Spatial Lasso with Applications to GIS Model Selection

Hsin-Cheng Huang
Institute of Statistical Science, Academia Sinica

Nan-Jung Hsu
National Tsing-Hua University

David Theobald
Colorado State University

F. Jay Breidt
Colorado State University

September 27, 2006

Abstract

Geographic information systems (GIS) organize spatial data in multiple two-dimensional arrays called layers. In many applications, a response of interest is observed on a set of sites in the landscape, and it is of interest to build a regression model from the GIS layers to predict the response at unsampled sites. Model selection in this context then consists not only of selecting appropriate layers, but also of choosing appropriate neighborhoods within those layers. We formalize this problem and propose the use of Lasso to simultaneously select variables, choose neighborhoods, and estimate parameters. Spatial smoothness in selected coefficients is incorporated through use of a priori spatial covariance structure, and this leads to a modification of the Lasso procedure. The LARS algorithm, which can be used in a fast implementation of Lasso, is also modified to yield a fast implementation of spatial Lasso. The spatial Lasso performs well in numerical examples, including an application to prediction of soil moisture.

Key words: Cross validation, Least angle regression, Spatial regression, Variable selection.

1 Introduction

A raster-based geographic information systems (GIS) organizes spatial data in multiple two-dimensional arrays called layers. A vector-based GIS describes spatial data as features: points, lines, and polygons. We focus on raster-based GIS, noting that it is possible to convert vector to raster and back. Layers in the raster-based GIS may include information on geology, topography, weather and climate, ownership, political administration, land cover, land use, and so on. In many applications, a response of interest is observed on a set of sites in the landscape, and it is of interest to build a regression model from the GIS layers to predict the response at unsampled sites. Some layers
may have no explanatory power for the response of interest. A layer that is a useful predictor may
influence the response at a particular site through its values at the same site or a set of neighboring
sites. This neighborhood of influence may vary from layer to layer. Model selection in this context
then consists not only of selecting appropriate layers, but also of choosing appropriate neighbor-
hoods within those layers. Allowance for choice of neighborhoods leads to an enormous expansion
in the number of models considered. We write this spatial regression with unknown layers and
neighborhoods as a linear model with a large number of unknown regression coefficients. We as-
sume that regression coefficients are spatially homogeneous from neighborhood to neighborhood
across the landscape, after correcting for known sources of spatial inhomogeneity. Even with this
spatial homogeneity assumption, there is still a very large number of regression coefficients, for
which direct least squares estimation would be inefficient. We propose the use of Lasso (Tibshirani
1996) to simultaneously select variables, choose neighborhoods, and estimate parameters. Lasso
can be applied “off the shelf” to this problem, but this standard implementation does not exploit
the possibility that in a given layer, regression coefficients in the same neighborhood may be similar.
We further allow for spatially dependent errors and the possibility of spatial smoothness in selected
coefficients through use of a priori spatial covariance structure. This leads to a modification of the
Lasso procedure that we call spatial Lasso. The model and the implementation of standard Lasso
and spatial Lasso are described in Sections 2.1–2.3. The least angle regression (LARS) algorithm
of Efron, Hastie, Johnstone, and Tibshirani (2004) can be used in a fast implementation of Lasso,
but does not apply directly to a fast implementation of spatial Lasso. Like the standard Lasso,
the spatial Lasso also has a piecewise linear solution path. We briefly describe a closely-related
alternative algorithm that we call equal-length projection and its modification for computing spatial
Lasso in Section 2.4. In Sections 3.1 and 3.2, we illustrate the Lasso methods in two numerical
experiments, and in Section 3.3 we consider an application to prediction of soil moisture. Spatial
Lasso performs well in all of these examples. Section 4 provides a summary and directions for future
research. Finally, the appendix is devoted to the details of the equal-length projection algorithm.
2 Model and Methods

2.1 Spatial Regression with GIS Layers

Suppose that we have available $p$ known GIS layers $\{x_k(s) : s \in D\}; k = 1, \ldots, p$, mapped to a common spatial resolution described by a regular grid $D \subset \mathbb{Z}^2$ of $m_1 \times m_2$ points, where we assume without loss of generality that $0 \in D$. Responses $Y = (Y(s_1), \ldots, Y(s_n))'$ are observed at spatial locations $\{s_1, \ldots, s_n\} \subset D$. Our goal is to build up a regression model for the response $Y$ from the $p$ GIS layers by selecting not only layers, but also neighborhoods within those layers.

For each $k = 1, \ldots, p$, define the neighborhood set $\mathcal{N}_k \subset \mathbb{Z}^2$, where $0 \in \mathcal{N}_k$ if $\mathcal{N}_k$ is non-empty. The proposed spatial regression model is

$$Y(s_i) = \sum_{j=1}^{J} a_j \phi_j(s_i) + \sum_{k=1}^{p} \sum_{u \in \mathcal{N}_k} b_k(s_i, u)x_k(s_i + u) + \varepsilon(s_i); \quad i = 1, \ldots, n, \quad (1)$$

where $\phi_1(\cdot), \ldots, \phi_J(\cdot)$ are known functions of spatial location only, $b_k(s_i, u)$ is the coefficient corresponding to the neighbor $u$ of $s_i$ at the $k$th layer, and $\varepsilon \equiv (\varepsilon(s_1), \ldots, \varepsilon(s_n))'$ are (spatially dependent) error variables with $\text{var}(\varepsilon) = \sigma^2 \Sigma$. Note that $Y(s_i)$ does not depend on layer $k$ if $\mathcal{N}_k$ is empty; depends on layer $k$ only through $x_k(s_i)$ if $\mathcal{N}_k = \{0\}$; and in general depends on layer $k$ through the set of neighboring values $\{x_k(s_i + u) : u \in \mathcal{N}_k\}$.

The coefficients $\{b_k(s_i, u)\}$ depend on layer, neighborhood within layer, and site within neighborhood. To reduce the number of coefficients that we need to estimate, we make the assumption that neighborhoods within layers are spatially homogeneous across the landscape, after accounting for known sources of spatial heterogeneity. Specifically, we assume that

$$b_k(\cdot, u) = \sum_{l=1}^{L_k} c_{k,l}(u)\psi_{k,l}(\cdot); \quad u \in \mathcal{N}_k,$$  \quad (2)

where $c_{k,l}(u)$ are unknown, spatially homogeneous regression coefficients (i.e., no dependence on $s_i$), and $\psi_{k,l}(\cdot)$ are known basis functions (possibly depending on additional GIS layers) that adjust for known sources of spatial heterogeneity. The special case of spatial homogeneity is obtained with $L_k = 1$ and $\psi_{k,1}(\cdot) \equiv 1$, since $b_k(\cdot, u)$ then reduces to a constant function, for each $u$. An interaction effect between $x_k(s + u)$ and $x_{k'}(s)$ can also be introduced in (1) by taking $L_k = 1$ and $\psi_{k,1}(\cdot) = x_{k'}(\cdot)$. A common type of spatial heterogeneity arises when the domain $D$ of interest is composed of $M$ different subregions $D_1, \ldots, D_M$ (e.g., watersheds), in which case $\psi_{k,l}(\cdot)$ can be
specified to account for different spatial features in different subregions as
\[
\psi_{k,l}(s) = \sum_{r=1}^{M} I(s \in D_r) f_{k,l,r}(s); \quad l = 1, \ldots, L_k, \quad k = 1, \ldots, p,
\]
where \(\{f_{k,l,1}(\cdot), \ldots, f_{k,l,M}(\cdot)\}\) are known functions. For example, different flow directions in watersheds can be reflected by choosing \(f_{k,l,r}(s) = f_{k,l}(B_r s)\) (\(s \in \mathbb{R}^2, \ r = 1, \ldots, M\)) for some known function \(f_{k,l}(\cdot)\), where \(B_r\) is a \(2 \times 2\) rotation matrix corresponding to the \(r\)th watershed.

The model given by (1) and (2) can be written as a spatial regression model:
\[
Y \equiv (Y(s_1), \ldots, Y(s_n))' = X\beta + \epsilon, \quad (3)
\]
where \(\beta\) consists of all of the unknown coefficients, \(\{a_j\}\) and \(\{c_{k,l}(u) : u \in N_k\}\). Because the total number of regression parameters \(m = J + \sum_{k=1}^{p} L_k |N_k|\) is typically very large, ordinary least squares (OLS) estimates or generalized least squares (GLS) estimates would be inefficient due to large estimation errors. On the other hand, traditional variable selection methods, such as stepwise search, sequential testing, or AIC (Akaike 1973), may be either inefficient or computationally too intensive. Instead of using OLS or GLS with variable selection, we propose to select variables and estimate the parameters simultaneously for this problem by adapting the least absolute shrinkage and selection operator (Lasso) technique introduced by Tibshirani (1996).

### 2.2 Standard Lasso for Spatial Regression

In this section, we consider the usual linear regression with independent errors (i.e., \(\Sigma = I\)). For simplicity, the intercept effect is assumed to be subtracted out from the response variable and the covariates throughout the paper. The Lasso is a constrained version of OLS that trades off bias for lower variance. Specifically, let \(X^* \equiv XM\) be the columnwise standardized version of \(X\) and let \(\beta^* \equiv M^{-1}\beta\). Then (3) can be rewritten as
\[
Y = X^*\beta^* + \epsilon.
\]
The Lasso estimate \(\hat{\beta}\) of \(\beta\) is obtained by minimizing the residual sum of squares,
\[
(Y - X^*\beta^*)'(Y - X^*\beta^*),
\]
with respect to \( \beta^* \) subject to \( \sum |\beta^*_j| \leq t \), where \( \beta^*_j \) is the \( j \)th element of \( \beta^* \), and \( t \) is a tuning parameter. Equivalently, \( \hat{\beta} \) can be obtained by minimizing

\[
(Y - X^* \beta^*)'(Y - X^* \beta^*) + \lambda \sum_{j=1}^m |\beta^*_j|, \tag{4}
\]

where \( \lambda \) is a tuning parameter depending on \( t \).

As explained by Tibshirani (1996), Lasso is readily understood graphically in two dimensions. Picture the \((\beta^*_1, \beta^*_2)\) plane with elliptical contours of the residual sum of squares \( \text{RSS}(\beta^*_1, \beta^*_2) \) superimposed. The constraint \( \sum |\beta^*_j| \leq t \) is a diamond-shaped region centered on \((0, 0)\), with corners on the \( \beta^*_1 \) and \( \beta^*_2 \) axes. Lasso finds the point within the diamond that is as close as possible to the minimum residual sum of squares. If \( t \) is large enough, then the diamond includes the OLS estimate and Lasso yields OLS. If \( t \) is small enough, then the diamond excludes OLS, and the Lasso estimate is shrunken toward \((0, 0)\). In particular, the Lasso point may fall on a corner of the diamond, corresponding to \( \hat{\beta}^*_1 = 0 \) or \( \hat{\beta}^*_2 = 0 \), so that a sub-model may be selected. Lasso thus computes shrinkage estimators and performs model selection simultaneously.

Note that \( \hat{\beta}^* \) is also the Bayes posterior mode if the prior distributions of \( \{\beta^*_1, \ldots, \beta^*_j\} \) are conditionally independent given \( \sigma^2 \), and have a common double-exponential density:

\[
p(\beta^*_j) = \frac{\lambda}{4\sigma^2} \exp \left( -\frac{\lambda}{2\sigma^2} |\beta^*_j| \right), \tag{5}\]

where \( \lambda > 0 \) is a scaling parameter, and \( \sigma^2 \) is the variance of \( \epsilon(s_i) \) in (1).

Because the spatial regression model in (3) is a linear model, standard Lasso can be applied directly to simultaneously select variables, choose neighborhoods, and estimate parameters. The tuning parameter \( \lambda \) in (4) can be estimated using cross validation (CV) or generalized cross validation as in Tibshirani (1996), or using Mallows’ \( C_p \) as in Efron et al. (2004), where the corresponding degrees of freedom can be estimated using Stein unbiased risk estimation (Zou, Hastie, and Tibshirani 2004) or a data perturbation technique (Ye 1998). For implementation, the Lasso estimate in (4) can be solved very efficiently using the least angle regression (LARS) algorithm (Efron et al. 2004).

Note that standard Lasso assumes no structure among regression coefficients in the same neighborhood for a given layer. In practice, it may be reasonable to suppose that these neighboring regression coefficients are similar, so that the response reacts in a similar way to neighboring co-
variate values. One method for incorporating such spatial smoothness is to suppose that neighboring coefficients are spatially correlated, as we do in the next subsection.

### 2.3 Spatial Lasso

To account for both spatially dependent errors and spatial smoothness among coefficients within the same neighborhood, we consider a generalized residual sum of squares and specify a spatial dependence prior for \( \beta \). For simplicity, the error correlation structure \( \Sigma \) is assumed known; see Section 3.3 for an application with unknown \( \Sigma \). We modify (4) by incorporating a prior for \( \beta \) in which the components of \( \beta^{**} \equiv \Gamma^{-1/2}\beta^* \) are independent and follow a common double-exponential density (5), where \( \Gamma \) is the prior correlation matrix of \( \beta \) accounting for the spatial smoothness in each layer. Then the posterior mode of \( \beta \) can be obtained by minimizing

\[
(Y - X^{**}\beta^{**})'\Sigma^{-1}(Y - X^{**}\beta^{**}) + \lambda \sum_{j=1}^{m} |\beta^{**}_j| \tag{6}
\]

over \( \beta^{**} \in \mathbb{R}^m \), where \( X^{**} \equiv X^{*}\Gamma^{1/2} \) and \( \beta^{**}_j \) is the \( j \)th element of \( \beta^{**} \). We call the resulting estimate of \( \beta \) the spatial Lasso estimate. An example of \( \Gamma \) is

\[
\text{cor}(c_{k,l}(s), c_{k',l'}(s')) = \delta_{k,k'}\delta_{l,l'} \exp(-\|s - s'\|/\gamma), \tag{7}
\]

\[
\text{cor}(a_j, c_{k,l}(u)) = 0, \tag{8}
\]

\[
\text{cor}(a_j, a_{j'}) = \delta_{j,j'}, \tag{9}
\]

for \( j, j' = 1, \ldots, J; k, k' = 1, \ldots, p; l, l' = 1, \ldots, L_k; \) and \( s, s' \in \mathcal{N}_k \), where \( \| \cdot \| \) is the \( L_2 \) norm, \( \delta \) is the Kronecker delta function, and \( \gamma > 0 \) is a spatial dependence parameter.

Both the spatial dependence parameter \( \gamma \) and the tuning parameter \( \lambda \) in (6) can be estimated using cross validation (CV) or generalized cross validation as in Tibshirani (1996), or by other methods as discussed in Section 2.2.

### 2.4 Equal-Length Projection

For implementation, the Lasso estimate in (4) can be solved very efficiently using LARS (Efron et al. 2004) because the column vectors of \( X^* \) are standardized. However, the spatial Lasso estimate in (6) cannot be obtained directly using LARS, because the column vectors of \( X^{**} \) in (6) are not standardized and cannot be standardized without distorting the a priori correlation structure for
the regression coefficients. To overcome this difficulty, we developed a generalized version of LARS for solving spatial Lasso. Unlike LARS, which proceeds along an equiangular (or equi-correlation) path, the proposed algorithm proceeds along an equal-length projection path, as explained in the Appendix. We call the new algorithm equal-length projection (ELP). Just as LARS requires a slight modification to compute the standard Lasso, ELP requires a slight modification to compute the spatial Lasso. The modification of ELP for computing the spatial Lasso is also described in the Appendix.

3 Examples

We examine the performance of the proposed method based on two numerical experiments and one application. The first experiment is designed to mimic variable-selection problems for GIS data, in which covariates are available at multiple spatial layers. The second experiment is designed to further examine the efficiency gain of using spatial Lasso when the regression coefficients in each layer are smoothly varied. Finally, we consider an application of real GIS data to predict an index of soil moisture.

3.1 Numerical Experiment I: Layer and Neighborhood Selection

In our first numerical experiment, we considered six spatial layers, \(\{x_k(s) : s \in D\}; k = 1, \ldots, 6\), defined on a regular lattice \(D \equiv \{(i_1, i_2) : i_1, i_2 = 1, \ldots, 110\}\) of size 110 \(\times\) 110. For each \(k\), \(\{x_k(s) : s \in D\}\) were independently generated from a zero-mean stationary Gaussian spatial process with an exponential covariance function,

\[
C(h) \equiv \text{cov}(x_k(s), x_k(s + h)) = \exp(-\|h\|/\theta); \quad k = 1, \ldots, 6. \tag{10}
\]

We used the “GaussRF” function in the “RandomFields” package of R, available through the Comprehensive R Archive Network (CRAN) website (http://cran.r-project.org), and chose a turning-band method, called TBM3, to generate these processes. We considered two values of \(\theta\) (2 and 0.5), corresponding to strong and weak spatial dependence.

The underlying process was generated from

\[
\mu(s) = 1 + \frac{1}{3} \left\{ x_1(s) + \sum_{j=-1}^{1} x_1(s + (j,1)) + \sum_{j=-2}^{2} x_1(s + (j,2)) \right\} + \frac{1}{3} \sum_{j=-1}^{1} \sum_{k=-1}^{1} x_2(s + (j,k)),
\]
for $s \in D$. The neighborhood for layer 1, $N_1$, is an inverted 5-3-1 pyramid consisting of one site, the three sites above it, and the five sites above the three. The neighborhood for layer 2, $N_2$, is a centered 3×3 block of sites. Neighborhoods $N_3, N_4, N_5, N_6$ are all empty.

The response variables $Y = (Y(s_1), \ldots, Y(s_n))'$ were generated by adding noise to the above mean function:

$$Y(s_i) = \mu(s_i) + \varepsilon(s_i); \quad i = 1, \ldots, n,$$

where $\{s_1, \ldots, s_n\}$ were sampled from $D^* \equiv \{(i_1, i_2) : i_1, i_2 = 6, \ldots, 105\}$ using simple random sampling with sample size $n = 100$, and $\varepsilon(s_1), \ldots, \varepsilon(s_n)$ are independent standard normal random variables. Figure 1 (a) shows a realization of $\{\mu(s) : s \in D^*\}$, where the underlying spatial layers $x_1(\cdot)$ and $x_2(\cdot)$ were generated from (10) with $\theta = 2$.

We applied OLS, standard Lasso, and spatial Lasso to the generated data. Specifically, we considered three OLS estimators ($\hat{\mu}_{OLS,1}, \hat{\mu}_{OLS,3}, \text{and} \hat{\mu}_{OLS,5}$) of the mean $\mu$ based on the following three regression models:

$$Y(s_i) = \beta_0 + \sum_{l=1}^{2} \beta_l x_l(s_i) + \varepsilon(s_i),$$

$$Y(s_i) = \beta_0 + \sum_{l=1}^{2} \sum_{j=-1}^{1} \sum_{k=-1}^{1} \beta_{l,j,k} x_l(s_i + (j,k)) + \varepsilon(s_i),$$

$$Y(s_i) = \beta_0 + \sum_{l=1}^{2} \sum_{j=-2}^{2} \sum_{k=-2}^{2} \beta_{l,j,k} x_l(s_i + (j,k)) + \varepsilon(s_i).$$

Note that in each case, the OLS estimator is based on the correct layers, but only the third OLS estimator has a neighborhood structure large enough to contain the true neighborhoods.

To apply standard Lasso, we set $J = L_1 = \cdots = L_6 \equiv 1$ with $\phi_1(\cdot) = \psi_{1,1}(\cdot) \cdots = \psi_{6,1}(\cdot) \equiv 1$, corresponding to a model with an intercept and spatial homogeneity across neighborhoods. Unlike the OLS models, standard Lasso is not based on the correct layers. In every layer, we considered five different neighborhood sets,

$$N^{(2q+1)} = \{0, \pm 1, \pm 2, \ldots, \pm (2q + 1)\}^2; \quad q = 1, \ldots, 5,$$

resulting in five regression models with $m = 1 + p |N^{(2q+1)}| = 28, 76, 148, 244, 364$ parameters to be selected and estimated. We used multiple neighborhood sets, rather than choosing only one large
set and allowing Lasso to do all of the selection, for two reasons. First, since it is not clear how large a set is large enough, choice of a single set can lead to either bias or excessive estimation error in standard Lasso. Second, spatial Lasso for different neighborhood sizes corresponds to different, non-nested models. In either case, selecting among different neighborhood sets using cross-validation leads to better performance empirically.

Let $\hat{\mu}_{L,2q+1}$ be the Lasso estimate of $\mu$ associated with (4) based on the neighborhood set $\mathcal{N}^{(2q+1)}$ for $q = 1, \ldots, 5$, where the tuning parameter $t$ for each case was selected using ten-fold CV. The final Lasso estimate, denoted as $\hat{\mu}^*_{L}$, is selected among $\{\hat{\mu}_{L,3}, \ldots, \hat{\mu}_{L,11}\}$ having the smallest (ten-fold) CV value.

For one particular realization of the response variables, Figures 1 (b)–(d) show the absolute prediction errors of the three OLS estimates. Figures 1 (e)–(h) show the absolute prediction errors of four standard Lasso estimates, $\{\hat{\mu}_{L,5}, \hat{\mu}_{L,7}, \hat{\mu}_{L,9}, \ldots, \hat{\mu}_{L,11}\}$. For this particular realization, the first OLS model has large systematic errors that match features in the true mean function, due to incorrect specification of the neighborhoods. The second and third OLS estimates have smaller and less systematic errors than the first OLS, but larger absolute errors overall than any of the four Lasso estimates.

For each neighborhood set $\mathcal{N}^{(2q+1)}$, we used the proposed ELP algorithm with modifications to
compute six spatial Lasso estimates (6) corresponding to $\gamma = 0, 1, \ldots, 5$, under $\Sigma = I$. Note that the spatial Lasso estimate with a larger $\gamma$ corresponds to stronger spatial dependence of regression coefficients in a layer, and it reduces to the standard (non-spatial) Lasso estimate if $\gamma = 0$. For each $q \in \{1, \ldots, 5\}$, let $\hat{\mu}_{SL,2q+1}$ be the estimate that has the smallest (ten-fold) CV value among the six spatial Lasso estimates corresponding to $\gamma = 0, 1, \ldots, 5$. Then the final spatial Lasso estimate $\hat{\mu}_{SL}^*$ is defined to be the one among $\{\hat{\mu}_{SL,3}, \ldots, \hat{\mu}_{SL,11}\}$ having the smallest CV value.

To assess the performance of the various estimators across multiple realizations, we used the average squared error (ASE) of prediction across the entire spatial domain,

$$ASE = \frac{1}{10000} \sum_{s \in D^*} (\hat{\mu}(s) - \mu(s))^2,$$

where $\hat{\mu}(s)$ is a generic estimate of $\mu(s)$. The ASE values for various methods based on 100 simulation replicates are summarized in Figure 2. The boxplot labeled “True” uses OLS based on exactly the correct neighborhood structure (usually infeasible in practice), so that model selection is unnecessary. As in the single simulated realization of Figure 1, the worst estimator corresponds to the first OLS model. The best estimators are Lasso and spatial Lasso with large enough neighborhood sizes to encompass the true neighborhoods. These estimators dominate the corresponding OLS estimators with sufficiently large neighborhoods, even though the OLS estimators have the added advantage of correct layer specification. Comparing between the best spatial Lasso and the best standard Lasso, the best spatial Lasso appears to perform better by having a slightly smaller median ASE value.

It is of interest to see how well the Lasso procedures pick up the correct neighborhood structures within the candidate neighborhood sets $N^{(2q+1)}; q = 1, \ldots, 5$. Figure 3 shows the average proportions of $\{x_k(s) : s \in N^{(2q+1)}\}$ being selected for each $k$ ($k = 1, 2, 3$) and $q$ ($q = 1, \ldots, 5$) using the standard Lasso under the spatial process with $\theta = 2$. (That is, the figure records the number of times out of 100 simulated realizations that the regression coefficient was estimated as non-zero.) Results for layers $k = 4, 5, 6$ are similar to those for $k = 3$ and are not shown. Clearly, the Lasso procedures are picking up the correct neighborhood structures: inverted pyramid for layer 1, centered block for layer 2, and empty set for layer 3. The results for $\theta = 0.5$ are similar and are not reported here.
3.2 Numerical Experiment II: Spatial Smoothness

The setting of the second numerical experiment is designed to demonstrate the advantages of spatial Lasso when the regression coefficients vary smoothly within neighborhoods. The domain $D$ is the same as that for the first experiment, but the underlying mean surface was generated from a single covariate layer,

$$
\mu(s) = 1 + 5 \sum_{i_1=-2}^{2} \sum_{i_2=-2}^{2} w_{i_1,i_2} x(s + (i_1, i_2)); \quad s \in D,
$$

where $\{x(s) : s \in D\}$ consists of independent standard normal random variables and the weights are given by a truncated Gaussian kernel,

$$
w_{i_1,i_2} \equiv \frac{\exp\left(-\left(i_1^2 + i_2^2\right)/4\right)}{\sum_{i_1=-2}^{2} \sum_{i_2=-2}^{2} \exp\left(-\left(i_1^2 + i_2^2\right)/4\right)}; \quad i_1, i_2 = 0, \pm 1, \pm 2,
$$

(see Figure 5a). As in the first experiment, the response variables $Y = (Y(s_1), \ldots, Y(s_n))'$ were generated according to (11), where $\{s_1, \ldots, s_n\}$ were sampled from $D^*$ using simple random sampling of size $n = 100$. We considered $p = 1$, $J = L_1 = 1$, $\phi_1(\cdot) = \psi_{1,1}(\cdot) \equiv 1$, five different neighborhood sets $\{N^{(2q+1)} : q = 1, \ldots, 5\}$ of (15), and $\Sigma = I$. For each neighborhood set $N^{(2q+1)}$, six spatial Lasso estimates defined in (6) corresponding to $\gamma = 0, 1, \ldots, 5$ were computed using the proposed ELP algorithm. As in the first simulation experiment, for each $q \in \{1, \ldots, 5\}$, let $\hat{\mu}_{SL,2q+1}$ be the estimate that has the smallest (ten-fold) CV value among the six spatial Lasso estimates.
Figure 3: Average proportions of \( \{ x_k(s) : s \in \mathcal{N}^{2q+1} \} \) being selected for \( \theta = 2 \) under various neighboring structures, where the three columns correspond to \( k = 1, 2, 3 \) and the five rows correspond to \( q = 1, \ldots, 5 \), respectively.
corresponding to $\gamma = 0, 1, \ldots, 5$. Then the final spatial Lasso estimate $\hat{\mu}_{SL}^*$ is defined to be the one among $\{\hat{\mu}_{SL,3}, \ldots, \hat{\mu}_{SL,11}\}$ having the smallest CV value.

The spatial Lasso estimate $\hat{\mu}_{SL}^*$ is compared with the standard Lasso estimate $\hat{\mu}_L^*$, as well as with the three OLS estimates corresponding to models (12)–(14) with $\beta_2 = \beta_{2,j,k} = 0$ for all $j$ and $k$. Note that $\hat{\mu}_{OLS,5}$ is the OLS estimate based on a true model. The ASE results for various estimation procedures based on 100 simulation replicates are summarized in Figure 4.

As in the first numerical experiment, $\hat{\mu}_{OLS,1}$ performs badly due to the incorrect neighborhood specification. The estimator $\hat{\mu}_{OLS,3}$ performs quite well relative to the estimator based on the true model, $\hat{\mu}_{OLS,5}$, because it imposes a hard shrinkage to zero on all of the small coefficients in the outer band of the neighborhood. Standard Lasso has no particular advantages in this numerical experiment. The OLS estimators based on larger neighborhoods have both the correct layer specification and correct or nearly correct neighborhood specification, so the model selection capability of standard Lasso is not useful in this example. Consequentially, standard Lasso performs similarly to the OLS estimators. Further, standard Lasso does not take advantage of spatial smoothness of the regression coefficients. Spatial Lasso does use this information to advantage, and so $\hat{\mu}_{SL}^*$ clearly outperforms all the other estimates by a large margin.

Finally, Figure 5 shows the images of the true regression coefficients and the average estimated coefficients from spatial Lasso under various neighborhood sets. Spatial Lasso is able to pick up the correct neighborhood structure and to capture the smooth structure of the regression coefficients.

![Figure 4: Boxplots of ASE performance for various estimation methods based on 100 simulation replicates for the second experiment.](image-url)
3.3 Application to Soil Moisture Prediction

In this example, the response of interest $Y(\cdot)$ is a soil moisture index, known for all sites on a spatial lattice of $100 \times 100$ regular grid points $D \equiv \{(i_1, i_2) : i_1, i_2 = 1, \ldots, 100\}$. (The soil moisture index is a modeled product from other GIS layers, not a field-measured value at every site.) To assess the prediction ability of the proposed spatial Lasso method, the response variable on a sample of sites was used for model fitting, and the responses on the remaining sites were put aside for testing. Specifically, we used $n = 100$ responses $\{Y(s_1), \ldots, Y(s_n)\}$ for model fitting, where $\{s_1, \ldots, s_{100}\}$ were sampled from $\{(i_1, i_2) : i_1, i_2 = 6, \ldots, 95\}$ using simple random sampling.

From the GIS, we also have five explanatory variables: aspect, hillshade, elevation, slope, and precipitation, denoted by $x_1^*(s), \ldots, x_5^*(s)$. Figure 6 shows the images for the response and the five explanatory variables. We are interested in building a regression model to predict the soil moisture index based on layers constructed from the five explanatory variables. The variable aspect is first transformed by taking the cosine of the angle from the north, so that 1 represents the north and −1 represents the south. Then the five explanatory variables were standardized to
have zero mean and unit variance,

\[ x_k = \frac{x_k^* - \text{mean}(x_k^*)}{\sqrt{\text{var}(x_k^*)}}; \quad k = 1, \ldots, 5. \]

The model (1) was applied with \( J = 1 \) and \( p = 10 \) layers, where the first five layers are \( x_1, \ldots, x_5 \). The sixth to eighth layers, \( x_6 = x_3, x_7 = x_4, \) and \( x_8 = x_1 \), were used to form interaction effects with respect to precipitation \((x_5)\) through \( \psi_{6,1} = \psi_{7,1} = \psi_{8,1} = x_5 \). In addition, \( x_9 = x_3 \) and \( x_{10} = x_4 \) were used to form interaction effects with respect to aspect \((x_1)\) through \( \psi_{9,1} = \psi_{10,1} = x_1 \).

We set \( \mathcal{N}_k = \{0\} \) for layers \( k = 5, 6, 7, 8 \) involving precipitation, because precipitation takes on only two possible values, and allowing larger neighborhoods would lead to considerable redundancy. We chose \( \mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4 \) separately among four different neighborhood sets \( \{\mathcal{N}^{(1)}, \mathcal{N}^{(3)}, \mathcal{N}^{(7)}, \mathcal{N}^{(11)}\} \). In addition, \( \mathcal{N}_9 \) corresponding to the interaction effect between \( x_1 \) and \( x_3 \) was chosen to be the larger set between \( \mathcal{N}_1 \) and \( \mathcal{N}_3 \). Similarly, \( \mathcal{N}_{10} \) corresponding to the interaction effect between \( x_1 \) and \( x_4 \) was chosen to be the larger set between \( \mathcal{N}_1 \) and \( \mathcal{N}_4 \). Note that the number of choices for layer selection in this example is \( 2^{10} \), but the number of model choices for layer and
neighborhood selection is $2^4 \left(2^{121}\right)^6$ when $N_1 = N_2 = N_3 = N_4 = N^{(11)}$. Clearly, allowance for choice of neighborhoods leads to an enormous expansion in the number of models considered.

The spatial Lasso of (6) was applied with the prior covariance structure for the regression coefficients given by (7)–(9), where seven different values $\gamma = 0, 1, \ldots, 6$, ranging from no spatial dependence to strong spatial dependence, were considered. The error vector $\epsilon$ is considered to be spatially dependent with the $(i,j)$th element of $\Sigma$ given by $\exp(-\|s_i - s_j\|/\phi)$, where five different values $\phi = 0, 1, \ldots, 4$, ranging from independent errors to strong spatially dependent errors were used. For each neighborhood set $N_k \in \{N^{(1)}, N^{(3)}, N^{(7)}, N^{(11)}\}$; $k = 1, 2, 3, 4$, each $\gamma \in \{0, 1, \ldots, 6\}$, and each $\phi \in \{0, 1, \ldots, 4\}$, we computed the (ten-fold) CV value. Figure 7 (a) and (b) show the boxplots of the 100 smallest CV values corresponding to $\phi = 0, 1, \ldots, 4$ and $\gamma = 0, 1, \ldots, 6$. We found that the smallest CV value is achieved at $N_1 = N_4 = N^{(1)}$, $N_2 = N_3 = N^{(11)}$, $\phi = 2$, and $\gamma = 4$ among the $4^4 \times 7 \times 5 = 8960$ possible combinations of $(N_1, \ldots, N_4, \gamma, \phi)$. Therefore, the corresponding model was selected as the final model. The whole procedure is done using R, which takes about 49 hours on a Pentium-4 (3.4 GHz) PC.

Figure 8(a) shows the predicted soil moisture index surface based on the final fitted model, where the crosses indicate the 100 locations sampled for model fitting. The corresponding prediction standard errors and the absolute standardized prediction errors are shown in Figures 8 (b) and (c), where the prediction standard errors were also obtained using bootstrap by sampling with replacement from the residuals of the selected model based on 100 replicates. The predicted soil

Figure 7: (a) Boxplots of 100 smallest CV values for various $\phi$ values; (b) Boxplots of 100 smallest CV values for various $\gamma$ values.
The standardized coefficients of some variables: (a) \textit{hill shade}; (b) \textit{elevation}; (c) \textit{aspect-elevation}, where the standard errors of the estimated coefficients were obtained using a bootstrap method.

moisture index can be seen to match the true index well. The standardized coefficients of $x_2$, $x_3$ and $x_9$ are shown in Figure 9, where the standard errors were obtained using bootstrap by sampling with replacement from the residuals of the selected model based on 100 replicates. Some spatial dependence features are apparent as expected. The coefficients for $x_1$, $x_4$, and $x_5$ were estimated as 0. The standardized coefficients of the remaining variables are 1.54 for $x_6$, 1.93 for $x_7$, $-2.27$ for $x_8$, and 3.56 for $x_{10}$.

4 Discussion

In this paper, we have considered the problem of spatial regression using a sparse spatial sample and multiple GIS layers. This is an increasingly important problem as GIS data become more and more prevalent. Model selection in this context is complex because there are typically many layers from which to choose, and appropriate neighborhoods within layers must also be chosen. Our
approach is to formalize this problem by constructing a large and flexible linear model, then using a Lasso method to simultaneously select variables, choose neighborhoods, and estimate parameters. Spatial Lasso is our extension of Lasso that allows for spatially dependent errors and the possibility of smoothness in selected coefficients, incorporated through use of a GLS formulation and a priori spatial covariance structure. Standard Lasso is a special case with an OLS formulation and a priori independence. We have generalized the LARS algorithm to provide a new algorithm, ELP, for fast implementation of spatial Lasso. The spatial Lasso performs well in the numerical examples we have considered, including an application to prediction of soil moisture using real GIS data.

The spatial Lasso has an advantage of performing variable selection and estimation simultaneously. Nevertheless, like the standard Lasso, a two-step procedure can also be considered by first applying the spatial Lasso to select the model but not to estimate the coefficients, and then refitting the model using GLS. The resulting GLS estimate has a smaller bias and a smaller generalized residual sum of squares at the cost of having a larger variance.

Although we focus on the Lasso method in this paper, other variable selection methods based on different penalized least square formulations, such as SCAD (Fan and Li 2001), the nonnegative garrote (Breiman 1995), and the elastic net (Zou and Hastie 2005), can also be considered. The proposed method can be easily extended under these settings.

Appendix: Equal-Length-Projection Algorithm

Consider the linear regression model of (3), where $X$ is an $n \times m$ matrix consisting of $m$ covariates. Let $x_j = (x_{j1}, \ldots, x_{jn})'$ be the $j$th column vector of $X$, for $j = 1, \ldots, m$. Also assume that the response and the covariates have been centered such that $1'Y = 0$ and $1'x_1 = \cdots = 1'x_m = 0$.

Like the LARS algorithm, the ELP algorithm starts with $\hat{\mu}^{(0)} = 0$ and successively obtains estimate $\hat{\mu}^{(j)} \equiv X\hat{\beta}^{(j)} = \hat{\mu}^{(j-1)} + \hat{\gamma}_j v_j$ by going along an “equal-length projection” path $v_j$, for each $j = 1, \ldots, m$, before reaching the ordinary least squares fit $\hat{\mu}^{(m)}$ based on the full model. At each step, one additional covariate is added to the model and hence at most $j$ components of $\hat{\beta}^{(j)}$ are nonzero at the $j$th step. At the first step, the model consists of only one covariate $x_{k_1}$ such that $|x_{k_1}'Y| \geq \max_{j=1}^m |x_j'Y|$. Let $v_1$ be the projection vector of $Y$ into $x_{k_1}$. Then the algorithm
goes along $v_1$ from 0 to $\tilde{\mu}^{(1)} = \tilde{\gamma}_1 v_1$, where $\tilde{\gamma}_1 \geq 0$ is the smallest value satisfying

$$\max_{j \neq k_1} |x_j' (Y - \tilde{\gamma}_1 v_1)| = |x_{k_1}' (Y - \tilde{\gamma}_1 v_1)|. \tag{16}$$

Let $r^{(1)} \equiv Y - \tilde{\mu}^{(1)}$ be the residual of $\tilde{\mu}^{(1)}$ at the first step. Then, in the second step, the covariate $x_{k_2}$ enters the model with $|x_{k_1}' r^{(1)}| = |x_{k_2}' r^{(1)}| = \max_{j \neq k_1} |x_j' r^{(1)}|$, i.e., $k_2 = \arg \max_{j \neq k_1} |x_j' r^{(1)}|$ being the index other than $k_1$ maximizing (16). Let $v_2$ be the projection vector of $r^{(1)}$ in the column space of $x_{k_1}$ and $x_{k_2}$. It turns out that $v_2$ has an equal-length projection to both $x_{k_1}$ and $x_{k_2}$ (i.e., $|x_{k_1}' v_2| = |x_{k_2}' v_2|$), and the algorithm goes along the “equal-length projection” direction $v_2$ from $\tilde{\mu}^{(1)}$ to $\tilde{\mu}^{(2)} = \tilde{\mu}^{(1)} + \tilde{\gamma}_2 v_2$, where $\tilde{\gamma}_2 \geq 0$ is the smallest value satisfying

$$\max_{j \neq \{k_1, k_2\}} |x_j' (r^{(1)} - \tilde{\gamma}_2 v_2)| = |x_{k_1}' (r^{(1)} - \tilde{\gamma}_2 v_2)| = |x_{k_2}' (r^{(1)} - \tilde{\gamma}_2 v_2)|.$$

Consequently, we have $|x_{k_1}' r^{(2)}| = |x_{k_2}' r^{(2)}| = \max_{j \neq k_1, k_2} |x_j' r^{(2)}|$. Suppose the model consists of covariates $\{x_j : j \in A_l\}$ at the $l$th step where $A_l = \arg \max_{j = 1}^m |x_j' r^{(l-1)}| = \{k_1, \ldots, k_l\}$ indicates the active set. Then, the algorithm at the $(l + 1)$th step $(l = 1, \ldots, m - 2)$ contains the following steps:

- Select the covariate $x_{k_{l+1}}$ with $k_{l+1} = \arg \max_{j \in A_l^c} |x_j' r^{(l)}|$. Consequently,

  $$|x_{k_{l+1}}' r^{(l)}| = |x_j' r^{(l)}|, \quad \text{for all } j \in A_l.$$

- Update the active set with $A_{l+1} = A_l \cup \{k_{l+1}\}$.

- Define $v_{l+1}$ to be the projection vector of $r^{(l)}$ in the column space of $\{x_j, j \in A_{l+1}\}$. Because $|x_j' v_{l+1}|$ is a constant for $j \in A_{l+1}$, $v_{l+1}$ is an “equal-length projection” direction.

- Define $\tilde{\gamma}_{l+1} \geq 0$ such that it is the smallest value satisfying

  $$\max_{k \in A_{l+1}} |x_k' (r^{(l)} - \tilde{\gamma}_{l+1} v_{l+1})| = |x_j' (r^{(l)} - \tilde{\gamma}_{l+1} v_{l+1})|, \quad \text{for all } j \in A_{l+1}.$$

- Define $\tilde{\mu}^{(l+1)} = \tilde{\mu}^{(l)} + \tilde{\gamma}_{l+1} v_{l+1}$ and $r^{(l+1)} \equiv Y - \tilde{\mu}^{(l+1)}$. The algorithm goes along the “equal-length projection” direction $v_{l+1}$ from $\tilde{\mu}^{(l)}$ to $\tilde{\mu}^{(l+1)}$.

Finally, at the $m$th (i.e., the final) step, we have $|x_1' r^{(m-1)}| = \cdots = |x_m' r^{(m-1)}|$, and the algorithm goes along the path with the equal-length projection to all the covariates before reaching the
ordinary least-squares estimate $\hat{\mu}^{(m)}$ based on the full model. Note that the ELP algorithm reduces to the LARS algorithm if we further assume that $x'_1x_1 = \cdots = x'_mx_m = 1$ in (3).

The LARS algorithm with a slight modification can be used to obtain the full set of the Lasso solutions corresponding to the standardized covariates (Efron et al. 2004). The same modification can be used for the ELP algorithm to obtain the full set of the spatial Lasso solutions. Recall that the spatial Lasso estimate is the minimizer of (6). Without loss of generalization, it suffices to consider (3), because to obtain the spatial Lasso solutions is equivalent to obtain the Lasso solutions for the linear regression model (3) with $Y$, $X$, and $\beta$ being replaced by $\Sigma^{-1/2}Y$, $\Sigma^{-1/2}X^{**}$, and $\beta^{**}$. First, as noted by Efron et al. (2004), the sign of a Lasso estimate $\hat{\beta}_j$ of (3) has to be equal to the sign of $x'_j(Y - X\hat{\beta})$:

$$\text{sign}(\hat{\beta}_j) = \text{sign}(x'_j(Y - X\hat{\beta})); \quad j = 1, \ldots, m,$$

which may not be satisfied by a ELP estimate, where $\text{sign}(x) \equiv 1$ if $x > 0$, $\text{sign}(x) \equiv -1$ if $x < 0$, and $\text{sign}(0) \equiv 0$. Similar to the LARS algorithm that gives all Lasso solutions under $x'_1x_1 = \cdots = x'_mx_m = 1$ if the LARS algorithm is restricted to satisfy (17), the full set of Lasso solutions of (3) can be obtained if the ELP solutions are restricted to satisfy (17).

Let $\hat{\mu}_A = X\hat{\beta}$ be an estimate obtained from the ELP algorithm at some step, where $A$ is the corresponding active set. Suppose that $\hat{\mu}_A$ is also a Lasso estimate under model (3). Denote the active set for the next step to be $A^+$. Then the ELP estimate for the next step can be expressed as

$$\hat{\mu}_{A^+} = \hat{\mu}_A + \hat{\gamma}v. \quad (18)$$

Define

$$\hat{\mu}(\gamma) \equiv X\hat{\beta}(\gamma) \equiv \hat{\mu}_A + \gamma v; \quad \gamma \in \mathbb{R},$$

and the $j$th component of $\hat{\beta}(\gamma) = (\hat{\beta}_1(\gamma), \ldots, \hat{\beta}_m(\gamma))'$ can be written as

$$\hat{\beta}_j(\gamma) = \hat{\beta}_j + \gamma b_j; \quad j \in A,$$

for some $b_j$, where $\hat{\beta}_j$ is the $j$th component of $\hat{\beta}$. It is clear that the sign of $\hat{\beta}_j(\gamma)$ switches at $\gamma = -\hat{\beta}_j/b_j$, whereas the sign of $x'_j(Y - X\hat{\beta}(\gamma))$ is unchanged for $0 \leq \gamma \leq \hat{\gamma}; \quad j \in A$ (Efron et
al. 2004). It follows that (17) holds if and only if \( \tilde{\gamma} \equiv \min\{-\hat{\beta}_j/b_j > 0 : j \in \mathcal{A}\} \geq \hat{\gamma} \), where the minimum of the empty set is defined to be \( \infty \).

The following theorem shows that the ELP algorithm can be modified to provide all (spatial) Lasso solutions, proof of which is essentially the same as that of Efron et al. (2004), and hence is omitted.

**Theorem 1** Under the above conditions, if \( \hat{\gamma} \) in (18) of the ELP algorithm is replaced by \( \tilde{\gamma} \) with \( \mathcal{A}^+ = \mathcal{A} \setminus \{\tilde{j}\} \) whenever \( \tilde{\gamma} < \hat{\gamma} \), and \( \tilde{j} \) is the only element satisfying \( -\hat{\beta}_{\tilde{j}}/b_{\tilde{j}} = \tilde{\gamma} \). Then the modified ELP algorithm solves all (spatial) Lasso solutions.

**References**


