Bayesian Spatial Classification

Candace Berrett*1 and Catherine A. Calder†2

1Department of Statistics, Brigham Young University, Provo, UT, USA
2Department of Statistics, The Ohio State University, Columbus, OH, USA

June 17, 2014

Abstract

In analyses of spatially-referenced data, researchers often have one of two goals: to quantify relationships between a response variable and covariates while accounting for residual spatial dependence, or to predict the value of a response variable at unobserved locations. In this second case, when the response variable is categorical, prediction can be viewed as a classification problem. However, many existing classification methods either ignore response variable and covariate relationships and rely only on spatially proximate observations for classification, or they ignore spatial dependence and use only the covariates for classification. The Bayesian spatial generalized linear (mixed) model offers a tool to accommodate both sources of information in classification problems. We take a close look at different versions of this model that have been proposed in the literature and, within the canonical classification framework, build on these models to construct spatial classifiers. We demonstrate the utility of our proposed classification methodology through a comparison to other methods using an analysis of satellite-derived land cover data from Southeast Asia.

Keywords: generalized linear model, land cover, mixed models, MCMC, probit regression

1 Introduction

Prediction of unobserved binary or categorical variables can be cast as a classification problem, where a classification rule is used to assign an unobserved variable to a class, or category, based on a collection of observed inputs (e.g., predictors or covariate information). A classification rule is determined by a decision function, a function of the inputs, which can be...
derived from either an underlying statistical model (e.g., logistic regression and discriminant analysis) or an algorithmic method such as support vector machines (SVM) and k-nearest neighbors (kNN) (see Hastie et al., 2001 for an overview). In this paper, we consider the spatial classification problem. That is, we seek to define classification rules to assign an unobserved variable associated with a spatial location (a particular point in a continuously-indexed spatial domain or an area in a discretely-indexed spatial domain) to one or more discrete classes. We refer to this spatial location as the focal location and the area surrounding it as the neighborhood of the focal location.

In classification problems involving spatially-referenced variables, we argue that neighboring values of the unobserved/unknown variable should be used as inputs to the decision function, along with other inputs associated with the focal location and its neighbors. For example, binary or categorical images derived from satellite remote sensing often contain unobserved pixels due to errors in processing the raw data or measurement complications such as cloud cover. In these situations, formal classification methods are needed to assign values to the unobserved pixels so that the images can be used for various purposes in scientific investigations. While values of inputs associated with the focal pixel (e.g., land cover) may contain valuable information, knowledge of the pixel values in the area surrounding the focal pixel may also be useful in classifying the focal pixel correctly. As we will illustrate, classification rules that rely on neighboring observations can be derived from the Bayesian spatial generalized linear and generalized linear mixed models (SGLMs and SGLMMs, respectively).

In this paper, we first provide a close look at different Bayesian SGLMs and SGLMMs that have been used in the literature in Section 2. In the following section, we define decision functions based on these models for performing spatial classification. Using land cover observations over a region in Southeast Asia, we compare these classification methods to other non-Bayesian model-based and algorithmic methods in Section 4. Finally, we conclude with a discussion of our findings and their implications.

2 Spatial Generalized Linear (Mixed) Models

Following the seminal work of Diggle et al. (1998), the generalized linear mixed model (GLMM) has become the go-to framework for modeling spatially-dependent phenomena for which Gaussian distributional assumptions are inappropriate. Through the introduction of spatially-dependent random effects within a generalized linear model (McCullagh and Nelder, 1989), standard models for both continuous and discretely-indexed Gaussian data (e.g., Gaussian processes, spatial autoregressive models) can be readily adapted to the non-Gaussian data situation. (See Paciorek, 2007 for several examples, and Hughes and Haran, 2013 for a more recent approach to dimension reduction and alleviation of confounding in these models.) We review the SGLMM for dichotomous spatial data below in order to make
connections between specific models that have appeared in the literature and to provide a framework for defining spatial classification methods.

Let \( \mathbf{Y} = (Y_1, \ldots, Y_n)' \) be binary response variables associated with locations \( \mathbf{s} = (s_1, \ldots, s_n)' \), which are a subset of either a continuous or discrete spatial domain. The standard generalized linear model (GLM) is specified through three components: the random component, the link function, and the systematic component. The SGLMM can be written using these components as well. For the random component of the model, we assume that conditional on unknown parameters \( \mathbf{\beta} \) and \( \mathbf{\nu} \), the \( Y_i \)'s are independent and that

\[
Y_i | \mathbf{\beta}, \mathbf{\nu} \sim \text{Bin}(1, p_i),
\]

where \( p_i = P(Y_i = 1) \). For the link function, we let

\[
g(p_i) \equiv \eta_i,
\]

where generally we assume \( g(\cdot) \) is a one-to-one, monotone, continuous, and differentiable function. For binary response variables, common link functions are the logit and probit functions, \( g(p_i) = \log(p_i/(1-p_i)) \) and \( g(p_i) = \Phi^{-1}(p_i) \), respectively, where \( \Phi() \) is the standard normal cumulative distribution function and \( \Phi^{-1}() \) denotes its inverse. Finally, the systematic component of the model is

\[
\eta_i = \mathbf{x}_i' \mathbf{\beta} + \nu_i,
\]

where \( \mathbf{x}_i \) is a \( k \times 1 \) vector of covariates measured at location \( s_i \), \( \mathbf{\beta} \) is a \( k \times 1 \) vector of regression coefficients, and \( \mathbf{\nu} = (\nu_1, \ldots, \nu_n)' \) is a spatial random effects vector with components associated with each spatial location.

By definition, the inclusion of random effects in the systematic component of the model makes the model a GLMM. It also allows for the introduction of spatial dependence through the dependence structure of \( \mathbf{\nu} \). In particular, we take

\[
\mathbf{\nu}|\mathbf{\theta} \sim \mathcal{N}(\mathbf{0}, \Sigma(\mathbf{\theta}))
\]

where \( \mathbf{0} \) is an \( n \times 1 \) vector of zeros and \( \Sigma(\mathbf{\theta}) \) is the \( n \times n \) spatial covariance matrix parameterized by \( \mathbf{\theta} \). (For example, the spatial covariance matrix could correspond to a Matérn covariance function or a spatial autoregressive model.) As Banerjee et al. (2004) point out, \( \Sigma(\mathbf{\theta}) \) would not have a nugget, so that we can write

\[
\Sigma(\mathbf{\theta}) = \theta_1 K(\theta_2),
\]

where \( \mathbf{\theta} = (\theta_1, \theta_2)' \) and \( K(\theta_2) \) is an \( n \times n \) spatial dependence matrix parameterized by \( \theta_2 \). Although not always the case, \( \theta_1 \) is often taken to be the common variance parameter among the \( \nu_i \) and \( K() \) is a spatial correlation function.
The probit link function is often used in GLMs/GLMMs in the Bayesian setting since the well-known latent-variable representation of the probit GLM facilitates model fitting via the Gibbs sampler (Albert and Chib, 1993). This same latent variable representation also suggests an alternative GLM for dichotomous spatial data (compared to the spatial GLMM described above) that has been used in the literature, so we briefly review it here. Introducing a collection of latent variables \( Z = (Z_1, \ldots, Z_n)' \) associated with each spatial location, we take

\[
Y_i = \begin{cases} 
1, & Z_i \geq 0, \\
0, & Z_i < 0
\end{cases}
\]

where

\[
Z_i = x_i'\beta + \nu_i + \epsilon_i,
\]

\( \epsilon_i \overset{iid}{\sim} N(0, 1) \), and, as above, \( \nu | \theta \sim N(0, \Sigma(\theta)) \). We can see that this is simply an alternative representation of the probit SGLMM described previously since

\[
p_i = P(Y_i = 1) = P(Z_i > 0) = \Phi(x_i'\beta + \nu_i).
\]

Notice, however, that

\[
(\nu + \epsilon) | \theta \sim N(0, \Sigma^*(\theta)),
\]

where \( \Sigma^*(\theta) = I + \Sigma(\theta) \) and \( I \) is the \( n \)-dimensional identity matrix. If \( \Sigma^*(\theta) \) is equal to \( \Sigma(\theta) \) instead (i.e., dropping the identity matrix), we have a spatial GLM (SGLM) as opposed to an SGLMM. That is, there are no random effects in the systematic component of the GLM since the \( \nu \)s now take the place of the random component of the GLM in the latent variable representation of the model. For this SGLM, we write

\[
Z_i = x_i'\beta + \nu_i.
\]

Both the probit SGLMM and probit SGLM have been used in the literature. Following De Oliveira (2000), Higgs and Hoeting (2010) and Berrett and Calder (2012) considered the SGLM, which is sometimes referred to as a clipped Gaussian process. More recent work by Schliep and Hoeting (2013) makes use of the SGLMM for the ordered-category response variable case. Given that both models are used, we briefly discuss the implications of their seemingly slight differences. Comparing the probit SGLMM and SGLM described above, we note that while the \( Y \)s are conditionally independent under the SGLMM since the dependence is introduced through the dependence structure of the latent \( \nu \)s \( \{\nu_i\} \), the \( Y \)s are not conditionally independent in the SGLM. To see this, consider the conditional likelihood function, \( L(\beta, \nu) = P(Y = y | \beta, \nu) \), for the SGLMM, which can be factored into \( n \) components:

\[
L(\beta, \nu) = \prod_{i=1}^{n} (\Phi(x_i'\beta + \nu_i) I(y_i = 1) + (1 - \Phi(x_i'\beta + \nu_i)) I(y_i = 0))
\]

\[
= \prod_{i=1}^{n} \int_{A_i} \phi(z_i; x_i'\beta + \nu_i, 1) dz_i,
\]
where \( \phi(\cdot; m, c^2) \) is the density function of the univariate normal distribution with mean \( m \) and variance \( c^2 \) and

\[
A_i = \begin{cases} 
(-\infty, 0), & \text{if } y_i = 0, \\
[0, \infty), & \text{if } y_i = 1.
\end{cases}
\]

For the spatial GLM where there are no random effects in the systematic component of the model, the likelihood, \( L(\beta, \theta) = P(Y = y | \beta, \theta) \), is the integral over a multivariate normal distribution:

\[
P(Y = y) = \int_{A} \phi_n(Z; X\beta, \Sigma(\theta)) dZ,
\]

where \( \phi_n(\cdot, m, C) \) is the density function of the \( n \)-dimensional multivariate normal distribution with mean \( m \) and covariance \( C \),

\[
X = \begin{bmatrix} 
x'_{1} \\
\vdots \\
x'_{n}
\end{bmatrix},
\]

and \( A = A_1 \times \cdots \times A_n \).

Since the primary focus of this paper is the spatial classification model, we defer discussion of model fitting to Appendix A. While the probit SGLMM and SGLM described above are nearly identical – they differ only in terms of the inclusion of a identity matrix component in the model for \( \Sigma^*(\theta) \) – it is not immediately clear how the models compare in terms of their predictive performance. The difference between the models is reminiscent of a nugget effect, which is frequently used in geostatistical models and its inclusion is known to influence both point predictions and corresponding uncertainty statements. Before conducting a comparison of the two models, we first describe how these models can be used to define spatial classifiers.

3 Model-Based Classification

3.1 The Classification Problem

In the canonical classification problem, we have a collection of paired observations \( \{(y_i, x_i); i = 1, \ldots, n\} \), where \( y_i \) is the observed value of a binary response variable \( Y_i \) and \( x_i \) is a \( k \times 1 \) vector of covariates used as inputs in determining decision function boundaries for the classes. The binary response variable is an indicator identifying the class to which the set of observed inputs belong. Although many classification methods can be generalized to the multi-category/class setting, we restrict our description of these methods to the binary setting in which there are two classes, \( C_0 \) and \( C_1 \), where \( C_j \) represents the class of observations where
\[ Y = j. \] Of the observations \( \{ (Y_i, x_i); i = 1, \ldots, n \} \), \( n_0 \) fall into class \( C_0 \) and \( n_1 \) fall into class \( C_1 \), and \( n_0 + n_1 = n \).

A classification method defines a decision function \( \delta(\omega) \) where \( \omega \) is the set of applicable inputs, model parameters, and in the spatial setting, surrounding observations. Based on this decision function, we can define a classification rule on an unobserved \( Y \) that we denote generically by \( Y^0 \) with corresponding covariate information \( x^0 \). As described in [Hastie et al. (2001)](https://doi.org/10.1007/978-0-387-77375-3), the Bayes classifier for a 0-1 loss function is

\[
y^0_{\text{pred}} = \arg \max_k P(C_k|\omega),
\]

where \( P(C_k|\omega) = p_k(\omega) \) is the conditional probability of class \( C_k \) given the inputs \( \omega \). In other words, the optimal classification of \( Y^0 \) is the most likely class given \( \omega \). In the two-category case, we can rewrite (9) as

\[
y^0_{\text{pred}} = \begin{cases} 
1, & \text{if } \delta(\omega) > 1 \\
0, & \text{otherwise}
\end{cases}
\]

(10)

for \( \delta(\omega) = p_1(\omega)/p_0(\omega) \).

For model-based classification, the conditional probabilities (and, hence decision functions) depend on unknown parameters denoted by \( \xi \), which must be estimated from the observed data. We write \( \omega_\xi \) to explicitly capture the inclusion of the parameters in the collection of inputs. In a frequentist setting, the unknown parameters, \( \xi \), are often set equal to the maximum likelihood estimates (MLE), so that

\[
\hat{p}_k(\omega) \equiv p_k(\omega_\xi),
\]

where \( \hat{\xi} \) is the MLE of \( \xi \). We refer to the classification rule given by (10) with \( \delta(\omega) \) set equal to \( \hat{\delta}(\omega) = \hat{p}_1(\omega)/\hat{p}_0(\omega) \) as the maximum likelihood classifier. Alternatively, there are two Bayesian estimation approaches we consider. First, we can set \( \xi \) equal to its posterior mean,

\[
\tilde{p}_k(\omega) \equiv p_k(\omega_\tilde{\xi}),
\]

where \( \tilde{\xi} = E[\xi|y] \). In this case, the classification rule is given by (10) with \( \delta(\omega) \) replaced by \( \tilde{\delta}(\omega) = \tilde{p}_1(\omega)/\tilde{p}_0(\omega) \). This classification rule can be viewed as the Bayesian analog of the maximum likelihood classifier, and we refer to it as the posterior mean classifier. An alternative Bayesian approach is to marginalize over the posterior distribution of \( \xi \) and take

\[
\bar{p}_k(\omega) \equiv \int_{\Xi} p_k(\omega_\xi) \pi(\xi|y) d\xi,
\]

(11)

where \( \pi(\xi|y) \) is the posterior distribution of \( \xi \) given \( y \) defined on the parameter space \( \Xi \). As before, we let \( \bar{\delta}(\omega) = \bar{p}_1(\omega)/\bar{p}_0(\omega) \) replace \( \delta(\omega) \) in (10) and refer to the resulting classification rule as the posterior predictive classifier.
In the following subsections, our focus is on defining decision functions as it provides a unifying framework for both the model-based classifiers discussed below and the alternative classifiers discussed in Appendix B.

### 3.2 Non-Spatial Classification

Before describing spatial model-based decision problems, we consider the corresponding aspatial versions. For non-spatially dependent binary data, the underlying model is defined by equations (1)–(3), where $\nu_i \equiv 0$ for all $i$. In this case, $\omega = (x^0, \beta)$,

$$p_1(\omega) \equiv p_1(x^0, \beta) = g^{-1}(x^0 \beta),$$

and $p_0(\omega) = 1 - p_1(\omega)$. It follows that the decision function is

$$\delta_{GLM}(\omega) \equiv \delta_{GLM}(x^0, \beta) = p_1(x^0, \beta)/p_0(x^0, \beta) = g^{-1}(x^0 \beta) / (1 - g^{-1}(x^0 \beta)). \quad (12)$$

For the logit link function,

$$\delta_{GLM-L}(\omega) = \delta_{GLM-L}(x^0, \beta) = \exp\{x^0 \beta\}, \quad (13)$$

and for the probit link function,

$$\delta_{GLM-P}(\omega) = \delta_{GLM-P}(x^0, \beta) = \Phi(x^0 \beta) / (1 - \Phi(x^0 \beta)). \quad (14)$$

For both decision functions, we can obtain maximum likelihood, posterior mean, and posterior predictive classifiers. In practice, we evaluate the integral in (11) via Monte Carlo integration: we approximate $\bar{p}_1(\omega)$ by drawing a realization of $Y_0^t \sim \text{Bernoulli}(p_1(x^0, \beta^t))$ for $t = 1, \ldots, T$ and setting $\bar{p}_1(\omega) = \sum_{t=1}^T I(Y_0^t = 1)/T$, where $\beta^t$ are the draws from the posterior distribution of $\beta$, and $T$ is the number of draws from the posterior distribution.

### 3.3 Spatial Classification

For the spatial case, the decision function $\delta(\omega)$ depends on the covariates and regression coefficients, as well as on the categories of the surrounding observations.

We generally define the same decision functions for the probit SGLMM and SGLM, since marginally, these models only differ in the definition of $\Sigma^*(\theta)$. Using equations (6) and (7), it follows that the distribution for the latent variable at an unobserved location is

$$Z^0 | x^0, X, \beta, \theta, Z, Y \sim \mathcal{N}(\mu_{Z^0}, \sigma_{Z^0})$$
where

\[ \mu_{Z^0} = x^0 \beta + \sigma(\theta)'_{-1} (\Sigma^*(\theta))^{-1} (Z - X'\beta), \]  
\[ \sigma^2_{Z^0} = \sigma(\theta)_1 - \sigma(\theta)'_{-1} (\Sigma^*(\theta))^{-1} \sigma(\theta)_1, \]

\( X = (x_1, \ldots, x_n)' \), \( \sigma(\theta) \) is an \((n+1) \times 1\) vector representing the variance of \( Z^0 \) and \((Z^0, Z)'\), \( \sigma(\theta)_1 \) is \( \sigma(\theta) \) with the first element removed, and \( \Sigma^*(\theta) \) is the spatial correlation matrix among \( Z = (Z_1, \ldots, Z_n)' \). We can easily include sampling from this distribution in the first step of the MCMC algorithm so that we can obtain draws from the posterior distribution of \( Z^0 \). In this spatially-dependent case, \( \omega = (x^0, \beta, \theta, Z, Y) \) and

\[ p_1(\omega) \equiv p_1(x^0, \beta, \theta, Z, Y) = P(Z^0 > 0|x^0, \beta, \theta, Z, Y) = \Phi \left( \frac{\mu_{Z^0}}{\sqrt{\sigma^2_{Z^0}}} \right). \]

Thus, the decision function at \( x^0 \) is

\[ \delta_{SGLM}(\omega) \equiv \delta_{SGLM}(x^0, \beta, \theta, Z, Y) = \frac{P(Z^0 > 0|x^0, \beta, \theta, Z, Y)}{1 - P(Z^0 > 0|x^0, \beta, \theta, Z, Y)} = \frac{\Phi \left( \mu_{Z^0}/\sqrt{\sigma^2_{Z^0}} \right)}{1 - \Phi \left( \mu_{Z^0}/\sqrt{\sigma^2_{Z^0}} \right)}. \]

4 Comparison of the Classification Methods

4.1 Error Rates

When developing and assessing classification methods, the observations are randomly divided into two subsets, the training and test data sets, of sizes \( n_{\text{train}} \) and \( n_{\text{test}} \), respectively. This allows us to fit the model on which the classifier is based using the training data and evaluate the method’s out-of-sample predictive ability. When we compare classification methods, we consider training and test error rates, or the number of incorrect classifications among training and test data sets, divided by the sample size of each data set, respectively.

In Section 3.1 we described three model-based classifiers derived from different approaches for estimating the decision function: maximum likelihood, posterior mean, and posterior predictive. Determining error rates using maximum likelihood and posterior mean classifiers is straightforward. However, for spatial classification, determining training error rates based on the posterior predictive classifier is not. Under the posterior predictive classifier, for the test data, we can simply use the latent \( Z_j \), for \( j = 1, \ldots, n_{\text{test}} \), as sampled within the Gibbs sampler. However, for the training data, the latent \( Z_i \), for \( i = 1, \ldots, n_{\text{train}} \), are
sampled within the Gibbs sampler given the observed \( Y_i \). In this case, to use the latent \( Z_i \) as inputs, we must sample these values as if the \( Y_i \) are unknown, otherwise we would have perfect prediction. For the Bayesian spatial probit regression model, we rely on the observed surrounding observations to provide information about the category of the unobserved locations. Because of this, evaluating the training error is not straightforward. We propose the following two approaches for the spatial probit posterior predictive classifier:

A. One-at-a-Time Training Error:

(i) Take samples \((\beta^{[t]}, \theta^{[t]}, Z^{[t]}_i)\) for \( t = 1, \ldots, T \), where \( Z^{[t]}_i \) is an \((n - 1) \times 1\) vector of sampled \( Z_j \) for \( j = 1, \ldots, i - 1, i + 1, \ldots, n_{\text{train}} \) and sample a corresponding \( Z^{[t]}_i \sim N(\mu_{Z_i}, \sigma^2_{Z_i}) \) where

\[
\hat{\mu}_{Z_i} = x'_i \beta^{[t]} + \sum^* (\theta^{[t]}_{i-i})^{-1} (Z^{[t]}_i - X_{-i} \beta^{[t]})
\]

\[
\hat{\sigma}^2_{Z_i} = \sum^* (\theta^{[t]}_{i-i}) - \sum^* (\theta^{[t]}_{i-i})^{-1} \sum^* (\theta^{[t]}_{i-i})^{-1}
\]

and \( X_{-i} \) is \( X \) with the \( i \)-th row removed, and \( \sum^* (\theta^{[t]}_{j-k}) \) is the \( j \)-th row of the estimated spatial correlation of \( Z \) with the \( k \)-th column removed. (Note that \( Z^{[t]}_i \) are the posterior samples obtained from the MCMC algorithm, however, the \( Z^{[t]}_i \)'s are not the same as those sampled in the MCMC algorithm.)

(ii) Determine \( \bar{p}_1(\omega) = \sum_{t=1}^T I(Z^{[t]}_i > 0) / T \) (suppressing the notation for the \( i \)-th observation in \( \bar{p}_1(\omega) \)), and let \( Y^0_i \) be the predicted value of \( Y_i \) using the posterior predictive classifier.

(iii) Repeat (i) and (ii) for all \( i = 1, \ldots, n_{\text{train}} \).

(iv) Compute the one-at-a-time training error: \( \sum_{i=1}^{n_{\text{train}}} I(Y^0_i \neq Y_i) / n_{\text{train}} \)

B. Joint Training Error:

(i) Take samples \((\beta^{[t]}, \theta^{[t]})\) for \( t = 1, \ldots, T \) and sample a corresponding \( Z^{[t]} \sim N(X\beta^{[t]}, \sum^*(\theta^{[t]})) \). (Note that the \( Z^{[t]} \) are not the same as those sampled in the MCMC algorithm.)

(ii) For \( i = 1, \ldots, n_{\text{train}} \), compute \( \bar{p}(\omega) = \sum_{t=1}^T I(Z^{[t]}_i > 0) / T \) and let \( Y^0_i \) be the predicted value of \( Y_i \) using the posterior predictive classifier.

(iii) Compute the joint training error: \( \sum_{i=1}^{n_{\text{train}}} I(Y^0_i \neq Y_i) / n_{\text{train}} \)

Both the one-at-a-time and joint training errors allow for spatial dependence among the binary predictions/classifications through the latent random variable. The joint training error
allows for spatial dependence only through the spatial dependence structure of the latent variables, $\Sigma^*(\theta)$. In contrast, the one-at-a-time training error allows for spatial dependence through $\Sigma^*(\theta)$, but also allows for spatial dependence by conditioning on the current values of the latent random variables at nearby locations, $Z_{t_i}^{[t]}$.

4.2 Land Cover Application

To illustrate our methods, we use satellite-derived land cover observations over Southeast Asia. In Southeast Asia, deforestation is a major concern and, over the last century, much of the original forests—as much as 12 percent—have been lost to other land uses [Munroe et al., 2008]. Researchers are interested in the economic, geographic, social, and demographic factors that contribute to deforestation and other land cover patterns.

The particular data used in our analyses are the National Aeronautics and Space Exploration (NASA)’s Moderate Resolution Imaging Spectroradiometer (MODIS) Land Cover Type Yearly Level 3 Global 500m (MOD12Q1 and MCD12Q1) data product for the year 2005. We selected land cover observations from this data product corresponding to the region bounded by 17° to 19°N and 98° to 100°E, which covers a portion of northwestern Thailand and a small part of Myanmar. Using a 24 × 24 grid over this region, we collapsed the response variable to two categories, forest and non-forest, where the land cover response variable associated with each grid cell was taken to be the most common observed land cover type. Note that forest was coded as “1” and non-forest was coded as “0”.

We consider four covariates: elevation, distance to the nearest major road, distance to the coast, and distance to the nearest big city. Elevation is measured in meters and distances are Euclidean and measured in degrees. The covariates are standardized, meaning that there were no costs taken into account in calculating distance (e.g., distance calculations do not take into account the fact that it might take longer to go over mountains than go around them).

With satellite data, missing observations often occur in clumps of missing data, rather than scattered points of missing data. Because of this, we randomly selected two test data sets: non-clustered and clustered. For the non-clustered test data, we randomly selected $n_{test} = 144$ locations as test data and used the remaining $n_{train} = 432$ as training data. For the clustered test data, we randomly selected 36 locations and then randomly selected four of each location’s eight neighbors (all locations, even those on the boundary, have 8 neighbors since we have access to the data over a larger region), removing any repeats and locations outside the region. This resulted in $n_{test} = 159$ locations assigned to the clustered test data, and $n_{train} = 417$ assigned to the training data. Figure 1 shows images of both the non-clustered and clustered training and test data sets.
Figure 1: Non-clustered (left) and clustered (right) training (dark colors) and test (light colors) data sets. Forested locations are in green and non-forest locations are brown.

4.3 Prior and Tuning Parameter Values

The Bayesian probit models require prior distributions on parameter values. For the independent probit model (i.e., $\Sigma^*(\theta)$ is the $n \times n$ identity matrix in (7)), SGLM, and SGLMM, we assign prior distribution $\beta \sim \mathcal{N}(0, 10 \times I)$, where $\beta$ are the coefficients for the centered and scaled covariates. For the working parameter in the data augmentation algorithms (see Appendix A), we use a scaled-inverse $\chi^2$ distribution, with $\nu_0 = 3$ degrees of freedom and $a_0 = 3$ as the scale parameter. For the spatial dependence parameter in SGLM and SGLMM, $\theta \sim \text{Unif}(a_\theta, b_\theta)$. We used a conditionally autoregressive (CAR) dependence structure (see, for example, Banerjee et al., 2008) and set $a_\theta = 0$ and $b_\theta = 1$. The SGLMM has an additional parameter, which we fit as a ratio (see Appendix A), $\kappa$, and assign it a Unif(0, 1) prior distribution. To fit each of these models, we run the MCMC for 120,000 iterations, at which point all chains are well-converged and use the last 100,000 for prediction and determining error rates.

Support vector machines (SVM) and k-nearest neighbors (kNN) classification methods require tuning parameters, $\lambda$, $\nu$, and $k$ (see Appendix B). We obtain optimal values of $\lambda$ and $k$ using five-fold cross-validation on each training data and assigned the value of the associated parameter to be the value with the lowest cross-validation error (CVE). Table I shows the chosen tuning parameter and associated CVEs for each classification method and data set.
| Classification Method | Tuning Parameter | Non-Clustered Optimal Value | CVE | Clustered Optimal Value | CVE |
|-----------------------|-----------------|-----------------------------|-----|------------------------|-----|
| Linear SVM            | $\lambda$       | 0.39                        | 0.2767 | 0.51                   | 0.3084 |
| Cubic SVM             | $\lambda$       | 0.98                        | 0.2837 | 0.8                     | 0.2964 |
| Radial SVM            | $\lambda, v$    | 0.515; 5.152                | 0.2140 | 0.875, 7.778            | 0.2410 |
| kNN-C                 | $k$             | 5                           | 0.2326 | 12                     | 0.2506 |
| kNN-G                 | $k$             | 5                           | 0.1953 | 3                      | 0.1880 |

Table 1: Optimal value of tuning parameters and the associated five-fold cross-validation errors for each classification method and data set.

4.4 Results

We compared error rates for the model-based classifiers discussed in Section 3 with alternative classification methods described in Appendix B: linear, diagonal, and quadratic discriminant analysis classifiers (LDA, DLDA, and QDA, respectively); linear, cubic, and radial support vector machines (SVM); and k-nearest neighbor (kNN) classifiers where neighbors are defined either in covariate space (kNN-C; the traditional approach) or geographic space (kNN-G). Table 2 shows the training and test error rates for each classification method. We considered two scenarios of missing data: non-clustered and clustered training/test data sets (see Figure 1).

The training and test error rates tend to be similar for each classification method. Furthermore, the non-clustered and clustered data sets have similar error rates. Most of the error rates from aspatial methods fall around 30%; however, the classification methods which make use of neighboring observations (i.e., the SGLM, SGLMM, and kNN-G classifiers) have much lower error rates.

The SGLM and SGLMM error rates are similar, with the SGLM doing slightly better. In Section 2, we noted the similarity of these two models – that they differ only in that the SGLMM has an additional term in the variance of the random effects. Because of this additional term, we might expect the SGLMM predictions to be slightly more noisy, and this is exhibited in the error rates. However, when we fit the SGLMM, the posterior variance of the spatial random effect is much larger than that aspatial variance component (i.e., $\kappa$ is close to 1 – see Appendix A), and therefore, the spatial dependence dominates the noise of the SGLMM probit link. In this case, the SGLMM in effect converges to the SGLM. While not included here, we note that the reported error rates were similar when we used a geostatistical-type exponential covariance structure rather than a CAR model. Therefore, we do not believe the differences in error rates are a consequence of the particular form of the spatial dependence structure and for computational convenience we prefer the CAR model.

The SGLM joint training error rates are slightly better than the error rates from the inde-
| Classification Method          | Non-clustered Data | Clustered Data |
|-------------------------------|--------------------|----------------|
|                               | Training | Test | Training | Test |
| **Model-based Classifiers**   |          |      |          |      |
| *SGLM*                        |          |      |          |      |
| Posterior Mean                | 0.1551   | 0.1667 | 0.1775   | 0.1447 |
| Posterior Predictive          | 0.1551; 0.2731 | 0.1667 | 0.1775; 0.2998 | 0.1447 |
| *SGLMM*                       |          |      |          |      |
| Posterior Mean                | 0.1644   | 0.1736 | 0.1823   | 0.1447 |
| Posterior Predictive          | 0.1829; 0.2894 | 0.1667 | 0.1823; 0.3046 | 0.1447 |
| *Probit - Bayesian*           |          |      |          |      |
| Posterior Mean                | 0.2824   | 0.2986 | 0.2854   | 0.2767 |
| Posterior Predictive          | 0.2824   | 0.2986 | 0.2854   | 0.2767 |
| *GLM - Maximum Likelihood*    |          |      |          |      |
| Logistic                      | 0.2824   | 0.3056 | 0.2854   | 0.2767 |
| Probit                        | 0.2824   | 0.2986 | 0.2854   | 0.2767 |
| **Alternative Classifiers**   |          |      |          |      |
| *Discriminant Analysis*       |          |      |          |      |
| LDA                           | 0.2778   | 0.2917 | 0.2878   | 0.2704 |
| DLDA                          | 0.3611   | 0.3889 | 0.3549   | 0.3522 |
| QDA                           | 0.2593   | 0.2986 | 0.2638   | 0.2767 |
| *SVM*                         |          |      |          |      |
| Linear SVM                    | 0.2847   | 0.3264 | 0.2926   | 0.2956 |
| Cubic SVM                     | 0.2685   | 0.2847 | 0.2806   | 0.3396 |
| Radial SVM                    | 0.0949   | 0.2639 | 0.0530   | 0.1950 |
| *k-Nearest Neighbors*         |          |      |          |      |
| kNN-C                         | 0.1597   | 0.2639 | 0.2062   | 0.2579 |
| kNN-G                         | 0.1204   | 0.1458 | 0.1055   | 0.2075 |

Table 2: Training and test errors for the SE Asia land cover data obtained using various classification methods. The posterior predictive errors for the training data list two errors: one-at-a-time (left) and joint (right).

Independent GLM models, providing evidence that allowing for spatial dependence in the model can help predictions. However, they are not as small as the one-at-a-time training error rates which has the added benefit of using of the observed classes of neighboring locations. The one-at-a-time training error rates are also more similar to the test error rates, implying that the one-at-a-time training error rates would be more useful in a practical classification setting.
For the clustered test data, the SGLM and SGLMM-based classifiers have the lowest error rates. kNN-G also uses neighboring information in classification, and has small error rates relative to the other competing methods. Although this classifier has smaller rates for the training data, it does not have the smallest error rates across the board. Furthermore, the error rates for this classifier are not fixed because of the random classification when there are ties. For example, repeating this classification method several times for the non-clustered data, the test error rates for kNN-G ranged from 12% up to 21%. Although including only the classes of neighboring locations improves the error rates over those methods which do not make use of neighbors, also including covariates in the classifier – as in the SGLM or SGLMM – provides as low and more consistent error rates.

SVM classification methods offer more flexibility in the relationship between the inputs and the probability of the classes than a GLM. The radial SVM classifier does quite well, and in fact has the smallest training error rates. However, the test error rates are not as small as those for the spatial methods, suggesting that allowing for this more complicated structure among the inputs is not sufficient to produce better predictions when classifying out-of-sample data in space.

Comparing the error rates of all these classification methods suggests that including inputs associate with the focal location as well as neighboring observations, as in the SGLM and SGLMM classifiers, leads to better classification methods.

5 Discussion

In this paper, we develop Bayesian classifiers for spatial data by building on the Bayesian probit SGLM and SGLMM. Using a illustrative example of land cover over Southeast Asia, the classifiers we proposed were more consistently accurate at predicting unobserved data than other aspatial classification methods. While the SGLM and the SGLMM differ slightly in model-specification, in our empirical example, the classifiers performed about the same and both appear to be powerful tools for spatial classification. k-nearest neighbors using geographic space and radial SVM also offered small error rates, but the SGLM and SGLMM classifiers which account for both the inputs and the spatial neighbors performed better for the out-of-sample data.

While we did not consider computation time in determining the best classifier, this may be a factor in some classification contexts. Comparing the SGLM to the SGLMM, the SGLM appears to have a slight computational advantage, as it has one less parameter and for the CAR dependence structure it does not require inverting the $n \times n$ covariance matrix. However, both these models are much more computationally intensive than the others described in this paper.
We focused on building spatial classifiers using a probit SGLM and SGLMM, but other spatial models may also offer the ability to classify using inputs and neighboring observations. Hoeting et al. (2000) consider this scenario using an autologistic model.

Finally, we note that this analysis assumes that the unobserved locations are missing completely at random (MCAR), meaning that the locations of the missing data are unrelated to other variables. Of course, this may not be the case in practice, and future work could investigate the utility of these classifiers under other missing data scenarios.

Acknowledgements

The authors gratefully acknowledge support for this work from NASA’s Land-Cover/Land-Use Change Program (NNG06GD31G), as well as from the National Science Foundation (DMS-0934595 and DMS-1209161). They also thank Dr. Ningchuan Xiao, Department of Geography at The Ohio State University, for his assistance working with the land cover data.

References

Albert, J. H. and Chib, S. (1993). “Bayesian analysis of binary and polychotomous response data.” Journal of the American Statistical Association, 88, 669–679.

Banerjee, S., Carlin, B., and Gelfand, A. (2004). Hierarchical Modeling and Analysis for Spatial Data. Boca Raton, FL: Chapman & Hall/CRC.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). “Gaussian predictive process models for large spatial data sets.” Journal of the Royal Statistical Society, Series B, 70, 825–848.

Berrett, C. and Calder, C. A. (2012). “Data augmentation strategies for the Bayesian spatial probit regression model.” Computational Statistics and Data Analysis, 56, 478–490.

Cortes, C. and Vapnik, V. (1995). “Support-vector network.” Machine Learning, 20, 273–297.

De Oliveira, V. (2000). “Bayesian prediction of clipped Gaussian random fields.” Computational Statistics and Data Analysis, 34, 299–314.

Diggle, P. J., Tawn, J. A., and Moyeed, R. A. (1998). “Model-based geostatistics.” Applied Statistician, 47, 299–350.
Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A. (2010). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.5-24.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.

Higgs, M. D. and Hoeting, J. A. (2010). “A clipped latent-variable model for spatially correlated ordered categorical data.” *Computational Statistics and Data Analysis*, 54, 1999–2011.

Hoeting, J. A., Leecaster, M., and Bowden, D. (2000). “An Improved Model for Spatially Correlated Binary Responses.” *Journal of Agricultural, Biological, and Environmental Statistics*, 5, 102–114.

Hughes, J. and Haran, M. (2013). “Dimension reduction and alleviation of confounding for spatial generalized linear mixed models.” *Journal of the Royal Statistical Society, Series B*, 75, 139–159.

Imai, K. and van Dyk, D. A. (2005). “A Bayesian analysis of the multinomial probit model using marginal data augmentation.” *Journal of Econometrics*, 124, 311–334.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.

Munroe, D. K., Wolfinbarger, S. R., Calder, C. A., Shi, T., Xiao, N., Lamb, C. Q., and Li, D. (2008). “The relationships between biomass burning, land-cover/-use change, and the distribution of carbonaceous aerosols in mainland Southeast Asia: a review and synthesis.” *Journal of Land Use Science*, 3, 161–183.

Paciorek, C. J. (2007). “Computational techniques for spatial logistic regression with large data sets.” *Computational Statistics and Data Analysis*, 51, 3631–3653.

Schliep, E. M. and Hoeting, J. A. (2013). “Multivariate multilevel latent Gaussian process model to evaluate wetland condition.” *Journal of Agricultural, Biological, and Environmental Statistics*, 18, 492–513.
APPENDICES

A Data Augmentation Algorithms

In Section 2, $\epsilon_i$ in equation (6) has a fixed variance of 1. This is for identifiability purposes and also creates the probit link. However, for model-fitting purposes, instead of fixing the variance parameter to 1, we can use and marginalize over a variance parameter (for examples in the Bayesian probit model, see Imai and van Dyk, 2005; Berrett and Calder, 2012). Berrett and Calder (2012) describe a model-fitting algorithm for the probit SGLM. Here we show how this algorithm can be extended to create a data augmentation model-fitting algorithm for the SGLMM. We briefly review the model notation and model-fitting algorithm here and then provide an adapted algorithm for the probit SGLMM. In both algorithms, we make use of a non-identifiable variance parameter of the latent variable to facilitate mixing of the MCMC. Within this appendix, we use the $\tilde{}$ notation to represent unidentifiable parameters. For both algorithms, we used what Berrett and Calder (2012) call the Non-collapsed Marginal-Scheme 1 Algorithm.

Equation (5) in Section 2 describes the spatial covariance matrix $\Sigma(\theta)$ to be a scalar, $\theta_1$, times a spatial dependence matrix $K(\theta_2)$. For the SGLM, $\theta_1$ is the non-identifiable parameter which Berrett and Calder (2012) make use of as a working parameter. For consistency of the $\tilde{}$ notation, let $\tilde{K}(\theta) = \sigma^2 K(\theta)$, where $\tilde{K}(\theta) = \Sigma(\theta)$ is the spatial covariance matrix, $\sigma^2 = \theta_1$ is a non-identifiable scalar used as the working parameter of the algorithm, and $K(\theta)$ is the identifiable spatial dependence matrix parameterized by a single parameter, $\theta$.

The data augmentation model-fitting algorithm for the SGLM is then

**Step 1:** Sample $\sigma^2_{\text{temp}} \sim \pi(\sigma^2)$
Sample $\tilde{Z}|Y, \beta, \theta, \sigma^2_{\text{temp}}$
Set $Z = \tilde{Z}/\sigma_{\text{temp}}$

**Step 2:** Sample $(\sigma^2, \beta)|\tilde{Z}, Y, \theta$
Set $\beta = \tilde{\beta}/\sigma$

**Step 3:** Sample $\theta|\tilde{Z}, Y, \tilde{\beta}, \sigma^2$.

For more details, see Berrett and Calder (2012).

For the SGLMM, we can build on this model-fitting algorithm by adding an additional step to sample the additional parameter. Instead of letting $\epsilon_i \sim N(0, 1)$, let the non-identifiable $\tilde{\epsilon}_i \sim N(0, \tau^2)$. Then,

$$(\nu + \tilde{\epsilon}) \sim N(0, \tilde{\Sigma}'(\theta)),$$
Step 2: Sample \( \kappa \) to the SGLMM, we let \( \gamma \) or the spatial covariance matrix of the SGLM. Therefore, to adapt the SGLM data augmentation algorithm for the SGLMM, we use priors \( \beta \).

Let \( \tilde{Z}_i \) and \( \tilde{Z}_i \) are defined in equations (15) and (16), and set \( \tilde{Z}_i = \tilde{Z}_i / \gamma \).

**Step 1:** Sample from \( Z_i | Y, \beta^{[t-1]}, \theta^{[t-1]}, \kappa^{[t-1]} \):

Draw \( \gamma_{\text{temp}} \sim \pi(\gamma^2) \).

For \( i = 1, \ldots, n \), define \( Z_i \) and sample \( \tilde{Z}_i \) from

\[
\tilde{Z}_i | Y, \beta^{[t-1]}, \theta^{[t-1]}, \kappa^{[t-1]}, \gamma \sim \begin{cases} \mathcal{N}(\mu_{\tilde{Z}_i}, \tau_{\tilde{Z}_i}^2, 0, \infty) & \text{if } Y_i = 1 \\ \mathcal{N}(\mu_{\tilde{Z}_i}, \tau_{\tilde{Z}_i}^2, -\infty, 0) & \text{if } Y_i = 0 \end{cases},
\]

where \( \mathcal{N}(\mu_{\tilde{Z}_i}, \tau_{\tilde{Z}_i}^2, \ell, u) \) is a truncated normal distribution with lower and upper bounds \( \ell \) and \( u \), respectively, and mean and variance

\[
\mu_{\tilde{Z}_i} = \gamma \beta^{[t-1]} + \boldsymbol{Z}_i \beta^{[t-1]} - [\Sigma^*(\theta^{[t-1]})]_{i,i}^{-1} \gamma_{\text{temp}} (Z_i^{[t-1]} - X_i \beta^{[t-1]})
\]

\[
\tau_{\tilde{Z}_i}^2 = \gamma_{\text{temp}}^2 \left( [\Sigma^*(\theta^{[t-1]})]_{i,i}^{-1} - [\Sigma^*(\theta^{[t-1]})]_{i,i}^{-1} \right)
\]

Set \( Z_i = \tilde{Z}_i / \gamma_{\text{temp}} \).

For an unobserved location, sample

\[
\tilde{Z} \sim \mathcal{N}(\mu, \kappa^2)
\]

where \( \mu_{\tilde{Z}} \) and \( \sigma_{\tilde{Z}}^2 \) are defined in equations (15) and (16), and set \( Z_i^{[t]} = \tilde{Z} / \gamma_{\text{temp}} \).

**Step 2:** Sample \( \gamma \) from \( \gamma | Y, \tilde{Z}, \theta^{[t-1]}, \kappa^{[t-1]} \):

Sample

\[
(\gamma^2)^{[t]} \sim \left( (\tilde{Z} - X \beta') (\Sigma^*(\theta^{[t-1]})^{-1}) (\tilde{Z} - X \beta) + a_0^2 + \beta' C_{\beta}^{-1} \beta \right) (\chi_{n+\nu}^2)^{-1},
\]

where \( \Sigma^*(\theta) = \tau^2 I + \Sigma(\theta) = \tau^2 I + \sigma^2 K(\theta) \),

This is simply a scalar times a spatial dependence matrix, just as in the model-fitting algorithm for the SGLM. Notice that \( \kappa \) is bounded by 0 and 1 and that if \( \kappa = 1 \), \( \Sigma^*(\theta) = \sigma^2 K(\theta) \), or the spatial covariance matrix of the SGLM. Therefore, to adapt the SGLM data augmentation algorithm to the SGLMM, we let \( \gamma \) be the non-identifiable working parameter and add a step to sample \( \kappa \).

Here we provide the model-fitting algorithm and the full conditional distributions for the SGLM. We use priors \( \beta \sim \mathcal{N}(0, C_{\beta}) \), \( \gamma \sim a_0 (\chi_{\nu}^2)^{-1} \), \( \theta \sim \pi(\theta) \), and \( \kappa \sim \text{Unif}(0, 1) \). We use superscript \([t]\) to denote the value of a parameter at the \( t \)th iteration of the algorithm. Let \( \Sigma^*(\theta) = \gamma^2 \Sigma^*(\theta) \), where \( \Sigma^*(\theta) = (1 - \kappa) I + \kappa K(\theta) \).
where $\hat{\beta} = (X' (\Sigma^*(\theta^{[t-1]}))^{-1} X + C_\beta^{-1})^{-1} X' (\Sigma^*(\theta^{[t-1]}))^{-1} \tilde{Z}^{[t]}$. Sample

$$\tilde{\beta} \sim \mathcal{N} \left( \hat{\beta}, (\gamma^2)^{[t]} \left( X (\Sigma^*(\theta^{[t-1]}))^{-1} X + C_\beta^{-1} \right)^{-1} \right).$$

Set $\beta^{[t]} = \tilde{\beta} / \gamma^{[t]}$.

**Step 3:** Sample $\theta^{[t]}$ from $\theta | Y, \tilde{Z}^{[t]}, \tilde{\beta}^{[t]}, (\gamma^2)^{[t]}, \kappa^{[t-1]}$ via a random walk Metropolis step:
Sample a proposal value $\theta_{prop}$ from a proposal distribution. We used a normal distribution, $\mathcal{N}(\theta|\theta^{[t-1]}, \tau_\theta^2)$, where $\tau_\theta^2$ is the fixed variance of the proposal distribution. Define

$$\theta^{[t]} = \begin{cases} 
\theta_{prop} & \text{with probability } c(\theta^{[t-1]}, \theta_{prop}) \\
\theta^{[t-1]} & \text{with probability } 1 - c(\theta^{[t-1]}, \theta_{prop})
\end{cases}$$

where

$$c(\theta^{[t-1]}, \theta_{prop}) = \min \left\{ \frac{\pi(\theta_{prop}|Y, \tilde{Z}, \tilde{\beta}, (\gamma^2)^{[t]}, \kappa^{[t-1]})}{\pi(\theta^{[t-1]}|Y, \tilde{Z}, \tilde{\beta}, (\gamma^2)^{[t]}, \kappa^{[t-1]})}, 1 \right\}.$$

The posterior distribution of $\theta$ in the acceptance probability is

$$\pi(\theta_{prop}|Y, \tilde{Z}, \tilde{\beta}, (\gamma^2)^{[t]}, \kappa^{[t-1]}) \propto \phi(\tilde{Z}; X\tilde{\beta}, \gamma^2 \Sigma^*(\theta)) \pi(\theta),$$

where $\phi(\cdot)$ is the multivariate normal density function.

**Step 4:** Sample $\kappa^{[t]}$ from $\kappa | Y, \tilde{Z}^{[t]}, \tilde{\beta}^{[t]}, (\gamma^2)^{[t]}, \theta^{[t]}$ via a random walk Metropolis step:
Sample a proposal value $\kappa_{prop}$ from a proposal distribution. We used a normal distribution, $\mathcal{N}(\kappa|\kappa^{[t-1]}, \tau_\kappa^2)$, where $\tau_\kappa^2$ is the fixed variance of the proposal distribution. Define

$$\kappa^{[t]} = \begin{cases} 
\kappa_{prop} & \text{with probability } c(\kappa^{[t-1]}, \kappa_{prop}) \\
\kappa^{[t-1]} & \text{with probability } 1 - c(\kappa^{[t-1]}, \kappa_{prop})
\end{cases}$$

where

$$c(\kappa^{[t-1]}, \kappa_{prop}) = \min \left\{ \frac{\pi(\kappa_{prop}|Y, \tilde{Z}, \tilde{\beta}, (\gamma^2)^{[t]}, \theta^{[t]})}{\pi(\kappa^{[t-1]}|Y, \tilde{Z}, \tilde{\beta}, (\gamma^2)^{[t]}, \theta^{[t]})}, 1 \right\}.$$

The posterior distribution of $\theta$ in the acceptance probability is

$$\pi(\kappa_{prop}|Y, \tilde{Z}, \tilde{\beta}, (\gamma^2)^{[t]}, \theta^{[t]}) \propto \phi(\tilde{Z}; X\tilde{\beta}, \gamma^2 \Sigma^*(\theta)) \pi(\kappa),$$

where $\phi(\cdot)$ is the multivariate normal density function.
B Alternative Classification Methods

In this appendix, we provide decision functions for alternatives to the model-based classification methods discussed in Sections 3.2 and 3.3. For these alternative methods, we assume that the \( x \)s only include the inputs, and thus do not include a term to allow for an intercept as in the model-based (GLM/GLMM) classification methods. Unless otherwise noted, the descriptions of the classification methods are based on Hastie et al. (2001).

B.1 Discriminant Analysis

In discriminant analysis, rather than considering the explanatory variables \( x \) as fixed as they are in regression analyses, the \( x \)s are viewed as random variables, with class-specific density functions \( f_j(\cdot) \) corresponding to each class \( C_j \). The classes also have prior probabilities \( \pi_j \), such that \( \pi_0 + \pi_1 = 1 \). To determine the probability that a set of inputs will fall into class \( j \), we employ Bayes’ theorem which implies that

\[
P(C_j|x) = \frac{f_j(x)\pi_j}{f_1(x)\pi_1 + f_0(x)\pi_0}.
\]

(17)

As with the model-based classification methods, the Bayes’ classifier is to classify an observation to class \( C_1 \) when \( P(C_1|x) > P(C_0|x) \) and to \( C_0 \) otherwise.

We first describe discriminant analysis in its general form, allowing a arbitrary form for the \( f_j(\cdot) \)s and the decision function, and then discuss threes special cases that we use in our analysis. Let

\[
X_j = \begin{bmatrix} x_1 \\ \vdots \\ x_{n_j} \end{bmatrix}
\]

where \( \{x_1, \ldots, x_{n_j}\} = \{x_i : y_i = j\} \) and \( X_j \) is an \( n_jk \times 1 \) vector of inputs corresponding to observations in class \( C_j \). To classify \( Y^0 \), for each class we define

\[
X_j^0 = \begin{bmatrix} x^0 \\ X_j \end{bmatrix},
\]

where \( X_j^0 \) is a \( (n_j+1)k \times 1 \) vector and \( x^0 \) is a \( k \times 1 \) vector of inputs associated with \( Y^0 \). Our goal is to determine a decision boundary for classifying \( Y^0 \).

In discriminant analysis, \( f_j(\cdot) \) is typically the multivariate normal density function,

\[
f_j(X_j^0) = \frac{1}{(2\pi)^{(n_j+1)k/2}|\Sigma_j|^{1/2}} \exp \left\{ -\frac{1}{2}(X_j^0 - \mu_j^X)'(\Sigma_j^X)^{-1}(X_j^0 - \mu_j^X) \right\},
\]

(20)
where $\mu_j^X$ is the $(n_j + 1) \times 1$ class-specific mean vector and $\Sigma_j^X$ is the $(n_j + 1) \times (n_j + 1)$ class-specific covariance matrix. It follows that

$$f_j(x^0|x_1, \ldots, x_{n_j}) = \frac{1}{(2\pi)^{k/2}|\Sigma_j^x|^{1/2}} \exp \left\{ -\frac{1}{2} (x^0 - \mu_j^x)' \left( \Sigma_j^x \right)^{-1} (x^0 - \mu_j^x) \right\}$$

(18)

where

$$\mu_j^x = \mu_j(\{1:k\}) + \Sigma_j^X (\Sigma_j^x)^{-1} (x_j - \mu_j(\{1:k\}))$$

$$\Sigma_j^x = \Sigma_j^X (\Sigma_j^x)^{-1} \Sigma_j^X - \Sigma_j^X (\Sigma_j^x)^{-1} \Sigma_j^X (\Sigma_j^x)^{-1} \Sigma_j^X.$$

We use the subscript notation ($\{1:k\}$) to indicate the first $k$ elements of the corresponding matrix or vector (i.e., those indices corresponding to $x^0$) and ($-\{1:k\}$) indicates the matrix or vector without the first $k$ elements (i.e., the remaining indices corresponding to $X_j$).

Considering the log-odds, it follows from (17) and (18) that

$$\log \left( \frac{P(C_1|x^0)}{P(C_0|x^0)} \right) = \log \frac{\pi_1}{\pi_0} + \frac{1}{2} \log \frac{\Sigma_0^x}{\Sigma_1^x} - \frac{1}{2} \mu_1^x \mu_0^x (\Sigma_1^x)^{-1} \mu_0^x + \frac{1}{2} \mu_0^x (\Sigma_0^x)^{-1} \mu_0^x$$

(19)

$$+ x_0^0 (\Sigma_1^x)^{-1} \mu_1 - (\Sigma_0^x)^{-1} \mu_0 + \frac{1}{2} \left( (\Sigma_1^x)^{-1} - (\Sigma_0^x)^{-1} \right) x_0^0.$$

Here, $\alpha_0$ is a scalar, $\alpha_1$ is a $k \times 1$ vector, and $\alpha_2$ is a $k \times k$ matrix, which we define for notational convenience. A discriminant-analysis decision function corresponding to the classification rule defined in equation (10) is

$$\delta_{DA}(\omega) \equiv \delta_{DA}(x^0, \mu_0^x, \mu_1^x, \Sigma_0^x, \Sigma_1^x) = \exp \{ \alpha_0 + x_0^0 \alpha_1 - x_0^0 \alpha_2 x_0^0 \}.$$

(20)

We now consider special cases to (20). Each of these special cases assumes that the mean of $x_i$ is equal across all observations, so that $\mu_j^X = [\mathbf{1}_{n_j+1} \otimes \mu_j]$ where $\mathbf{1}_{n_j+1}$ is an $(n_j + 1) \times 1$ vector of ones and $\mu_j$ is a $k \times 1$ class specific mean vector. The difference between each of these special cases is in the specification of the covariance matrix $\Sigma_j^X$. We describe three popular discriminant analysis methods (linear discriminant analysis, diagonal linear discriminant analysis, and quadratic discriminant analysis) all of which assume that the $x_i$ are independent.

Assuming the $x_i$ are independent results in the following form for the covariance of $X_j^0$:

$$\text{var}(X_j^0) = \Sigma_j^X = (\mathbf{I}_{n_j+1} \otimes \Lambda_j),$$

(21)

where $\mathbf{I}_{n_j+1}$ is an $(n_j + 1) \times (n_j + 1)$ identity matrix and $\Lambda_j$ is a class-specific covariance matrix for the $k$ components of $x_i$. Under this assumption, $x^0$ is independent of $x_1, \ldots, x_{n_j}$, so

$$f_j(x^0|x_1, \ldots, x_{n_j}) = f_j(x^0) = \frac{1}{(2\pi)^{k/2}|\Lambda_j|^{1/2}} \exp \left\{ -\frac{1}{2} (x^0 - \mu_j)' \Lambda_j^{-1} (x^0 - \mu_j) \right\}.$$

21
Assuming a constant variance across classes (i.e., $\mathbf{\Lambda}_j = \mathbf{\Lambda}$ for $j = 0, 1$) results in linear discriminant analysis (LDA) because the decision boundary is linear in the $x$'s. The log odds in equation (19) can be written in this case as

$$
\log \left( \frac{P(C_1|x^0)}{P(C_0|x^0)} \right) = \log \left( \frac{\pi_1}{\pi_0} - \frac{1}{2} (\mu_1 + \mu_0)' \mathbf{\Lambda}^{-1} (\mu_1 - \mu_0) + x^0' \mathbf{\Lambda}^{-1} (\mu_1 - \mu_0) \right),
$$

where $a_0^{LDA}$ is a scalar and $\alpha_1^{LDA}$ is a $k \times 1$ vector, defined for notational convenience. Therefore, the decision function for LDA is

$$
\delta_{LDA}(\omega) \equiv \delta_{LDA}(x^0, \mu_0, \mu_1, \mathbf{\Lambda}) = \exp \{ a_0^{LDA} + x^0' \alpha_1^{LDA} \}. \quad (22)
$$

Note that this decision function is effectively equivalent to the one based on the logistic regression model in (13), however, in logistic regression, we assume the $x$'s are fixed and thus make no distributional assumptions on $x$ as in discriminant analysis.

In practice, the parameters $\pi_j, \mu_j, \mathbf{\Lambda}$ (and thus $a_0^{LDA}$ and $\alpha_1^{LDA}$) are unknown but can be estimated using maximum likelihood:

- $\hat{\pi}_j = n_j/n$,
- $\hat{\mu}_j = \sum_{i:y_i=j} x_i/n_j$,
- $\hat{\mathbf{\Lambda}} = \sum_{j \in \{0,1\}} \sum_{i:y_i=j} (x_i - \hat{\mu}_j)' (x_i - \hat{\mu}_j) / (n-2)$.

Diagonal linear discriminant analysis (DLDA) additionally assumes independence between the $k$ inputs so that $\text{var}(x_i) = \mathbf{\Lambda}$ is a diagonal matrix. The DLDA-based decision function is the same as (22), but using a diagonal matrix $\mathbf{\Lambda}$. The $m^{th}$ diagonal element of $\mathbf{\Lambda}$ is estimated by $\hat{\Lambda}_{(m,m)} = \sum_{j \in \{0,1\}} \sum_{i:y_i=j} (x_{im} - \hat{\mu}_{jm})^2 / (n-2)$ where $x_{im}$ and $\hat{\mu}_{jm}$ are the $m^{th}$ elements of $x_i$ and $\mu_j$, respectively.

In LDA, we assume a constant covariance for $x_i$ among the classes (i.e., $\mathbf{\Lambda}_j = \mathbf{\Lambda}$ for $j = 0, 1$). On the other hand, quadratic discriminant analysis (QDA) allows for each class to have its own covariance. The log odds now contains a quadratic term in the $x$'s:

$$
\log \left( \frac{P(C_1|x^0)}{P(C_0|x^0)} \right) = \log \left( \frac{\pi_1}{\pi_0} + \frac{1}{2} \log \left| \mathbf{\Lambda}_0 \right| - \frac{1}{2} \mu_1' \mathbf{\Lambda}_1^{-1} \mu_1 + \frac{1}{2} \mu_0' \mathbf{\Lambda}_0^{-1} \mu_0 \right) + x^0' \left( \mathbf{\Lambda}_1^{-1} \mu_1 - \mathbf{\Lambda}_0^{-1} \mu_0 \right) - x^0' \left( \frac{1}{2} (\mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_0^{-1}) \right) x^0. \quad \alpha_2^{QDA}
$$

Note that this decision function is effectively equivalent to the one based on the logistic regression model in (13), however, in logistic regression, we assume the $x$'s are fixed and thus make no distributional assumptions on $x$ as in discriminant analysis.
Here, $\alpha_0^{QDA}$ is a scalar, $\alpha_1^{QDA}$ is a $k \times 1$ vector, and $\alpha_2^{QDA}$ is a $k \times k$ matrix, which are defined for notational convenience. This results in the QDA-based decision function:

$$
\delta_{QDA}(\omega) \equiv \delta_{QDA}(x^0, \mu_0, \mu_1, \Lambda_0, \Lambda_1) = \exp\{\alpha_0^{QDA} + x^0' \alpha_1^{QDA} + x^0' \alpha_2^{QDA} x^0\},
$$

where we estimate the parameters by taking

- $\hat{\pi}_j = n_j/n,$
- $\hat{\mu}_j = \sum_{i:y_i=j} x_i/n_j,$
- $\hat{\Sigma}_j = \sum_{i:y_i=j} (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)'/(n_j - 1).$

### B.2 Support Vector Machines

First introduced by Cortes and Vapnik (1995), the goal of support vector machines (SVM) is to determine a hyperplane in covariate space separating the classes in such a way that the margin between the two classes is maximized. The margin is the minimum distance between the the inputs $x_i$ of the two classes in the direction perpendicular to the hyperplane. The resulting function determining this hyperplane is the decision function.

In SVM, the classes are labeled as either 1 or $-1$ (instead of 1 or 0, as before). To accommodate this convention, we redefine observations $y_i^* = 2y_i - 1$, for $i = 1, \ldots, n$, so that $y_i^* \in \{-1, 1\}$.

Consider the decision function

$$
\delta_{SVM}(\omega) \equiv \delta_{SVM}(x, \beta, \beta_0) = \exp\{x' \beta + \beta_0\}.
$$

(23)

Given observations $\{y_i^*, x_i\}$ for $i = 1, \ldots, n$, maximizing the margin between the two classes and the hyperplane is equivalent to minimizing $||\beta||$ subject to $y_i^*(x_i' \beta + \beta_0) \geq 1$ for all $i = 1, \ldots, n$. This problem can be represented as the following Lagrange optimization function

$$
\max_{\zeta} L = \max_{\zeta} \left(\sum_{i=1}^n \zeta_i - \frac{1}{2} \sum_{i=1}^n \sum_{i^*=1}^n \zeta_i \zeta_i^* y_i^* y_{i^*}^* x_i x_{i^*}'\right),
$$

(24)

where $\zeta = (\zeta_1, \ldots, \zeta_n)'$ are the Lagrangian multipliers, which are subject to the constraints that $\sum_{i=1}^n \zeta_i y_i^* = 0$ and $\zeta_i \geq 0$ for all $i$. The $x_i$ where $\zeta_i > 0$ are the support vectors and are the only vectors which influence the position of the hyperplane. We can write the relationship between $\zeta$ and $\beta$ as

$$
\beta = \sum_{i=1}^n \zeta_i y_i^* x_i
$$

23
and the relationship between $\zeta$ and $\beta_0$ as

$$\beta_0 = \frac{1}{n_{sv}} \sum_{i: \zeta_i > 0} (\beta x_i - y_i^*)$$

where $n_{sv}$ is the number of support vectors.

While the above hyperplane is linear, SVM can be extended to create nonlinear boundaries between the classes. This extension can be achieved by transforming the inputs into a space where they can be separated linearly, and again find the separating hyperplane in this transformed covariate space. We can use the Lagrange optimization function in (24) with the transformed inputs $K(x_i, x_j)$:

$$\max L = \max \left( \sum_{i=1}^{n} \zeta_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \zeta_i \zeta_j y_i^* y_j^* K(x_i, x_j) \right)$$

subject to $\sum_{i=1}^{n} \zeta_i y_i^* = 0$ and $0 \leq \zeta_i \leq \lambda$ for all $i$, where $\lambda$ is a tuning parameter allowing for crossover among the two classes and $K(\cdot, \cdot)$ is a symmetric positive (semi-) definite function.

In our data analysis, we consider the following three popular kernels:

- Linear: $K(x_i, x_j) = x_i' x_j$
- $d^{th}$ Degree Polynomial: $K(x_i, x_j) = (1 + x_i' x_j)^d$
- Radial: $K(x_i, x_j) = \exp\{-v||x_i - x_j||^2\}$, where $v$ is a fixed constant.

Now, (23) can be written as

$$\delta_{SVM}(\omega) \equiv \delta_{SVM}(x, \zeta, \beta_0, \lambda) = \sum_{i=1}^{n} \zeta_i y_i^* K(x, x_i) + \beta_0,$$

(26)

And $\hat{\delta}_{SVM}(x^0, \zeta, \beta_0, \lambda) = \sum_{i=1}^{n} \hat{\zeta}_i y_i K(x^0, x_i) + \hat{\beta}_0$.

When implementing this classification method, we use the R package e1071 \cite{Dimitriadou et al., 2010}, to compute $\hat{\zeta}_i$ and $\hat{\beta}_0$ via a quadratic optimization function for a fixed value of $\lambda$.

### B.3 k-Nearest Neighbors in Covariate Space

The k-nearest neighbors (kNN) classification method makes no assumptions about an underlying model. Using this method, for a point $\{Y^0, x^0\}$, the closest $k$ points $\{x(r), r = 1, \ldots, k\}$
to $x^0$ are identified, and $Y^0$ is assigned to the most popular class among the $k$ neighbors, where ties are broken at random. “Distance” here is measured in covariate space, not geographic space, and could be defined using any valid distance metric. In our implementation of the method, we use Euclidean distance so that

$$d_{(r)} = ||x_{(r)} - x^0||.$$ 

Here, $d_{(i)}$ represents the ordered distances where the minimum is $d_{(1)}$ and the maximum is $d_{(n)}$, and $x_{(r)}$ are the $x_i$ corresponding to $d_{(r)}$. Using this measure of distance requires standardization of the variables so that no variable is given more weight than another.

For the binary case, a decision function can then be defined as

$$\delta_{kNN-C}(\omega) \equiv \delta_{kNN-C}(x^0, x, Y) = 2 \sum_{r=1}^{k} y_{(r)} / k;$$

where $y_{(r)}$ is the $y_i$ associated with $d_{(r)}$. When classifying $Y^0$, if $\delta_{kNN-C}(\omega) = 1$, $y^0_{pred} = 1$ with probability .5 and $y^0_{pred} = 0$ with probability .5.

### B.4 k-Nearest Neighbors in Geographic Space

Instead of using covariates to determine proximity as in Section B.3, we can also use geographic space. Using this approach, for prediction at location $s^0$, the closest $k$ observed points are identified $\{s_{(r)}, r = 1, \ldots, k\}$, and $Y^0$ is assigned to the most popular class among its $k$ geographic neighbors. Just as in k-Nearest Neighbors for covariate space, we can use any distance metric, but we again use Euclidean distance so that

$$d^*_{(r)} = ||s_{(r)} - s^0||.$$ 

The $d^*_{(i)}$ represent ordered distances due to geographical space, and $s_{(r)}$ are the $s_i$ corresponding to $d^*_{(r)}$. The decision function for kNN based on geographic space can then be defined as

$$\delta_{kNN-G}(\omega) \equiv \delta_{kNN-G}(s^0, s, Y) = 2 \sum_{1}^{k} y^*_{(r)} / k;$$

where $y^*_{(r)}$ are the $y_i$ associated with $d^*_{(r)}$. When classifying $Y^0$, if $\delta_{kNN-G}(\omega) = 1$, we assign $y^0_{pred} = 1$ with probability .5 and $y^0_{pred} = 0$ with probability .5.