

# **svc: Spatially Varying Coefficient Models in R**

Justice Akuoko-Frimpong, Edward Shao, Jonathan Ta

## **1 Introduction**

Heterogeneity in spatial processes presents significant challenges for traditional regression modeling approaches. Consider air pollution monitoring across an urban-rural gradient: the relationship between particulate matter (PM2.5) concentrations and predictor variables like traffic density or industrial emissions often varies substantially across space. Global regression models that assume constant coefficients fail to capture these localized relationships, potentially leading to incorrect scientific conclusions and suboptimal policy decisions.

Our work develops a computationally efficient Bayesian implementation of spatially varying coefficient (SVC) models, which address this limitation by allowing regression coefficients to change smoothly over space.

This flexible approach captures complex, location-dependent relationships while accounting for measurement error through an independent noise term. Our computational framework combines three innovations to enable efficient Bayesian inference: (1) a low-rank Gaussian process approximation for improved efficiency; (2) a robust adaptive Metropolis (RAM) to automatically achieve optimal acceptance rate for spatial decay random walk sampling; and (3) optimized linear algebra operations using pre-computed distance matrices and Cholesky decompositions to accelerate covariance calculations. Together, these advances SVC models practical for large-scale environmental datasets.

## 2 Methods

### 2.1 SVC Model

Let  $\mathcal{D} \subset \mathbb{R}^2$  denote a spatial domain,  $s_i \in \mathcal{D}$  for each  $i = 1, \dots, n$  denote a spatial location with collected data, and  $\mathbf{s} = (s_1, \dots, s_n)^\top$  be the vector of all such locations. Then  $Y(\mathbf{s})$  are univariate dependent variables and  $\mathbf{X}(\mathbf{s}) = (X_1(\mathbf{s}), \dots, X_p(\mathbf{s}))^\top$  are  $p \times 1$  vectors of covariates. A linear SVC regression model assumes  $Y(\mathbf{s})$  are dependent on  $\mathbf{X}(\mathbf{s})$  as follows:

$$Y(\mathbf{s}) = \sum_{j=1}^p X_j(\mathbf{s})w_j(\mathbf{s}) + \epsilon(\mathbf{s}),$$

where  $w_r(\mathbf{s})$  are the SVC's corresponding to  $X_r(\mathbf{s})$  and  $\epsilon(\mathbf{s})$  are independently and identically distributed multivariate normal measurement errors, i.e.  $\epsilon(\mathbf{s}) \sim \text{MVN}(0, \tau^2 I_n)$ . Note that spatially-varying intercept can be included by defining  $X_1(\mathbf{s})$  as  $\mathbf{1}_n$ , a vector of length  $n$  with every element equal to 1.

For each  $w_r(\mathbf{s})$ , we assign a Gaussian process (GP) prior with squared exponential covariance, i.e.  $w_r \sim \text{GP}(0, C(\boldsymbol{\theta}_r))$  where  $C(\boldsymbol{\theta}_r) = [C(s_i, s_{i'}; \boldsymbol{\theta}_r)]_{i,i'=1}^n$ . We refer to  $C(\cdot)$  as the covariance function, because the covariance of  $w_r(s_i)$  and  $w_r(s_{i'})$  at locations  $s_i$  and  $s_{i'}$  is  $\sigma_r^2 C(\boldsymbol{\theta}_r)_{i,i'}$ . In particular, we use the squared exponential covariance function where  $\boldsymbol{\theta}_r = (\sigma_r^2, \phi_r)$ ,

$$C(\boldsymbol{\theta}_r) = \sigma_r^2 K(\phi_r), \text{ and}$$

$$K(\phi_r) = [K(s_i, s_{i'}; \phi_r)]_{i,i'=1}^n = [\exp\{\phi_r^{-1} \|s_i - s_{i'}\|^2\}]_{i,i'=1}^n,$$

due to its useful properties such as infinite smoothness and gradual decrease in covariance with distance. The parameter  $\sigma_r^2$  is the spatial variance and  $\phi_r$  is the spatial decay, which indicates how quickly correlation decreases with the squared distance.

We assign inverse-gamma conjugate priors for the variance parameters  $\tau^2$  and each  $\sigma_r^2$ .

In addition, each  $\phi_r$  is assigned a uniform prior. In summary,  $\tau^2 \sim \text{Inv.Gamma}(\alpha_\tau, \beta_\tau)$ ,  $\sigma_r^2 \sim \text{Inv.Gamma}(\alpha_r, \beta_r)$ , and  $\phi_r \sim \text{Uniform}(l_r, u_r)$ .

## 2.2 MCMC Algorithm

The primary function in the `svc` package, `svclm`, samples from the joint posterior distribution of  $w_r(\mathbf{s})$ ,  $\phi_r$ ,  $\sigma_r^2$ , and  $\tau^2$ , for  $r = 1, \dots, p$  using a Gibbs sampler. We will explain how each parameter is sampled at iteration  $t + 1$ .

The parameter  $\phi_r^{(t+1)}$  is updated via a random walk Metropolis algorithm. Because  $\phi_r^{(t)} \in (l_r, u_r)$ , each proposal is calculated as

$$\phi_r' = f^{-1}(f(\phi_r^{(t)}) + U) = l_r + \frac{u_r - l_r}{1 + \exp \left\{ \log \left( \frac{u_r - l_r}{\phi_r^{(t)} - l_r} - 1 \right) + U \right\}},$$

where  $U$  is sampled from a normal distribution with mean 0.

The initial standard deviation for  $U$  is specified using the "phi\_proposal\_sd\_start" argument or is 1 by default. Then the standard deviation is updated at each iteration via a Robust Adaptive Metropolis (RAM) algorithm as described by [Vihola \(2012\)](#).

The sample the SVC's, let  $\tilde{Y}(\mathbf{s}) = Y(\mathbf{s}) - \sum_{j \neq r} X_j(\mathbf{s})w_j^{(t)}(\mathbf{s})$ . Then

$$\frac{\tilde{Y}}{X_r}(\mathbf{s}) | w_r^{(t)}(\mathbf{s}), \tau^2 \sim \text{MVN} \left( w_r^{(t)}(\mathbf{s}), \tau^{2(t)} \text{diag} \left( \frac{1}{X_r^2(\mathbf{s})} \right) \right) := \text{MVN}(\mu, \Sigma).$$

Then assuming the prior  $w_r(\mathbf{s}) \sim \text{MVN}(0, C(\boldsymbol{\theta}_r^{(t)})) := \text{MVN}(0, \Sigma_0)$ , we sample  $w_r^{(t+1)}(\mathbf{s})$  from  $\text{MVN}(\mu_1, \Sigma_1)$ , where

$$\Sigma_1 = (\Sigma_0^{-1} + \Sigma^{-1}) \text{ and } \mu_1 = \Sigma_1 \Sigma^{-1} \frac{\tilde{Y}}{X_r}(\mathbf{s}).$$

We sample  $\sigma_r^{2(t+1)}$  from  $\text{Inv.Gamma} \left( \alpha_r + \frac{n}{2}, \beta_r + w_r^{(t)}(\mathbf{s})^\top K(\phi_r^{(t)})^{-1} w_r^{(t)}(\mathbf{s}) \right)$ .

We sample  $\tau^{2(t+1)}$  from  $\text{Inv.Gamma} \left( \alpha_t + \frac{n}{2}, \beta_t + \frac{1}{2} \left( Y(\mathbf{s}) - \sum_{j=0}^p X_r(\mathbf{s})w_r^{(t)}(\mathbf{s}) \right) \right)$ .

### 2.2.1 Low-Rank Gaussian Process

We can perform the above algorithm using the complete data for all  $n$  locations to estimate the full-rank Gaussian process model. However, notice that in each iteration, we must invert  $C(\boldsymbol{\theta}_r^{(t)})$  for  $r = 1, \dots, p$ . In order to do so, we perform a Cholesky decomposition to obtain an upper triangular matrix  $R$  and then invert  $R$ . These Cholesky decompositions on  $p$  symmetric positive definite matrices of size  $(n \times n)$  are the most flop-expensive operations in the algorithm resulting in an overall cost of  $O(pn^3)$  per iteration. We can dramatically improve the efficiency of this algorithm with a low-rank Gaussian process as described by [Banerjee et al. \(2008\)](#).

Consider a set of knots  $\mathbf{s}^* = (s_1^*, \dots, s_m^*)^\top$  which are a subset of locations  $\mathbf{s}$ . With the low-rank Gaussian process, we sample  $\tau^2$ ,  $\phi_r$ ,  $\sigma_r^2$ , and  $\mathbf{w}_r^* = [w(\mathbf{s}_i^*)]_{i=1}^m$  which follows a multivariate Gaussian distribution with covariance matrix  $C^*(\boldsymbol{\theta}_r) = \sigma_r^2 K^*(\phi_r) = [\sigma_r^2 \exp\{\phi_r^{-1} \|s_i^* - s_{i'}^*\|^2\}]_{i,i'=1}^m$ , for each  $r = 1, \dots, p$ .

Then for the original set of locations  $\mathbf{s}$ , we predict the spatial interpolants

$$\tilde{w}_r(\mathbf{s}) = \mathbb{E}[w_r(\mathbf{s}) | \mathbf{w}_r^*] = \mathbf{c}(\mathbf{s}; \boldsymbol{\theta}_r) C^{*-1}(\boldsymbol{\theta}_r) \mathbf{w}_r^*,$$

where  $\mathbf{c}(\mathbf{s}; \boldsymbol{\theta}_r)$  is an  $n \times m$  matrix with  $i, j$  element  $c_{i,j} = \sigma_r^2 K(s_i, s_j^*; \phi_r) = \sigma_r^2 \exp\{\phi_r^{-1} \|s_i - s_j^*\|^2\}$ .

Therefore, using the low-rank Gaussian process, we can replace  $w_r(\mathbf{s})$  in the original model with  $\tilde{w}_r(\mathbf{s})$  for each  $r = 1, \dots, p$ , but with an overall cost per iteration of  $O(pm^3)$ , where  $m < n$ .

### 3 Using `svc`

To fit a linear SVC regression model using `svc`, we use the `svclm` function. In most cases, we recommend providing the following arguments to `svclm`: `Y`, `X`, `coords`, `Y_knots`, `X_knots`, `knots`, `phi_lower`, `phi_upper`, and `mcmc`. `Y` is a vector of length  $n$  containing the response variable, `X` is the design matrix of size  $n \times p$ , and `coords` is a matrix of size  $n \times 2$  containing the spatial coordinates of each location. The arguments `Y_knots`, `X_knots`, and `knots` provide the data for the knots that will be used in the low-rank model. `Y_knots` is a vector of length  $m$  containing the response variable at the knots, `X_knots` is a matrix of size  $m \times p$  containing the covariates at the knots, and `knots` is a matrix of size  $m \times 2$  containing the spatial coordinates of each knot. The arguments `phi_lower` and `phi_upper` are each a vector of length  $p$  containing the lower and upper bounds for the uniform distribution priors assigned to spatial decay parameters  $\phi_r$ , respectively. The argument `mcmc` is an integer specifying the number of MCMC iterations to run.

The minimum required arguments to run `svclm` are `Y`, `X`, `coords`, `phi_lower`, and `phi_upper`. In this case, the function will use the same data for the knots as for the full model essentially fitting the full Gaussian process model. The default number of MCMC iterations is 1000.

We will not go into detail here since the arguments above should be sufficient for most use cases, but the `svclm` function does allow other arguments for additional settings. Users can specify starting values for  $\tau^2$ ,  $\sigma_r^2$ ,  $\phi_r$ , and  $\mathbf{w}_r^*$  where  $r = 1, \dots, p$ . They can also set the hyperparameters for the inverse gamma priors on  $\tau^2$  and  $\sigma_r^2$ . Lastly, users can set the starting proposal standard deviations for each  $\phi_r$  and the target acceptance rate for the RAM.

By default, the starting values for  $\tau^2$ ,  $\sigma_r^2$ ,  $\phi_r$ , and  $\mathbf{w}_r^*$  are set to 1, 1, the midpoint between

the corresponding upper and lower bounds, and 0 for all locations, respectively. The default priors for  $\tau^2$  and each  $\sigma_r^2$  are all  $\text{Inv.Gamma}(0.001, 0.001)$  to achieve uninformative priors. The starting proposal standard deviations for each  $\phi_r$  are set to 1 by default, and the target acceptance rate for the RAM algorithm is set to 0.234 which is the asymptotically optimal acceptance rate under most conditions (Gelman et al. 1997).

In all cases, `svclm` returns a list containing the following matrices: `w_samples`, an  $(\text{mcmc} \times n \times p)$  array of the posterior samples for the SVC's; `phi_samples`, an  $(\text{mcmc} \times p)$  matrix of the posterior samples for the spatial decay parameters; `phi_acceptance`, an  $(\text{mcmc} \times p)$  matrix which is 1 if the proposal for  $\phi_r$  was accepted and 0 otherwise; `sigmasq_samples`, an  $(\text{mcmc} \times p)$  matrix of the posterior samples for the spatial variances; and `tausq_samples`, a  $(\text{mcmc} \times 1)$  matrix of the posterior samples for the nugget parameter.

Although knot data can be chosen manually, the `svc` package also contains a function `simpleknots` to automatically select knots for the low-rank model. The function takes arguments `Y`, `X`, `coords`, and `k`, where `k` is an integer and `Y`, `X`, and `coords` match what would be provided to the `svclm` function. The function will return a list containing the objects `Y_knots`, `X_knots`, and `knots` which can then be passed directly to the `svclm` function. The knots are selected by sorting each dimension of coordinates and then selecting every  $k$ -th coordinate in each dimension.

## 4 Simulation

We compared the performance of our proposed method with `spBayes` using bias, RMSE, and computational time (in seconds), averaged over 1000 simulation replications. The true values for  $\beta_1$ ,  $\beta_2$ , and  $w$  were all set to 1.

Table 1: Comparison of methods based on bias, RMSE, and computation time (seconds).  
Placeholder values shown.

| Method                | $\beta_1$ |      |      | $\beta_2$ |      |      | $w$  |      |      |
|-----------------------|-----------|------|------|-----------|------|------|------|------|------|
|                       | Bias      | RMSE | Time | Bias      | RMSE | Time | Bias | RMSE | Time |
| <b>svc</b> (low rank) | -4.36     | 9.72 | 60.5 | -3.34     | 3.68 | 60.5 | 1.15 | 2.11 | 60.5 |
| <b>spBayes</b>        | -0.01     | 0.13 | 408  | -0.03     | 0.07 | 408  | 0.05 | 0.11 | 408  |

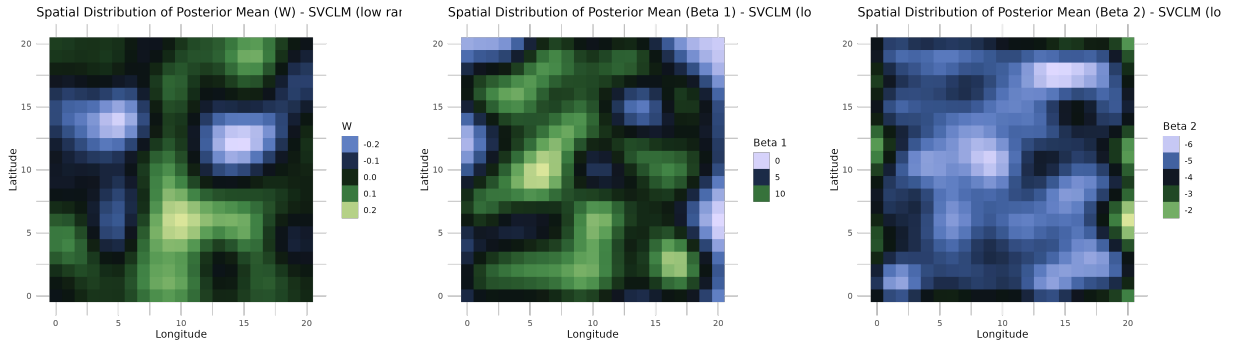


Figure 1: Posterior Mappings for Coefficients - SVCLM low rank

## 5 Data Analysis

The real world data used in this section is a gridded satellite data with latitude, longitude, temperature, Normalized Difference Vegetation Index (NDVI); which measures greenness, and emissivity which represents a surface’s efficiency in emitting thermal radiation (Hulley & Hook 2008, Hulley et al. 2008, Hulley & Hook 2009, Hulley et al. 2009, Hulley & Hook 2011, Hulley et al. 2012, NASA JPL 2014, Hulley et al. 2015).

We used temperature as a univariate outcome and used emissivity and NDVI as covariates. Figure 3 below reveals the spatial distributions of the variables in the dataset.

Emissivity ranged from 0.88 to 0.96. Cooler regions often align with greener (higher NDVI) areas — vegetation can reduce surface temperature through evapotranspiration.

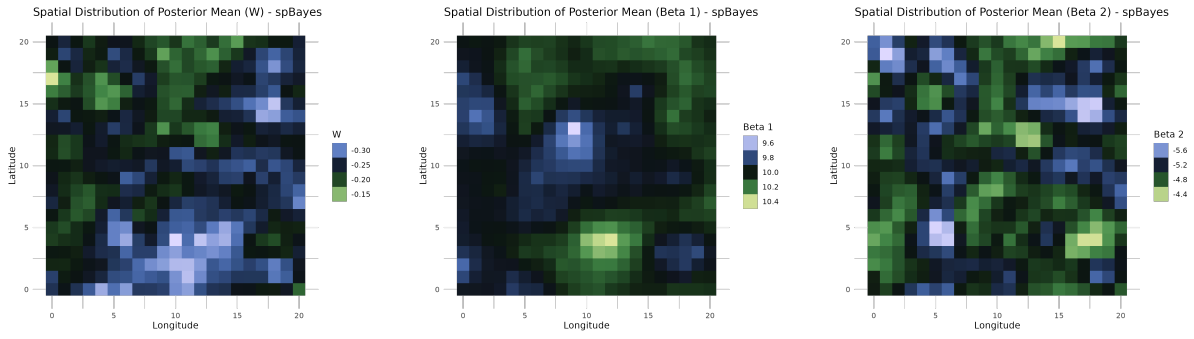


Figure 2: Posterior Mappings for Coefficients - spBayes

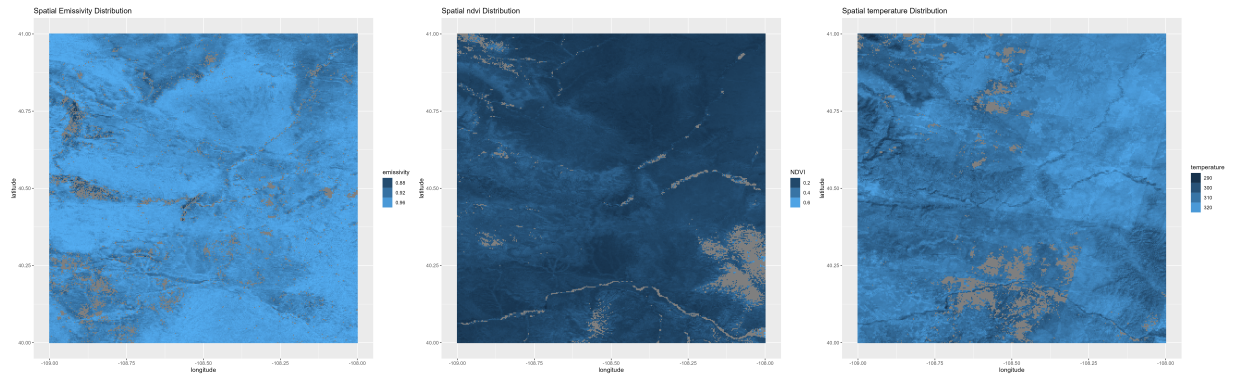


Figure 3: Spatial distributions of emissivity, NDVI, and land surface temperature in the study area.

## 5.1 Model

We modeled land surface temperature (LST) using an SVC framework:

$$\text{Temp}(\mathbf{s}) = w_1(\mathbf{s}) + w_2(\mathbf{s}) \cdot \text{NDVI} + w_3(\mathbf{s}) \cdot \text{Emissivity} + \epsilon(\mathbf{s}),$$

with priors as specified in 2.1.

We specified weakly informative priors:  $\tau^2 \sim \text{Inv-Gamma}(0.001, 0.001)$ ,  $\sigma_r^2 \sim \text{Inv-Gamma}(0.001, 0.001)$ , and  $\phi_r \sim \text{Uniform}(0.001, 500)$ , for each  $r = 1, \dots, 3$ .

The computational implementation featured low-rank approximation. We selected 1 knot per 100 grid cells ( $k = 10$ ) using the `simpleknots()` function. Removed knots with missing data.



## 5.2 Results

### Convergence Diagnostics

We made trace plots after burn-in to check for convergence.

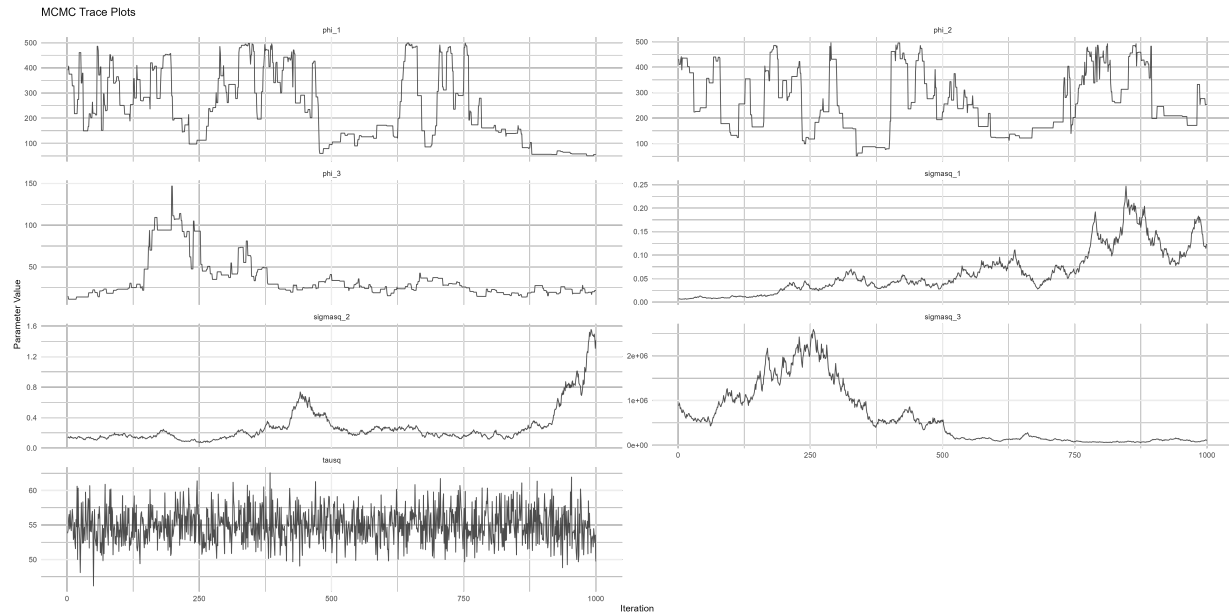


Figure 4: Trace plots for the parameters in the model

Some variables display stable and well-mixed behavior, suggesting convergence, while others show trends or irregular jumps which was expected in our setting.

### Interpretation of Coefficient Surfaces

Below is a plot of the posterior means of the coefficients over space.

The posterior mean NDVI coefficients show strong negative effects (cooling) across the entire study region, with values ranging from  $-0.7310$  to  $-0.7300$ . Each unit increase in NDVI corresponds to about  $0.73^{\circ}C$  decrease in land surface temperature across space. The emissivity coefficients reveal strong positive associations (mean  $w_3 \approx 330$ ) which implies that across space, a unit increase in emissivity increases temperature.

We generated predicted temperature values across the entire spatial domain using the posterior mean estimates of the SVC's. These predictions were computed for all locations

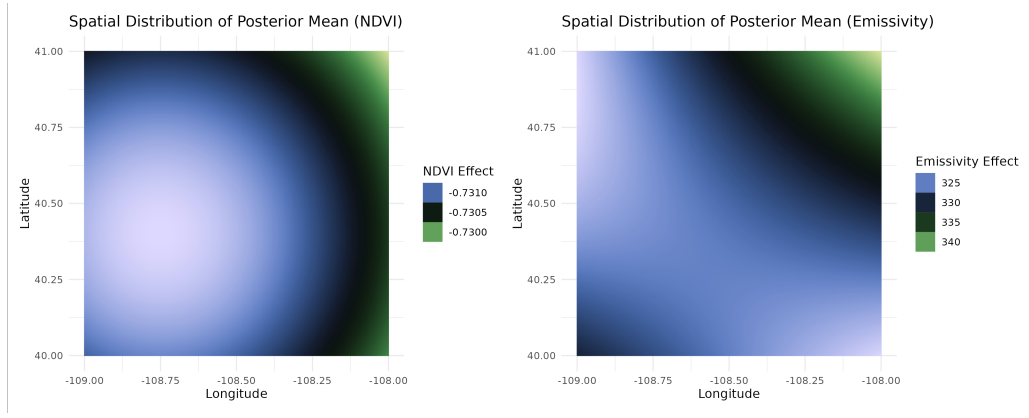


Figure 5: Spatially varying coefficient surfaces for (A) NDVI and (B) Emissivity effects on land surface temperature

with complete predictor data, regardless of whether the actual temperature was observed. The predicted values were obtained by multiplying the mean coefficient estimates with the corresponding covariate values at each location. We then visualized both the observed and predicted temperature surfaces side by side to assess model performance and spatial prediction accuracy.

The comparison reveals the model’s ability to capture spatial patterns in temperature, even in regions where observations were missing but predictor data were available.

## 6 Conclusion

The **svc** package provides a flexible and efficient framework for fitting SVC models that capture heterogeneous spatial processes. By leveraging low-rank Gaussian process approximations, RAM adaptation, and various sampling optimizations, we have made SVC models more computationally feasible for large-scale spatial datasets.

Compared to **spBayes**, our method performs.....

One limitation of our method is that it requires complete data at knot locations, whereas

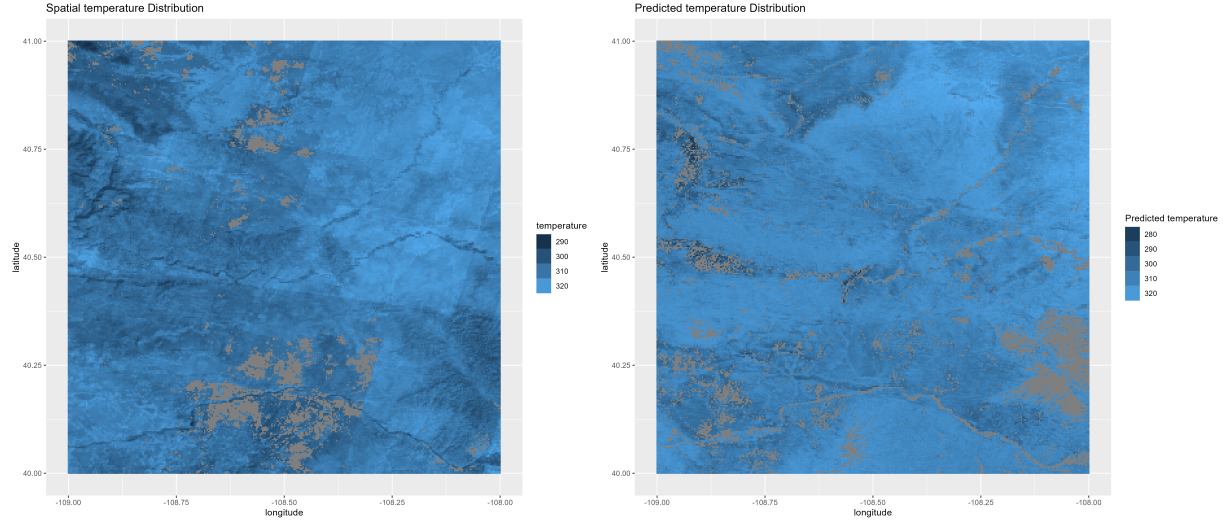


Figure 6: Side by side plot of temperature and their predicted values across space. Gray indicates missing values.

**spBayes** does not require any data at selected knot locations.

Further improvements to the package could include allowing for different covariance functions and implementing parallelization for sampling  $\phi_r$  and  $\sigma_r^2$  for  $r = 1, \dots, p$  since the sampling can be done independently for each  $r$ .

## References

Banerjee, S., Gelfand, A. E., Finley, A. O. & Sang, H. (2008), ‘Gaussian predictive process models for large spatial data sets’, **70**(4), 825–848.

**URL:** <https://academic.oup.com/jrsssb/article/70/4/825/7109503> 4

Gelman, A., Gilks, W. R. & Roberts, G. O. (1997), ‘Weak convergence and optimal scaling of random walk metropolis algorithms’, **7**(1).

**URL:** <https://projecteuclid.org/journals/annals-of-applied-probability/volume-7/issue-1/Weak-convergence-and-optimal-scaling-of-random-walk-Metropolis-algorithms/10.1214/aoap/1034625254.full> 6

Hulley, G. C. & Hook, S. J. (2008), ‘A new methodology for cloud detection and classification with ASTER data’, **35**(16).

**URL:** <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2008GL034644> 7

Hulley, G. C. & Hook, S. J. (2009), ‘The north american ASTER land surface emissivity database (NAALSED) version 2.0’, **113**(9), 1967–1975.

**URL:** <https://linkinghub.elsevier.com/retrieve/pii/S0034425709001631> 7

Hulley, G. C. & Hook, S. J. (2011), ‘Generating consistent land surface temperature and emissivity products between ASTER and MODIS data for earth science research’, **49**(4), 1304–1315.

**URL:** <http://ieeexplore.ieee.org/document/5582278/> 7

Hulley, G. C., Hook, S. J., Abbott, E., Malakar, N., Islam, T. & Abrams, M. (2015), ‘The ASTER global emissivity dataset ( ASTER GED ): Mapping earth’s

emissivity at 100 meter spatial scale’, **42**(19), 7966–7976.

**URL:** <https://agupubs.onlinelibrary.wiley.com/doi/10.1002/2015GL065564> 7

Hulley, G. C., Hook, S. J. & Baldrige, A. M. (2008), ‘ASTER land surface emissivity database of california and nevada’, **35**(13), 2008GL034507.

**URL:** <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2008GL034507> 7

Hulley, G. C., Hook, S. J. & Baldrige, A. M. (2009), ‘Validation of the north american ASTER land surface emissivity database (NAALSED) version 2.0 using pseudo-invariant sand dune sites’, **113**(10), 2224–2233.

**URL:** <https://linkinghub.elsevier.com/retrieve/pii/S0034425709001849> 7

Hulley, G. C., Hughes, C. G. & Hook, S. J. (2012), ‘Quantifying uncertainties in land surface temperature and emissivity retrievals from ASTER and MODIS thermal infrared data’, **117**, 2012JD018506.

**URL:** <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2012JD018506> 7

NASA JPL (2014), ‘ASTER global emissivity dataset, 100-meter, HDF5’.

**URL:** <https://lpdaac.usgs.gov/products/ag100v003/> 7

Vihola, M. (2012), ‘Robust adaptive metropolis algorithm with coerced acceptance rate’, **22**(5), 997–1008.

**URL:** <https://doi.org/10.1007/s11222-011-9269-5> 3