SVC: Bayesian Models for Spatially Varying Coefficients

Justice Akuoko-Frimpong, Edward Shao, Jonathan Ta

1 Introduction

Spatial heterogeneity in environmental 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. Let $s_i \in \mathrm{R}^2$ for each $i = 1, \ldots, n$ denote spatial locations where we observe response variables $Y(\boldsymbol{s})$ and covariates $\boldsymbol{X}(\boldsymbol{s}) = (X_1(\boldsymbol{s}), \ldots, X_p(\boldsymbol{s}))^\top$. The SVC model specification: We modeled spatial variation in outcomes through location-specific linear combinations of covariates, where each covariate's effect is allowed to vary smoothly across space using Gaussian processes. 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 via strategic knot placement, reducing computational complexity

from cubic to near-linear scaling; (2) an adaptive Metropolis-Hastings sampler that automatically optimizes proposal distributions for spatial parameters during burn-in; and (3) optimized linear algebra operations using pre-computed distance matrices and Cholesky decompositions to accelerate covariance calculations. Together, these advances make spatially varying coefficient models practical for large-scale environmental datasets.

2 Methods

2.1 SVC Model

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

$$Y(\boldsymbol{s}) = \sum_{j=1}^{p} X_r(\boldsymbol{s}) w_r(\boldsymbol{s}) + \epsilon(\boldsymbol{s}),$$

where $w_r(\boldsymbol{s})$ are the regression coefficients corresponding to $X_r(\boldsymbol{s})$ and $\epsilon(\boldsymbol{s})$ are independently and identically distributed multivariate normal (MVN) measurement errors, i.e.

$$\epsilon(\boldsymbol{s}) \sim \mathrm{MVN}(0, \tau^2 I_n).$$

Note that $X_1(\boldsymbol{s})$ can be defined as $\mathbf{1}_n$, a vector of length n with every element equal to 1. In this case, $w_1(\boldsymbol{s})$ would be equivalent to a spatially-correlated random intercept.

For each $w_r(\boldsymbol{s})$, we assign a Gaussian process (GP) prior with squared exponential covariance, i.e.

$$w_r \sim \mathrm{GP}(0, C(\boldsymbol{\theta}_r)), \text{ where}$$

$$C(\boldsymbol{\theta}_r) = [C(s_i, s_{i'}; \boldsymbol{\theta}_r)]_{i,i'=1}^n .$$

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

$$\begin{aligned}
\boldsymbol{\theta}_r &= (\sigma_r^2, \phi_r), \\
C(\boldsymbol{\theta}_r) &= \sigma_r^2 K(\phi_r), \text{ and} \\
K(\phi_r) &= [K(s_i, s_{i'}; \phi_r)]_{i,i'=1}^n = \left[\exp\{\phi_r^{-1} \|s_i - s_{i'}\|^2\}\right]_{i,i'=1}^n ,
\end{aligned}$$

due to its useful properties such as infinite smoothness and gradual decrease in covariance with distance. The parameter σ_r^2 is the spatial variance and ϕ_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 τ^2 and each σ_r^2. In addition, each phi_r is assigned a uniform prior. In summary,

$$\begin{aligned}
\tau^2 &\sim \text{Inv.Gamma}(\alpha_\tau, \beta_\tau), \\
\sigma_r^2 &\sim \text{Inv.Gamma}(\alpha_r, \beta_r), \text{ and} \\
\phi_r &\sim \text{Uniform}(l_r, u_r).
\end{aligned}$$

2.2 MCMC Algorithm

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

2.2.1 Spatial Decay

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).

2.2.2 Spatially Varying Coefficients

Let $\widetilde{Y}(\boldsymbol{s}) = Y(\boldsymbol{s}) - \sum_{j \neq r} X_j(\boldsymbol{s}) w_j^{(t)}(\boldsymbol{s})$. Then

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

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

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

2.2.3 Spatial Variance

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

2.2.4 Nugget

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

2.2.5 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 $r = 1, \ldots, p$. 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 Guassian process as described by Banerjee et al. (2008).

Consider a set of knots $\boldsymbol{s}^* = (s_1^*, \ldots, s_m^*)^\top$ which are a subset of locations $\boldsymbol{s}$. With the low-rank Guassian process, we can still sample τ^2, ϕ_r, and σ_r^2 for each $r = 1, \ldots, p$ because they were assumed to be constant within the spatial domain $\mathcal{D}$. We also sample $\boldsymbol{w}_r^* = [w(\boldsymbol{s}_i^*)]_{i=1}^m$ which follows a multivariate Gaussian distribution with covariance matrix $C^*(\boldsymbol{\theta}_r) = \sigma_r^2 K^*(\phi_r) = [\sigma_r^2 \exp\{\phi_r^{-1} \| s_i^* - s_i^* \|^2\}]_{i,i'=1}^m$.

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

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

where $\boldsymbol{c}(\boldsymbol{s}; \boldsymbol{\theta}_r)$ 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 $w_r(\boldsymbol{s})$ in the original model with $\widetilde{w}_r(\boldsymbol{s})$ for each $r = 1, \ldots, p$, but with an overall cost per iteration of $O(pm^3)$, where $m < n$.

3 Using svc

To fit a linear regression model with spatially varying coefficients 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 ϕ_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 τ^2, σ_r^2, ϕ_r, and $\boldsymbol{w}_r^*$ where $r = 1, \ldots, p$. They can also set the hyperparameters for the inverse gamma priors on τ^2 and σ_r^2. Lastly, users can set the starting proposal standard deviations for each ϕ_r and the target acceptance rate for the RAM.

By default, the starting values for τ^2, σ_r^2, ϕ_r, and $\boldsymbol{w}_r^*$ are set to 1, 1, the midpoint between the corresponding upper and lower bounds, and 0 for all locations, respectively. The default priors for τ^2 and each σ_r^2 are all Inv.Gamma$(0.001, 0.001)$ to achieve uninformative priors. The starting proposal standard deviations for each ϕ_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).

4 Simulation

The purposes of the simulation are to

- assess the performance of SVCLM,
- compare the differences in performance between between full and low rank for SVCLM,
- compare the SVCLM with varycoef and spBayes.

4.1 Metrics

To achieve the purposes of the simulation study, we used the following metrics to evaluate SVCLM:

- Bias,
- RMSE,
- Computational cost (seconds).

4.1.1 Spatial Priors

The spatial random effect w, and regression coefficients β_1 and β_2, are modeled as Gaussian processes with constant mean vectors and spatial covariance matrices. These are defined using exponential covariance functions. Table 1 summarizes the prior distributions used.

Table 1: Priors for spatial random effects and regression coefficients.

Parameter	**Variance (σ^2)**	**Range (ϕ)**	**Distribution**
w	0.1	2	$\mathcal{N}(1_n, \sigma_w^2 \cdot \Phi_w(s, s'))$
β_1	0.1	2	$\mathcal{N}(1_n, \sigma_{\beta_1}^2 \cdot \Phi_{\beta_1}(s, s'))$
β_2	0.1	2	$\mathcal{N}(1_n, \sigma_{\beta_2}^2 \cdot \Phi_{\beta_2}(s, s'))$

Here, $\Phi(s, s')$ denotes a spatial covariance function (e.g., exponential or Matérn), and 1_n is a vector of ones of length n.

4.1.2 Covariate and Response Generation

The covariates and response variable are generated using the distributions and formulae shown in Table 2.

Table 2: Covariate and response generation.

Variable	**Distribution / Formula**
X_1	$\mathcal{N}(0, 1)$
X_2	$\mathcal{N}(5, 1)$
ϵ	$\mathcal{N}(0, \sigma_\epsilon^2)$, with $\sigma_\epsilon^2 = 0.1$
Y	$\beta_1 X_1 + \beta_2 X_2 + w + \epsilon$

4.2 Global Settings and Simulation Replication

To assess the performance of our estimation procedures and reduce sampling variability, the entire data generation and model fitting pipeline is replicated 1000 times. Table 3 summarizes the global settings used throughout the simulations.

Table 3: Global simulation settings.

Setting	**Value / Description**
Spatial grid size	$21 \times 21 = 441$ locations
Number of replications	1000
Error variance σ^2_ϵ	0.1
Covariance function	Exponential kernel

4.3 Comparisons with Other Packages

We compare the performance of our proposed method with `varycoef` and `spBayes` using bias, RMSE, and computational time (in seconds), averaged over 1000 simulation replications. The true values for β_1, β_2, and w are all set to 1.

Table 4: Comparison of methods based on bias, RMSE, and computation time (seconds). Placeholder values shown.

Method	β_1			β_2			w		
	Bias	RMSE	Time	Bias	RMSE	Time	Bias	RMSE	Time
SVCLM (low rank)	-4.36	9.72	60.5	-3.34	3.68	60.5	1.15	2.11	60.5
`varycoef`	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000	0.0
`spBayes`	0.000	0.000	0.0	0.000	0.000	0.0	0.000	0.000	0.0

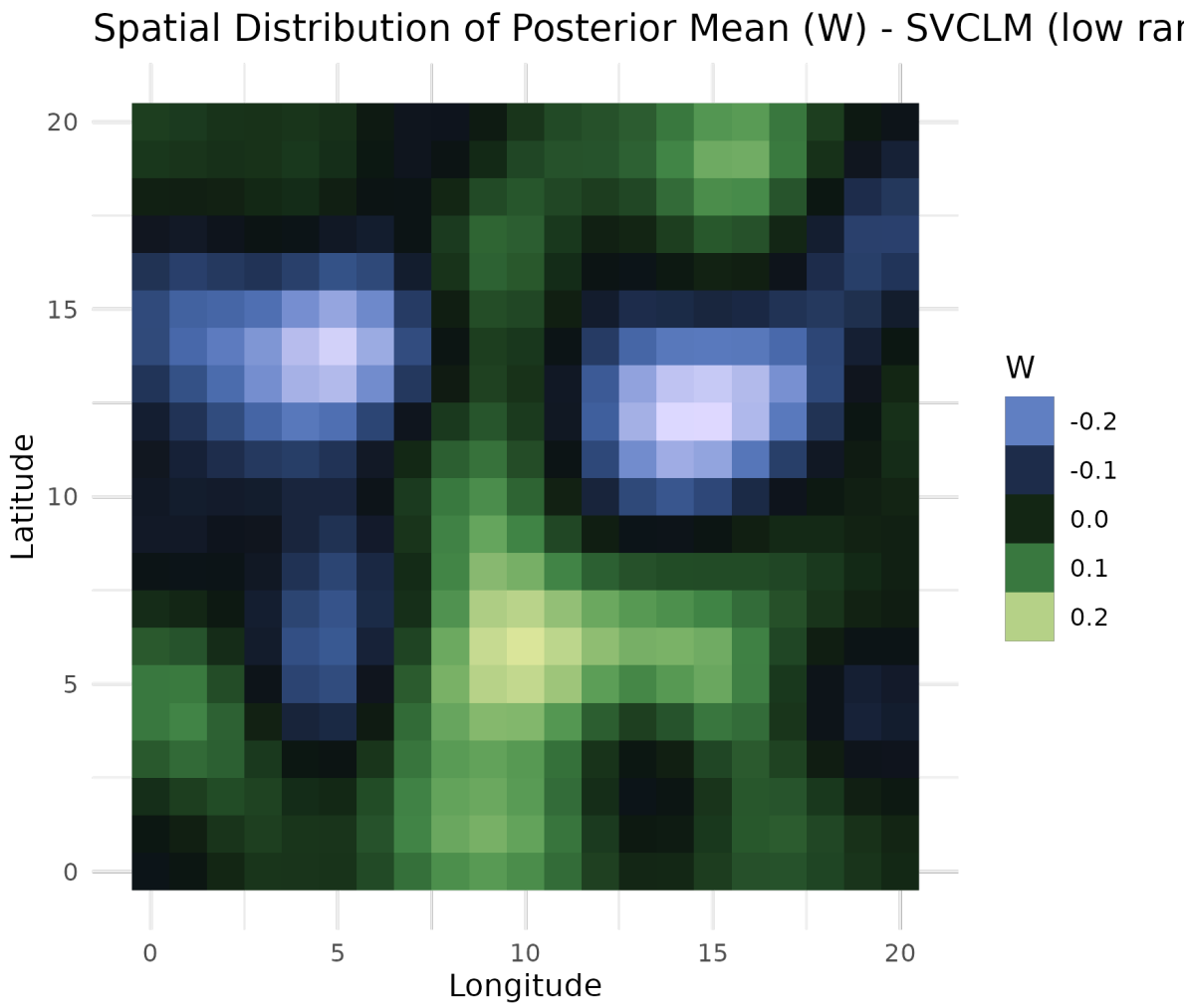

Figure 1: Posterior Mapping for W - SVCLM low rank

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 vegetation health/greenness, and emissivity which represents a surface's efficiency in emitting thermal radiation. We used temperature as a univariate outcome and used emissivity and NDVI as covariates. Figure 4 below reveals the spatial distributions of the variables in the dataset.

Emissivity ranges from 0.88 to 0.96. Light blue zones ($\epsilon > 0.94$) displays dense vegetation/water bodies (high thermal emission efficiency) while dark blue clusters ($\epsilon \approx$

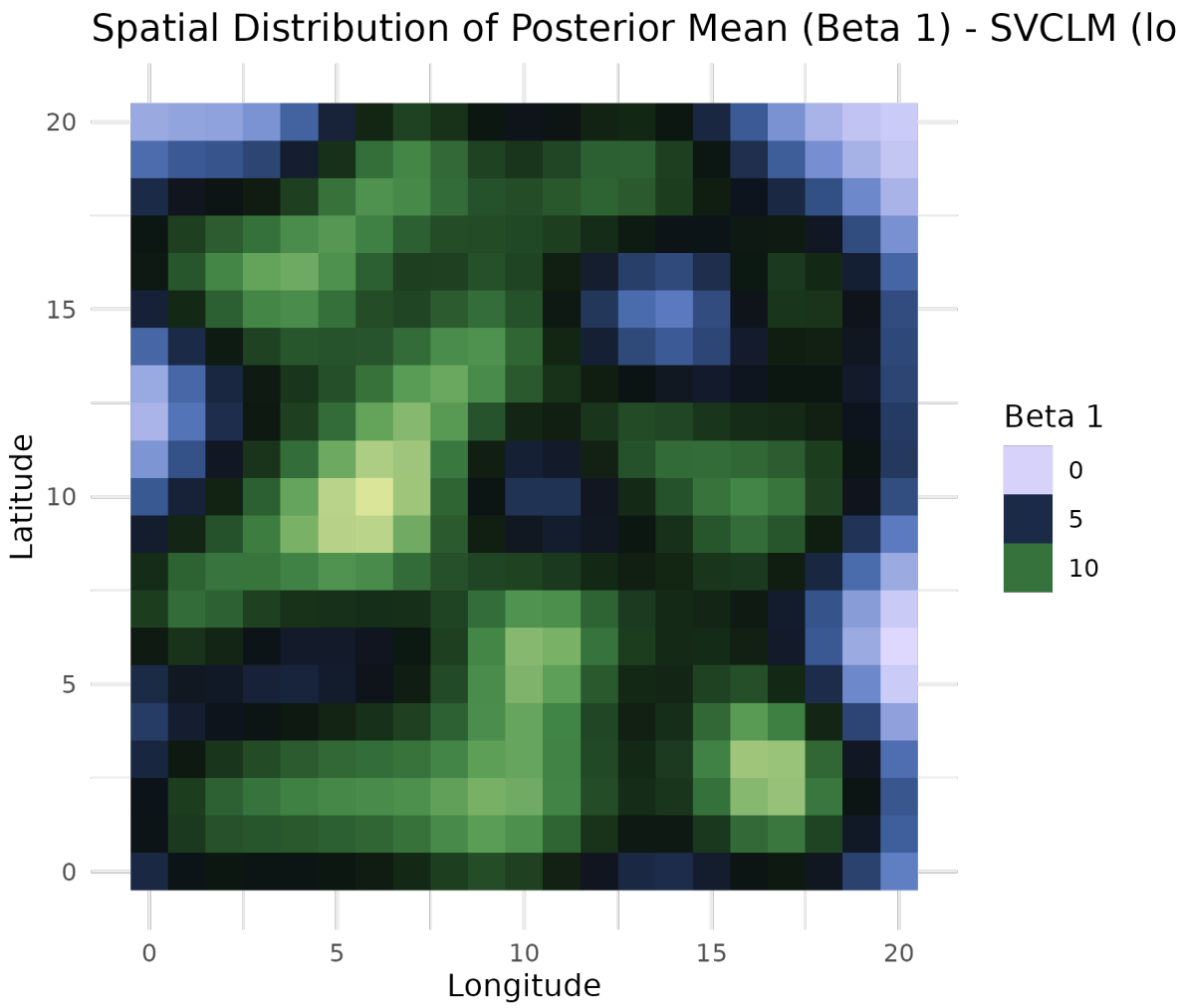

Figure 2: Posterior Mapping for beta 1 - SVCLM low rank

$0.88 - 0.90$) represents urbanized areas. Notice that notice that areas with high NDVI also tend to have higher emissivity which makes sense because vegetation emits efficiently. Cooler regions often align with greener (higher NDVI) areas — vegetation can reduce surface temperature through evapotranspiration.

We computed the empirical semivariograms for the variables in the dataset. The variables showed moderate spatial autocorrelation with low to moderate variance.

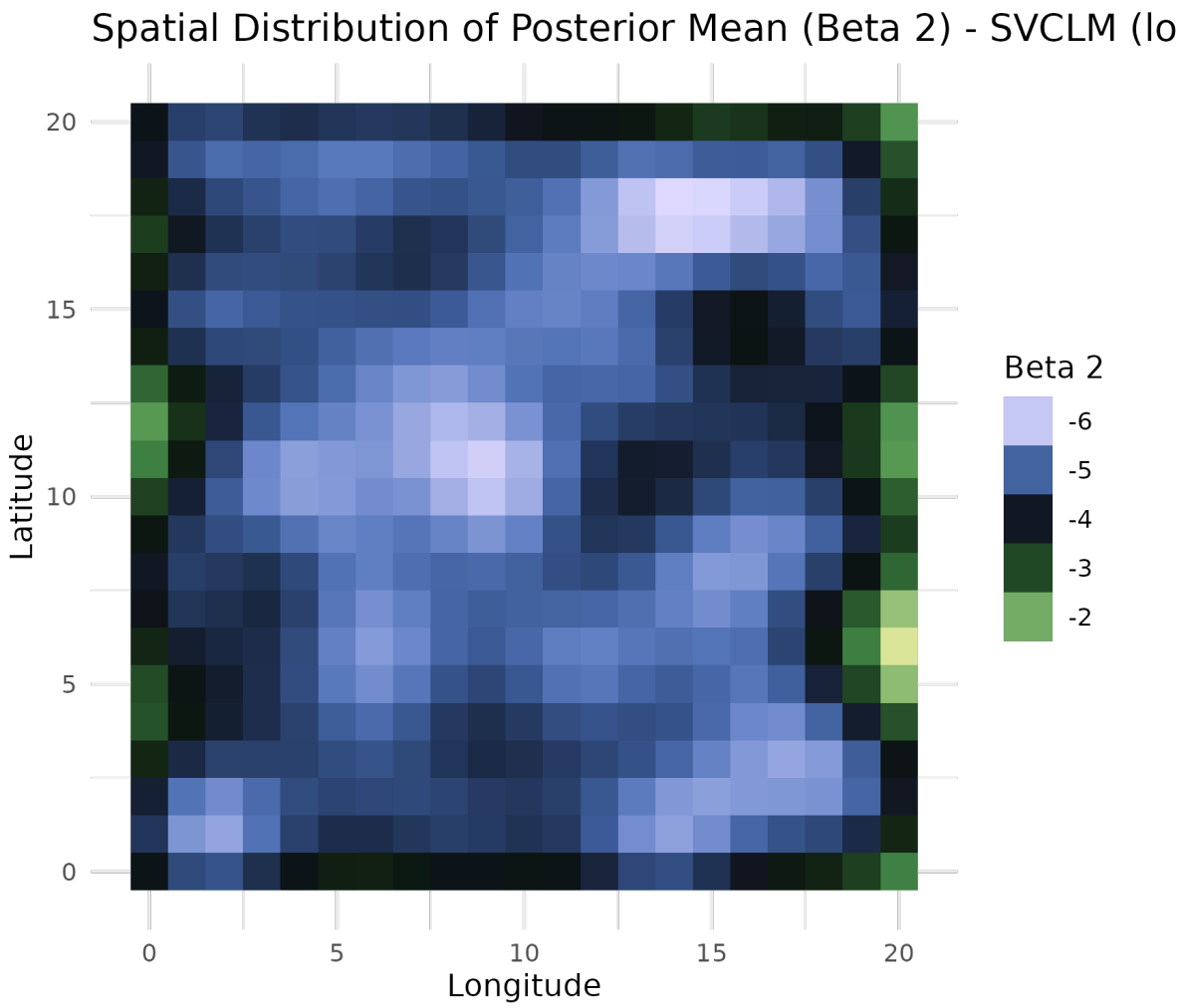

Figure 3: Posterior Mapping for beta 2 - SVCLM low rank

5.1 Model

5.2 Model fitting

Model Specification

We modeled land surface temperature (LST) using a spatially varying coefficient (SVC) framework:

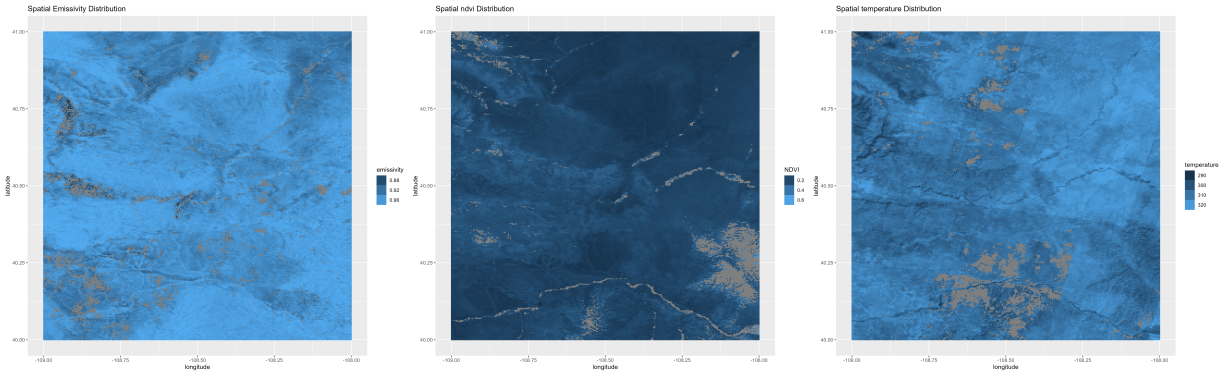

Figure 4: Spatial distributions of emissivity, NDVI, and land surface temperature in the study area.

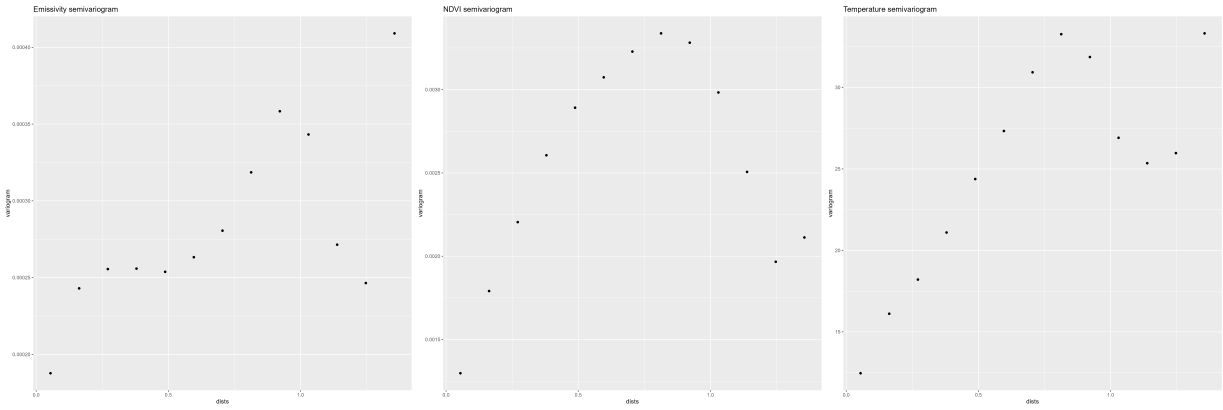

Figure 5: Empirical semivariograms for emissivity, temperature and NDVI

$$\text{Temp}(s) = \beta_0(s) + \beta_{\text{NDVI}}(s) \cdot \text{NDVI} + \beta_{\text{Emis}}(s) \cdot \text{Emissivity} + \epsilon(s)\ \epsilon(s) \quad \sim \text{N}(0, \tau^2)\beta_r(s) \sim \text{GP}(0, \sigma_r^2 \exp(- \tag{1}$$

Prior specifications: We specified weakly informative priors. Spatial variance; $\sigma_r^2 \sim$ Inv-Gamma$(0.001, 0.001)$. Spatial range; $\phi_r \sim$ Uniform$(0.001, 500)$ and the nugget effect; $\tau^2 \sim$ Inv-Gamma$(0.001, 0.001)$

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

5.3 Results

Convergence Diagnostics

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

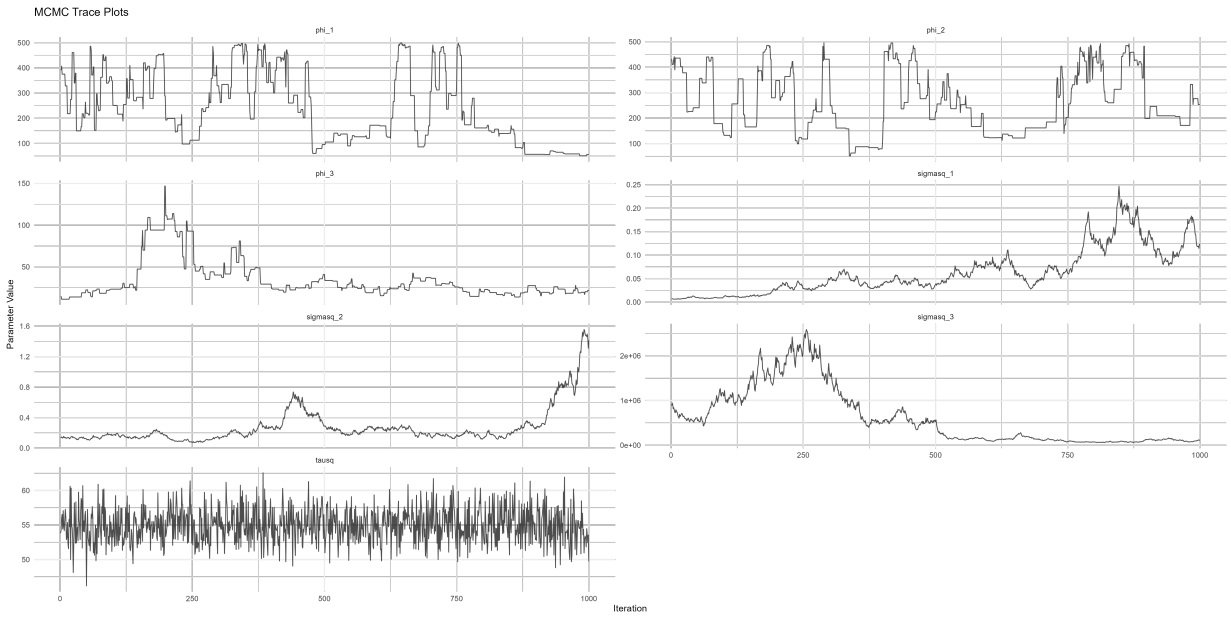

Figure 6: Trace plots for the parameters in the model

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 $0.73°C$ decrease in land surface temperature across space. The emissivity coefficients reveal strong positive associations (mean $\beta_{emis} \approx 330$) which implies that across space, a unit increase in emissivity increase temperature.

Predctions

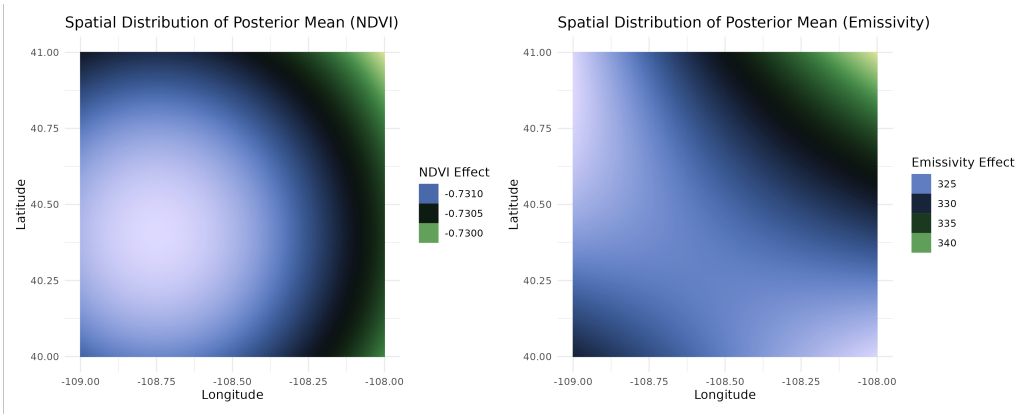

Figure 7: Spatially varying coefficient surfaces for (A) NDVI and (B) Emissivity effects on land surface temperature

We generated predicted temperature values across the entire spatial domain using the posterior mean estimates of the spatially varying coefficients. These predictions were computed for all locations 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

Conclusion

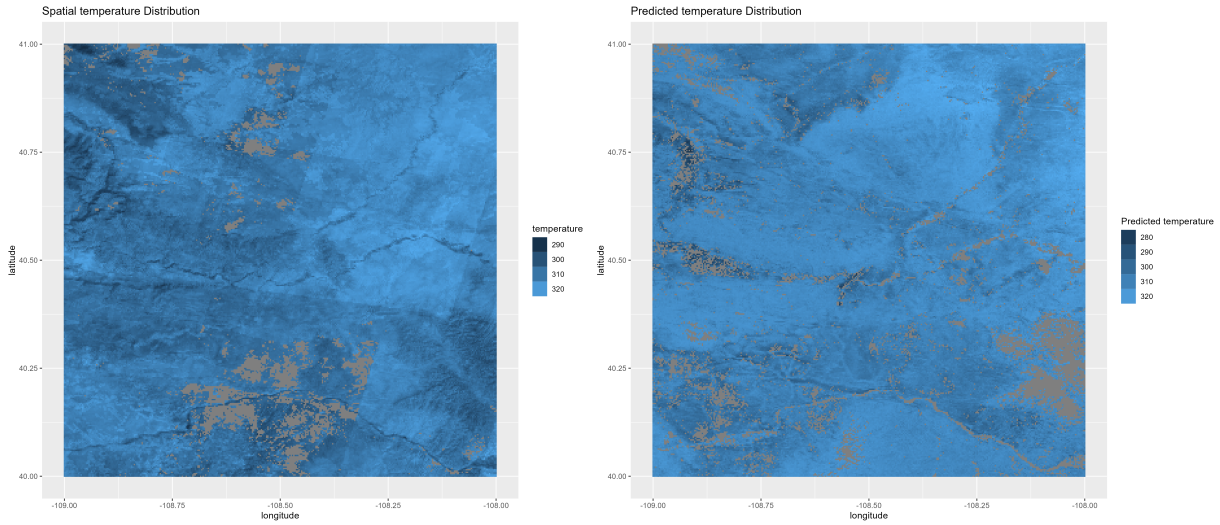

Figure 8: Side by side plot of tempearture and their predicted values across space

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* 5

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* 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* 4