$ classes and the maximum distance for h are specified by passing the `bins` and `cutoff` arguments to `splm()` (these arguments are passed via `...` to `esv()`). The default value for `bins` is 15 and the default value for `cutoff` is half the maximum distance of the spatial domain's bounding box.

Recall that the semivariogram is defined for a constant-mean process. Generally, $\mathbf{y}$ does not necessarily have a constant mean so the empirical semivariogram and $\hat{\boldsymbol{\theta}}_{wls}$ are typically constructed using the residuals from an ordinary least squares regression of $\mathbf{y}$ on $\mathbf{X}$. These ordinary least squares residuals are assumed to have mean zero.

w_i Name	w_i Form	`weight =`
Cressie	$\lvert N(h)\rvert/\gamma(h)_i^2$	`"cressie"`
Cressie (Denominator) Root	$\lvert N(h)\rvert/\gamma(h)_i$	`"cressie-dr"`
Cressie No Pairs	$1/\gamma(h)_i^2$	`"cressie-nopairs"`
Cressie (Denominator) Root No Pairs	$1/\gamma(h)_i$	`"cressie-dr-nopairs"`
Pairs	$\lvert N(h)\rvert$	`"pairs"`
Pairs Inverse Distance	$\lvert N(h)\rvert/h^2$	`"pairs-invd"`
Pairs Inverse (Root) Distance	$\lvert N(h)\rvert/h$	`"pairs-invrd"`
Ordinary Least Squares	1	`"ols"`

Table 3. Table of values for the `weights` argument in `splm()` when `estmethod = "sv-wls"`.

Composite Likelihood (`estmethod = "sv-cl"`) Composite likelihood approaches involve constructing likelihoods based on conditional or marginal events for which likelihoods are available and then adding together these individual components. Composite likelihoods are attractive because they behave very similar to likelihoods but are easier to handle, both from a theoretical and from a computational perspective. [27] derive a particular composite likelihood for estimating semivariogram parameters. The negative log of this composite likelihood, denoted $\mathrm{CL}(h)$, is given by

$$\mathrm{CL}(h) = \sum_{i=1}^{n-1} \sum_{j>i} \left(\frac{(y_i - y_j)^2}{2\gamma(h)} + \ln(\gamma(h)) \right) \tag{12}$$

where $\gamma(h)$ is the semivariogram. Minimizing Equation 12 yields $\hat{\boldsymbol{\theta}}_{cl}$, the semivariogram composite likelihood estimates of $\boldsymbol{\theta}$. After estimating $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ estimates are constructed using (empirical) generalized least squares: $\hat{\boldsymbol{\beta}}_{cl} = (\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{y}$.

An advantage of the composite likelihood approach to semivariogram estimation is that it does not require arbitrarily specifying empirical semivariogram bins and cutoffs. It does tend to be more computationally demanding than weighted least squares, however. The composite likelihood is constructed from $\binom{n}{2}$ pairs for a sample size n, whereas the weighted least squares approach only requires calculating $\binom{|N(h)|}{2}$ pairs for each distance bin $N(h)$. As with the weighted least squares approach, Equation 12 requires a constant-mean process, so typically the residuals from an ordinary least squares regression of $\mathbf{y}$ on $\mathbf{X}$ are used to estimate $\boldsymbol{\theta}$.

Optimization

Parameter estimation is performed using `stats::optim()`. The default estimation method is Nelder-Mead [28] and the stopping criterion is a relative convergence tolerance (`reltol`) of .0001. If only one parameter requires estimation (on the profiled scale if relevant), the Brent algorithm is instead used [29]. Arguments to `optim()` are passed via `...` to `splm()` and `spautor()`. For example, the default estimation method and convergence criteria are overridden by passing `method` and `control`, respectively, to `splm()` and `spautor()`. If the `lower` and `upper` arguments to `optim()` are specified in `splm()` and `spautor()` to be passed to `optim()`, they are ignored, as optimization for all parameters is generally unconstrained. Initial values for `optim()` are found using the grid search described next.

Grid Search

`spmodel` uses a grid search to find suitable initial values for use in optimization. For spatial linear models without random effects, the spatially dependent variance (σ^2_{de}) and

σ^2_{de}	σ^2_{ie}	ϕ	α	S
9	1	15	0	1
1	9	15	0	1
5	5	15	0	1
9	1	45	0	1
1	9	45	0	1
5	5	45	0	1

Table 4. Grid search parameter configurations for an isotropic exponential spatial covariance with inflated sample variance 10.

spatially independent variance (σ^2_{ie}) parameters are given "low", "medium", and "high" values. The sample variance of a non-spatial linear model is slightly inflated by a factor of 1.2 (non-spatial models can underestimate the variance when there is spatial dependence) and these "low", "medium", and "high" values correspond to 10%, 50%, and 90% of the inflated sample variance. Only combinations of σ^2_{de} and σ^2_{ie} whose proportions sum to 100% are considered. The range (ϕ) and extra (ξ) parameters are given "low" and "high" values that are unique to each spatial covariance function. The anisotropy (Section) rotation parameter (α) is given six values that correspond to 0, $\pi/6$, $2\pi/6$, $4\pi/6$, $5\pi/6$, and π radians. The anisotropy scale parameter (S) is given "low", "medium", and "high" values that correspond to scaling factors of 0.25, 0.75, and 1. Note that the anisotropy parameters are only used during grid searches for point-referenced data.

The crossing of all appropriate parameter values is considered. If initial values are used for a parameter, the initial value replaces all values of the parameter in this crossing. Duplicate crossings are then omitted. The parameter configuration that yields the smallest value of the objective function is then used as an initial value for optimization. Suppose the inflated sample variance is 10 and the exponential covariance is used assuming isotropy. The parameter configurations evaluated are shown in Table 4.

For spatial linear models with random effects, the same approach is used to create a crossing of spatial covariance parameters. A separate approach is used to create a set of random effect variances. The random effect variances are similarly first grouped by proportions. The first combination is such that the first random effect variance is given 90% of variance, and the remaining 10% is spread out evenly among the remaining random effect variances. The second combination is such that the second random effect variance is given 90% of the variance, and the remaining 10% is spread out evenly among the remaining random effect variances. And so on and so forth. These combinations ascertain whether one random effect dominates variability. A final grouping is lastly considered: all 100% of variance is spread out evenly among all random effects.

When finding parameter values σ^2_{de}, σ^2_{ie}, and the random effect variances ($\sigma^2_{u_i}$ for the ith random effect), three scenarios are considered. In the first scenario, σ^2_{de} and σ^2_{ie} get 90% of the inflated sample variance and the random effect variances get 10%. In this scenario, only the random effect grouping where the variance is evenly spread out is considered. This is because the random effect variances are already contributing little to the overall variability, so performing additional objective function evaluations is unnecessary. In the second scenario, the random effects get 90% of the inflated sample variances and σ^2_{de} and σ^2_{ie} get 10%. Similarly in this scenario, only the σ^2_{de} and σ^2_{ie} grouping where the variance is evenly spread out is considered. Also in this scenario, only the lowest value for `range` and `extra` are used. In the third scenario, the 50% of the inflated sample variance is given to σ^2_{de} and σ^2_{ie} and 50% to the random effects. In this scenario, the only parameter combination considered is the case where variances are evenly spread out among σ^2_{de}, σ^2_{ie}, and the random effect variances. Together, there are parameter configurations where the spatial variability dominates (scenario 1), the

σ^2_{de}	σ^2_{ie}	ϕ	α	S	$\sigma^2_{u_1}$	$\sigma^2_{u_2}$
8.1	0.9	15	0	1	0.5	0.5
0.9	8.1	15	0	1	0.5	0.5
4.5	4.5	15	0	1	0.5	0.5
8.1	0.9	45	0	1	0.5	0.5
0.9	8.1	45	0	1	0.5	0.5
4.5	4.5	45	0	1	0.5	0.5
0.5	0.5	15	0	1	8.1	0.9
0.5	0.5	15	0	1	0.9	8.1
0.5	0.5	15	0	1	4.5	4.5
2.5	2.5	15	0	1	2.5	2.5
2.5	2.5	45	0	1	2.5	2.5

Table 5. Grid search parameter configurations for an isotropic exponential spatial covariance with two random effects and inflated sample variance 10.

random variability dominates (scenario 2), and where there is an even contribution from spatial and random variability. The parameter configuration that minimizes the objective function is then used as an initial value for optimization. Recall that random effects are only used with restricted maximum likelihood or maximum likelihood estimation, so the objective function is always a likelihood.

Suppose the inflated sample variance is 10, the exponential covariance is used assuming isotropy, and there are two random effects. The parameter configurations evaluated are shown in Table 5.

This grid search approach balances a thorough exploration of the parameter space with computational efficiency, as each objective function evaluation can be computationally expensive.

Hypothesis Testing

The hypothesis tests for $\hat{\boldsymbol{\beta}}$ returned by `summary()` or `tidy()` of an `splm` or `spautor` object are asymptotic z-tests based on the normal (Gaussian) distribution (Wald tests). The null hypothesis for the test associated with each $\hat{\beta}_i$ is that $\beta_i = 0$. Then the test statistic is given by

$$\tilde{z} = \frac{\hat{\beta}_i}{\text{SE}(\hat{\beta}_i)},$$

where $\text{SE}(\hat{\beta}_i)$ is the standard error of $\hat{\beta}_i$, which equals the square root of the ith diagonal element of $(\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1}$. The p-value is given by $2 * (1 - \Phi(|\tilde{z}|))$, which corresponds to an equal-tailed, two-sided hypothesis test of level α where $\Phi(\cdot)$ denotes the standard normal (Gaussian) cumulative distribution function and $|\cdot|$ denotes the absolute value.

Random Effects (`splm()` only and `"reml"` or `"ml"` `estmethod` only)

The random effects contribute directly to the covariance through their design matrices. Let $\mathbf{u}$ be a mean-zero random effect column vector of length n_u, where n_u is the number of levels of the random effect, with design matrix $\mathbf{Z}_u$. Then $\text{Cov}(\mathbf{Z}_u\mathbf{u}) = \mathbf{Z}_u \text{Cov}(\mathbf{u}) \mathbf{Z}_u^\top$. Because each element of $\mathbf{u}$ is independent of one another, this reduces to $\text{Cov}(\mathbf{Z}_u\mathbf{u}) = \sigma_u^2 \mathbf{Z}_u \mathbf{Z}_u^\top$, where σ_u^2 is the variance parameter corresponding to the random effect (i.e., the random effect variance parameter).

The $\mathbf{Z}$ matrices index the levels of the random effect. $\mathbf{Z}$ has dimension $n \times n_u$, where n is the sample size. Each row of $\mathbf{Z}$ corresponds to an observation and each column to a level of the random effect. For example, suppose we have $n = 4$ observations, so

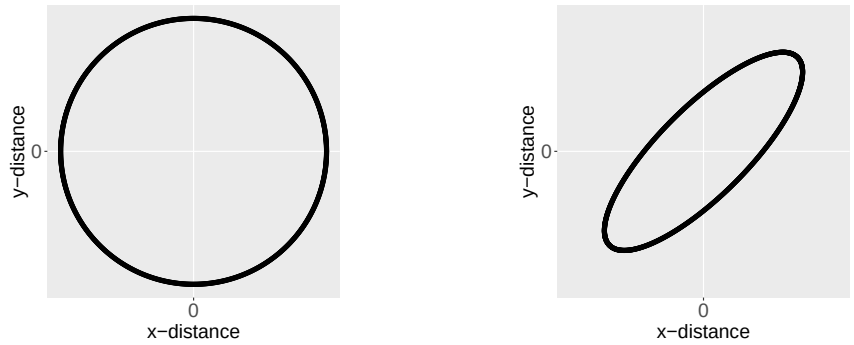

Fig 1. In the left figure, the ellipse of an isotropic spatial covariance function centered at the origin is shown. In the right figure, the ellipse of an anisotropic spatial covariance function centered at the origin is shown. The black outline of each ellipse is a level curve of equal correlation.

$\mathbf{y} = \{y_1, y_2, y_3, y_4\}$. Also suppose that the random effect $\mathbf{u}$ has two levels and that y_1 and y_4 are in the first level and y_2 and y_3 are in the second level. For random intercepts, each element of $\mathbf{Z}$ is one if the observation is in the appropriate level of the random effect and zero otherwise. So it follows that

$$\mathbf{Zu} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

where u_1 and u_2 are the random intercepts for the first and second levels of $\mathbf{u}$, respectively. For random slopes, each element of $\mathbf{Z}$ equals the value of an auxiliary variable, $\mathbf{k}$, if the observation is in the appropriate level of the random effect and zero otherwise. So if $\mathbf{k} = \{2, 7, 5, 4\}$ it follows that

$$\mathbf{Zu} = \begin{bmatrix} 2 & 0 \\ 0 & 7 \\ 0 & 5 \\ 4 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

where u_1 and u_2 are the random slopes for the first and second levels of $\mathbf{u}$, respectively. If a random slope is included in the model, it is common for the auxiliary variable to be a column in $\mathbf{X}$, the fixed effects design matrix (i.e., also a fixed effect). Denote this column as $\mathbf{x}$. Here $\boldsymbol{\beta}$ captures the average effect of $\mathbf{x}$ on $\mathbf{y}$ (accounting for other explanatory variables) and $\mathbf{u}$ captures a subject-specific effect of $\mathbf{x}$ on $\mathbf{y}$. So for a subject in the ith level of $\mathbf{u}$, the average increase in y associated with a one-unit increase x is $\beta + u_i$.

The `sv-wls` and `sv-cl` estimation methods do not use a likelihood, and thus, they do not allow for the estimation of random effects in `spmodel`.

Anisotropy (`splm()` only)

An isotropic spatial covariance function behaves similarly in all directions (i.e., is independent of direction) as a function of distance. An anisotropic spatial covariance function does not behave similarly in all directions as a function of distance.

Figure 1 shows ellipses for an isotropic and anisotropic spatial covariance function centered at the origin (a distance of zero). The black outline of each ellipse is a level

Fig 2. In the left figure, the ellipse of an anisotropic spatial covariance function centered at the origin is shown. The blue lines represent the original axes and the red lines the transformed axes. The solid lines represent the x-axes and the dotted lines the y-axes. Note that the solid, red line is the major axis of the ellpise and the dashed, red line is the minor axis of the ellipse. In the center figure, the ellipse has been rotated clockwise by the rotate parameter so the major axis is the transformed x-axis and the minor axis is the transformed y-axis. In the right figure, the minor axis of the ellipse has been scaled by the reciprocal of the scale parameter so that the ellipse becomes a circle, which corresponds to an isotropic spatial covariance function. The transformed coordinates are then used to compute distances and spatial covariances.

curve of equal correlation. The left ellipse (a circle) represents an isotropic covariance function. The distance at which the correlation between two observations lays on the level curve is the same in all directions. The right ellipse represents an anisotropic covariance function. The distance at which the correlation between two observations lays on the level curve is different in different directions.

To accommodate spatial anisotropy, the original coordinates must be transformed such that the transformed coordinates yield an isotropic spatial covariance. This transformation involves a rotation and a scaling. Consider a set of x and y coordinates that should be transformed into x^* and y^* coordinates. This transformation is formally defined as

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/S \end{bmatrix} \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

The original coordinates are first multiplied by the rotation matrix, which rotates the coordinates clockwise by angle α. They are then multiplied by the scaling matrix, which scales the minor axis of the spatial covariance ellipse by the reciprocal of S. The transformed coordinates are then used to compute distances and the spatial covariances in Table 2. This type of anisotropy is more formally known as "geometric" anisotropy because it involves a geometric transformation of the coordinates. Figure 2 shows this process step-by-step.

Anisotropy parameters (α and S) can be estimated in `spmodel` using restricted maximum likelihood or maximum likelihood. Estimating anisotropy can be challenging. First, we need to restrict the parameter space so that the two parameters are identifiable (there is a unique parameter set for each possible outcome). We restricted α to $[0, \pi]$ radians due to symmetry of the covariance ellipse at rotations α and $\alpha + j\pi$, where j is any integer. We also restricted S to occur on $[0, 1]$ because we have defined S as the scaling factor for the length of the minor axis relative to the major axis –

otherwise it would not be clear whether S refers to the minor or major axis. Given this restricted parameter space, there is still an issue of local maxima, particularly at rotation parameters near zero, which have a rotation very close to rotation parameter π, but zero is far from π in the parameter space. To address the local maxima problem, each optimization iteration actually involves two likelihood evaluations – one for α and another for $|\pi - \alpha|$, where $|\cdot|$ denotes absolute value. Thus one likelihood evaluation is always in $[0, \pi/2]$ radians and another in $[\pi/2, \pi]$ radians, exploring different quadrants of the parameter space and allowing optimization to test solutions near zero and π simultaneously.

Anisotropy parameters cannot be estimated in `spmodel` when `estmethod` is `sv-wls` or `sv-cl`. However, known anisotropy parameters for these estimation methods can be specified via `spcov_initial` and incorporated into estimation of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. Anisotropy is not defined for areal data given its (binary) neighborhood structure.

Partition Factors

A partition factor is a factor (or categorical) variable in which observations from different levels of the partition factor are assumed uncorrelated. A partition matrix $\mathbf{P}$ of dimension $n \times n$ can be constructed to represent the partition factor. The ijth element of $\mathbf{P}$ equals one if the observation in the ith row and jth column are from the same level of the partition factor and zero otherwise. Then the initial covariance matrix (ignoring the partition factor) is updated by taking the Hadmard (element-wise) product with the partition matrix:

$$\boldsymbol{\Sigma}_{updated} = \boldsymbol{\Sigma}_{initial} \odot \mathbf{P},$$

where $\odot$ indicates the Hadmard product. Partition factors impose a block structure in $\boldsymbol{\Sigma}$, which allows for efficient computation of $\boldsymbol{\Sigma}^{-1}$ used for estimation and prediction.

When computing the empirical semivariogram using `esv()`, semivariances are ignored when observations are from different levels of the partition factor. For the `sv-wls` and `sv-cl` estimation methods, semivariances are ignored when observations are from different levels of the partition factor.

Big Data (`splm()` only)

Big data model-fitting is accommodated in `spmodel` using a "local indexing" approach. Suppose there are m unique indexes, and each observation is in one index. Then $\boldsymbol{\Sigma}$ can be represented blockwise as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} & \dots & \dots & \boldsymbol{\Sigma}_{1,m} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} & \boldsymbol{\Sigma}_{2,3} & \dots & \boldsymbol{\Sigma}_{2,m} \\ \vdots & \boldsymbol{\Sigma}_{3,2} & \ddots & \boldsymbol{\Sigma}_{3,4} & \vdots \\ \vdots & \vdots & \boldsymbol{\Sigma}_{4,3} & \ddots & \vdots \\ \boldsymbol{\Sigma}_{m,1} & \dots & \dots & \dots & \boldsymbol{\Sigma}_{m,m} \end{bmatrix}, \tag{13}$$

To perform estimation for big data, observations with the same index value are assumed independent of observations with different index values, yielding a "big-data" covariance matrix given by

$$\boldsymbol{\Sigma}_{bd} = \begin{bmatrix} \boldsymbol{\Sigma}_{1,1} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{2,2} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \vdots & \vdots & \mathbf{0} & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \dots & \boldsymbol{\Sigma}_{m,m} \end{bmatrix}, \tag{14}$$

Estimation then proceeds as described in Section using $\boldsymbol{\Sigma}_{bd}$ instead of $\boldsymbol{\Sigma}$. When computing the empirical semivariogram, semivariances are ignored when observations have different local indexes. For the `sv-wls` and `sv-cl` estimation methods, semivariances are ignored when observations have different local indexes. Via Equation 14, it can be seen that the local index acts as a partition factor separate from the partition factor explicitly defined by `partition_factor`.

spmodel allows for custom local indexes to be passed to `splm()`. If a custom local index is not passed, the local index is determined using the `"random"` or `"kmeans"` method. The `"random"` method assigns observations to indexes randomly based on the number of groups desired. The `"kmeans"` method uses k-means clustering [30] on the x-coordinates and y-coordinates to assign observations to indexes (based on the number of clusters (groups) desired).

The estimate of $\boldsymbol{\beta}$ when using Equation 14 is given by

$$\hat{\boldsymbol{\beta}}_{bd} = (\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}_{bd}^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \hat{\boldsymbol{\Sigma}}_{bd}^{-1} \mathbf{y} = \mathbf{T}_{xx}^{-1} \mathbf{t}_{xy}, \tag{15}$$

where $\mathbf{T}_{xx} = \sum_{i=1}^{m} \mathbf{X}_i^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{X}_i$ and $\mathbf{t}_{xy} = \sum_{i=1}^{m} \mathbf{X}_i^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{y}_i$. Note that in $\hat{\boldsymbol{\beta}}_{bd}$, $\mathbf{X}_i$ and $\mathbf{y}_i$ are the subsets of $\mathbf{X}$ and $\mathbf{y}$, respectively, for the ith local index. Equation 15 acts as a pooled estimator of $\boldsymbol{\beta}$ across the indexes.

`spmodel` has four approaches for estimating the covariance matrix of $\hat{\boldsymbol{\beta}}_{bd}$. The choice is determined by the `var_adjust` argument to `local`. The first approach is implements no adjustment (`var_adjust = "none"`) and simply uses $\mathbf{T}_{xx}^{-1}$, which is the covariance matrix of $\hat{\boldsymbol{\beta}}_{bd}$ using $\boldsymbol{\Sigma}_{bd}$ (Equation 14). While computationally efficient, this approach ignores the covariance across indexes. It can be shown that the covariance of $\hat{\boldsymbol{\beta}}_{bd}$ using $\boldsymbol{\Sigma}$ (Equation 13) is given by

$$\mathbf{T}_{xx}^{-1} + \mathbf{T}_{xx}^{-1} \mathbf{W}_{xx} \mathbf{T}_{xx}^{-1}, \tag{16}$$

where

$$\mathbf{W} = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \hat{\boldsymbol{\Sigma}}_{i,j} \hat{\boldsymbol{\Sigma}}_{j,j}^{-1} \mathbf{X}_j) + (\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \hat{\boldsymbol{\Sigma}}_{i,j} \hat{\boldsymbol{\Sigma}}_{j,j}^{-1} \mathbf{X}_j)^\top$$

Equation 16 can be viewed as the sum of the unadjusted covariance matrix of $\hat{\boldsymbol{\beta}}_{bd}$ ($\mathbf{T}_{xx}^{-1}$) and a correction that incorporates the covariance across indexes ($\mathbf{T}_{xx}^{-1} \mathbf{W}_{xx} \mathbf{T}_{xx}^{-1}$). This adjustment is known as the "theoretically-correct" (`var_adjust = "theoretical"`) adjustment because it uses $\boldsymbol{\Sigma}$. The theoretical adjustment is the default adjustment in `spmodel` because it is theoretically correct, but it is the most computationally expensive adjustment. Two alternative adjustments are also provided, and while not equal to the theoretical adjustment, they are easier to compute. They are the empirical (`var_adjust = "empirical"`) and pooled (`var_adjust = "pooled"`) adjustments. The empirical adjustment is given by

$$\frac{1}{m(m-1)} \sum_{i=1}^{m} (\hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{\beta}}_{bd})(\hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{\beta}}_{bd})^\top,$$

where $\hat{\boldsymbol{\beta}}_i = (\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1} \mathbf{X}_i^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{y}_i$. A similar adjustment could use $\hat{\boldsymbol{\beta}}_i = (\mathbf{X}_i^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{X}_i)^{-1} \mathbf{X}_i \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{y}_i$, which more closely resembles a composite likelihood approach. This approach is sensitive to the presence of at least one singularity in $\mathbf{X}_i^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{X}_i$, in which case the variance adjustment cannot be computed. The `"pooled"` variance adjustment is given by

$$\frac{1}{m^2} \sum_{i=1}^{m} (\mathbf{X}_i^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{X}_i)^{-1}.$$

Note that the pooled variance adjustment cannot be computed if any $\mathbf{X}_i^\top \hat{\boldsymbol{\Sigma}}_{i,i}^{-1} \mathbf{X}_i$ are singular.

sprnorm()

Spatial normal (Gaussian) random variables are simulated by taking the sum of a fixed mean and random errors. The random errors have mean zero and covariance matrix $\boldsymbol{\Sigma}$. A realization of the random errors is obtained from $\boldsymbol{\Sigma}^{1/2}\mathbf{e}$, where $\mathbf{e}$ is a normal random variable with mean zero and covariance matrix $\mathbf{I}$. Then the spatial normal random variable equals

$$\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\mathbf{e},$$

where $\boldsymbol{\mu}$ is the fixed mean. It follows that

$$\begin{aligned} \mathrm{E}(\mathbf{y}) &= \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\mathrm{E}(\mathbf{e}) = \boldsymbol{\mu} \\ \mathrm{Cov}(\mathbf{y}) &= \mathrm{Cov}(\boldsymbol{\Sigma}^{1/2}\mathbf{e}) = \boldsymbol{\Sigma}^{1/2}\mathrm{Cov}(\mathbf{e})\boldsymbol{\Sigma}^{1/2} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{1/2} = \boldsymbol{\Sigma} \end{aligned}$$

vcov()

`vcov()` returns the variance-covariance matrix of estimated parameters. Currently, `vcov()` only returns the variance-covariance matrix of $\hat{\boldsymbol{\beta}}$, the fixed effects. The variance-covariance matrix of the fixed effects is given by $(\mathbf{X}^\top\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{X})^{-1}$.

A Note on Covariance Square Roots and Inverse Products

Often $\boldsymbol{\Sigma}^{-1}$ is not strictly needed for estimation, prediction, or other purposes, but at least the product between $\boldsymbol{\Sigma}^{-1}$ and some other matrix is needed. Consider the example of the covariance matrix of $\hat{\boldsymbol{\beta}}$ and observe $\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}$ is needed. The most direct way to find this product is certainly to obtain $\boldsymbol{\Sigma}^{-1}$ and then multiply by $\mathbf{X}^\top$ on the left and $\mathbf{X}$ on the right. This is both computationally expensive and cannot be used to compute products that involve $\boldsymbol{\Sigma}^{-1/2}$, which are often useful (Section). It is helpful to rewrite $\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}$ as $\mathbf{X}^\top(\mathbf{S}^\top)^{-1}\mathbf{S}^{-1}\mathbf{X} = (\mathbf{S}^{-1}\mathbf{X})^\top\mathbf{S}^{-1}\mathbf{X}$. Then one computes the inverse products by finding $\mathbf{S}$.

One way to find $\mathbf{S}$ is to use an eigendecomposition. The eigendecomposition of $\boldsymbol{\Sigma}$ (which is real and symmetric) is given by

$$\boldsymbol{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^\top,$$

where $\mathbf{U}$ is an orthogonal matrix of eigenvectors of $\boldsymbol{\Sigma}$ and $\mathbf{D}$ is a diagonal matrix with eigenvalues of $\boldsymbol{\Sigma}$ on the diagonal. Then $\boldsymbol{\Sigma}^{1/2} = \mathbf{U}\mathbf{D}^{1/2}\mathbf{U}^\top$, where $\mathbf{D}^{1/2}$ is a diagonal matrix with square roots of eigenvalues of $\boldsymbol{\Sigma}$ on the diagonal. This result follows because $\mathbf{U}$ being orthogonal implies $\mathbf{U}^\top = \mathbf{U}^{-1}$ and

$$\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{1/2} = \mathbf{U}\mathbf{D}^{1/2}\mathbf{U}^\top\mathbf{U}\mathbf{D}^{1/2}\mathbf{U}^\top = \mathbf{U}\mathbf{D}^{1/2}(\mathbf{U}^\top\mathbf{U})\mathbf{D}^{1/2}\mathbf{U}^\top = \mathbf{U}\mathbf{D}\mathbf{U}^\top = \boldsymbol{\Sigma}.$$

So then taking $\mathbf{S} = \mathbf{D}^{1/2}$ implies $\mathbf{S}^{-1} = \mathbf{D}^{-1/2}$, which is straightforward to calculate as $\mathbf{D}^{1/2}$ is diagonal. So not only does the eigendecomposition approach give us the inverse products, it also gives us $\boldsymbol{\Sigma}^{1/2}$ and $\boldsymbol{\Sigma}^{-1/2}$. While straightforward, this approach is less efficient than the Cholesky decomposition [31], which we discuss next.

The Cholesky decomposition decomposes $\boldsymbol{\Sigma}$ into the product between $\mathbf{C}$ and $\mathbf{C}^\top$ ($\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^\top$), where $\mathbf{C}$ is a lower triangular matrix. Note that $\mathbf{C}$ is generally not equal to $\boldsymbol{\Sigma}^{1/2}$. Taking $\mathbf{S}$ to be $\mathbf{C}$, we see that finding the inverse products requires solving $\mathbf{C}^{-1}\mathbf{X}$. Observe that $\mathbf{C}^{-1}\mathbf{X} = \mathbf{A}$ for some matrix $\mathbf{A}$. This implies $\mathbf{X} = \mathbf{C}\mathbf{A}$, which for $\mathbf{A}$ can be efficiently solved using forward substitution because $\mathbf{C}$ is lower triangular.

The products in this document that involve $\boldsymbol{\Sigma}^{1/2}$ and $\boldsymbol{\Sigma}^{-1/2}$ are actually implemented in `spmodel` using $\mathbf{C}$ and $\mathbf{C}^{-1}$ (instead of $\boldsymbol{\Sigma}^{1/2}$ and $\boldsymbol{\Sigma}^{-1/2}$). They are written in this document using $\boldsymbol{\Sigma}^{1/2}$ and $\boldsymbol{\Sigma}^{-1/2}$ because the underlying concepts are easier to communicate using square root notation.

References

1. Hoeting JA, Davis RA, Merton AA, Thompson SE. Model selection for geostatistical models. Ecological Applications. 2006;16: 87–98.

2. Kackar RN, Harville DA. Approximations for standard errors of estimators of fixed and random effects in mixed linear models. Journal of the American Statistical Association. 1984;79: 853–862.

3. Prasad NN, Rao JN. The estimation of the mean squared error of small-area estimators. Journal of the American statistical association. 1990;85: 163–171.

4. Harville DA, Jeske DR. Mean squared error of estimation or prediction under a general linear model. Journal of the American Statistical Association. 1992;87: 724–731.

5. Kenward MG, Roger JH. Small sample inference for fixed effects from restricted maximum likelihood. Biometrics. 1997; 983–997.

6. Satterthwaite FE. An approximate distribution of estimates of variance components. Biometrics bulletin. 1946;2: 110–114.

7. Schluchter MD, Elashoff JT. Small-sample adjustments to tests with unbalanced repeated measures assuming several covariance structures. Journal of Statistical Computation and Simulation. 1990;37: 69–87.

8. Hrong-Tai Fai A, Cornelius PL. Approximate f-tests of multiple degree of freedom hypotheses in generalized least squares analyses of unbalanced split-plot experiments. Journal of statistical computation and simulation. 1996;54: 363–378.

9. Littell RC, Milliken GA, Stroup WW, Wolfinger RD, Oliver S. SAS for mixed models. SAS publishing; 2006.

10. Pinheiro J, Bates D. Mixed-effects models in s and s-plus. Springer science & business media; 2006.

11. Kenward MG, Roger JH. An improved approximation to the precision of fixed effects from restricted maximum likelihood. Computational Statistics & Data Analysis. 2009;53: 2583–2595.

12. Myers RH, Montgomery DC, Vining GG, Robinson TJ. Generalized linear models: With applications in engineering and the sciences. John Wiley & Sons; 2012.

13. Self SG, Liang K-Y. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. Journal of the American Statistical Association. 1987;82: 605–610.

14. Stram DO, Lee JW. Variance components testing in the longitudinal mixed effects model. Biometrics. 1994; 1171–1177.

15. Goldman N, Whelan S. Statistical tests of gamma-distributed rate heterogeneity in models of sequence evolution in phylogenetics. Molecular Biology and Evolution. 2000;17: 975–978.

16. Cook RD. Influential observations in linear regression. Journal of the American Statistical Association. 1979;74: 169–174.

17. Cook RD, Weisberg S. Residuals and influence in regression. New York: Chapman; Hall; 1982.

18. Montgomery DC, Peck EA, Vining GG. Introduction to linear regression analysis. John Wiley & Sons; 2021.

19. Cressie N. Statistics for spatial data. John Wiley & Sons; 1993.

20. Henderson CR. Best linear unbiased estimation and prediction under a selection model. Biometrics. 1975; 423–447.

21. Searle SR, Casella G, McCulloch CE. Variance components. John Wiley & Sons; 2009.

22. Wolf H. The helmert block method-its origin and development. Proceedings of the second international symposium on problems related to the redefinition of north american geodetic networks,(NOAA, arlington-va, 1978). 1978. pp. 319–326.

23. Patterson D, Thompson R. Recovery of inter-block information when block sizes are unequal. Biometrika. 1971;58: 545–554.

24. Harville DA. Maximum likelihood approaches to variance component estimation and to related problems. Journal of the American Statistical Association. 1977;72: 320–338.

25. Wolfinger R, Tobias R, Sall J. Computing gaussian likelihoods and their derivatives for general linear mixed models. SIAM Journal on Scientific Computing. 1994;15: 1294–1310.

26. Cressie N. Fitting variogram models by weighted least squares. Journal of the international Association for mathematical Geology. 1985;17: 563–586.

27. Curriero FC, Lele S. A composite likelihood approach to semivariogram estimation. Journal of Agricultural, biological, and Environmental statistics. 1999; 9–28.

28. Nelder JA, Mead R. A simplex method for function minimization. The computer journal. 1965;7: 308–313.

29. Brent RP. An algorithm with guaranteed convergence for finding a zero of a function. The Computer Journal. 1971;14: 422–425.

30. MacQueen J, others. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth berkeley symposium on mathematical statistics and probability. Oakland, CA, USA; 1967. pp. 281–297.

31. Golub GH, Van Loan CF. Matrix computations. JHU press; 2013.